IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 26, 2020, accepted May 7, 2020, date of publication May 11, 2020, date of current version May 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993909

Ship Motion Attitude Prediction Based on an Adaptive Dynamic Particle Swarm Optimization Algorithm and Bidirectional LSTM Neural Network

GUOYIN ZHANG, FENG TAN, AND YANXIA WU
College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Corresponding author: Yanxia Wu (wuyanxia@hrbeu.edu.cn)

This work was supported in part by the Natural Science Foundation of Heilongjiang Province under Grant F2018008, in part by the Foundation for Distinguished Young Scholars of Harbin under Grant 2017RAYXJ016, and in part by the Fundamental Research Funds for the Central Universities under Grant 3072019CFT0602.

ABSTRACT A new neural network prediction model is proposed for predicting ship motion attitude with high accuracy. This prediction model is based on an adaptive dynamic particle swarm optimization algorithm (ADPSO) and bidirectional long short-term memory (BiLSTM) neural network, which is to optimize the hyperparameters of BiLSTM neural network by the proposed ADPSO algorithm. The ADPSO algorithm introduces dynamic search space strategy into the classical particle swarm optimization algorithm and adjusts the learning factor adaptively to balance the global and local search ability, so as to improve the optimization performance and improve its optimization effect in BiLSTM parameter optimization process. The results show that the model can obtain higher prediction accuracy and faster convergence speed, and has better prediction performance in the prediction of ship motion attitude.

INDEX TERMS Ship motion attitude, BiLSTM neural network, ADPSO algorithm, prediction accuracy.

I. INTRODUCTION

Under the influence of various external factors such as bad weather, ships sailing on the sea are easy to produce six degrees of freedom random and complicated motions, including roll, pitch, yaw, sway, surge and heave. These movements have a great impact on the safety of ships and its personnel, the efficiency and safety of maritime operations, and especially on the take-off and landing of carrier-borne aircraft on aircraft carriers [1]. As we all know, the aircraft carrier is the key of the navy, in order to safeguard countries' maritime rights and interests, accurate prediction of ship motion attitude is of practical value to effectively adjust and control the ship motion and timely adjust the take-off and landing of carrier aircraft, which is also of great significance to national defense.

Ship motion attitude prediction has been widely studied in recent years. The prediction methods mainly include statistical prediction method, convolution method, gray prediction method, Kalman filter method, time series method, neural network method, and combination prediction method. The statistical prediction method is based on the integral equation analysis. The calculation process is very complicated in order to obtain the power spectrum of the input signal. This method is only suitable for very short-term prediction, and it is difficult to apply to the actual ship motion prediction due to a lot of constraints there. The convolution method [2] is to predict the ship's motion after convolving the wave height measurement function of a selected ship's bow at a certain position with the kernel function of the ship response function because this method needs to know the accurate response kernel function and wave height measurement function, limited in practical applications. The gray prediction method is to use the gray theory to establish the gray differential model, and use the limited information to find the rules among the data through the information processing, so as to carry out the effective prediction. Professor Shen first introduced the theory of gray system into the prediction of ship swaying motion, the experimental results show that the model can basically describe the development trend of real data, but sometimes the error

The associate editor coordinating the review of this manuscript and approving it for publication was Choon Ki Ahn.

is large when the swaying range is large [3]. In addition, the training samples of the gray prediction method cannot be too many, so the gray prediction method has limitations in nonlinear time series prediction such as ship motion sequence. The Kalman filtering method has relatively mature theories, but it is more suitable for linear systems and requires accurate mathematical modeling of ship dynamics [4]. When the marine environment changes, the prediction accuracy will be greatly reduced. The time series method has a small calculation amount and low cost, but it is generally suitable for short-term prediction. When there is external interference, the prediction error is inevitable [5], and its prediction length and accuracy cannot meet the demand. The neural network method uses a neural network model to make a prediction, which takes historical data as input, adjusts its network structure through neural network training and learning, and then makes a prediction. In recent years, the neural network has developed rapidly. Because of its strong self-learning ability and self-adaptability, the neural network has been widely used in the study of nonlinear systems. However, due to the shortcomings of its own network initial structure parameters often set by experience, the prediction accuracy of a single neural network method still cannot meet the needs of practical application [6], [7]. The combined prediction method is to combine different methods and their respective advantages to get a new forecasting model. Compared with the single prediction method, the combined forecasting method is hoped to make a more accurate prediction. Peng *et al.* applied a particle swarm optimization algorithm to long short-term memory (LSTM) neural network [8], and many other scholars also applied the combined forecasting method to prediction problems [9]–[11]. The combined forecasting method is the hot spot of the current forecasting research. This paper combines an intelligent algorithm with a neural network and optimizes the neural network model by the intelligent algorithm to predict the ship's motion attitude. This method overcomes the shortcomings of a single neural network and is expected to achieve high prediction accuracy.

As a classic deep learning model, the recurrent neural network is widely used in the prediction research of nonlinear systems. For example, Song *et al.* solved the prediction of air quality problems by combining LSTM and Kalman [12], and Li *et al.* applied LSTM to the prediction of PM2.5 concentration [13]. BiLSTM neural network [14] is a variant of the LSTM neural network [15]. It consists of a forward LSTM network and a reverse LSTM network, and introduces the context information of time series. It can be trained on the influence of future information on the current state. Therefore, for the same time series, BiLSTM can better reflect the changing trend of time series [16]. However, BiLSTM does not solve the problem of random initialization of neural network parameters, which will affect the nonlinear learning ability. In view of this shortcoming, the existing methods to optimize the neural network are mainly to improve the error function and excitation function [17]–[19], or to optimize the initial parameters of the network by using intelligent algorithms [20]–[23]. The current intelligent algorithms for parameter optimization mainly include particle swarm optimization (PSO) algorithm [24] and ant colony optimization algorithm, but these algorithms are often prone to the problem of local optimal, which leads to the failure to find the global optimal solution in the process of parameter optimization. PSO algorithm combined with neural network has been used to solve a variety of time series prediction problems. However, other algorithms in metaheuristic algorithm such as genetic algorithm and simulated annealing algorithm are less popular in the field of ship motion attitude prediction, so this paper mainly studies the improvement and application of PSO algorithm.

Therefore, this paper proposes an improved adaptive dynamic particle swarm optimization algorithm (ADPSO), which is improved on the basis of PSO algorithm and can dynamically adjust the parameters of the algorithm to adjust the position of particles, so as to ensure that particles can find the global optimal solution. The ADPSO algorithm is used to optimize the hyperparameters of the BiLSTM neural network, and the optimized neural network model (ADPSO-BiLSTM) is used to predict the ship's motion attitude. In order to verify the predictive validity of the ADPSO-BiLSTM model proposed in this paper, the measured ship motion data was selected in the experiment and compared with the LSTM, BiLSTM, and PSO-BiLSTM neural network models. The results show that the ADPSO-BiLSTM neural network model reflects better ship motion prediction performance in ship motion attitude prediction.

The structure of this paper is as follows. The second section introduces LSTM and BiLSTM neural networks. In the third section, a particle swarm optimization algorithm is introduced. Based on the PSO algorithm, the ADPSO algorithm is proposed. The fourth section introduces the process of optimizing the BiLSTM neural network by the ADPSO algorithm. In the fifth section, the ADPSO-BiLSTM neural network model is used to predict the measured ship motion attitude and analyzed the experimental results. The sixth section concludes with some conclusions.

II. RELATED NEURAL NETWORK

A. LSTM NEURAL NETWORK

LSTM neural network was from Recurrent Neural Network (RNN) and solved the problems of the gradient disappearance of RNN. LSTM added a Memory Cell structure in the neural node of the hidden layer of RNN to store the historical information and added three gate structures, Input gate, Forget gate, and Output gate, to control the use of the historical information. By forgetting the useless information and memorizing the new information in a cell state, LSTM can transfer useful information in the subsequent time calculation. The LSTM cell structure is shown in Fig. 1.

In Fig. 1, c_t represents the current state and $\tilde{c}_t$ is the temporary state. i_t, f_t and o_t respectively represent Input gate,

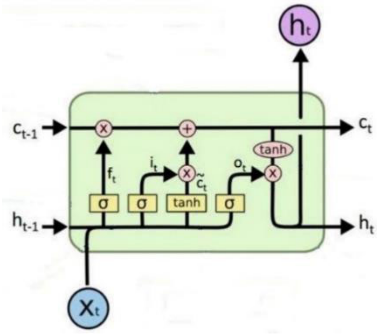

FIGURE 1. LSTM cell structure.

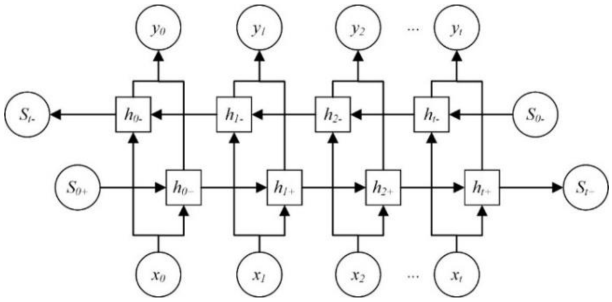

FIGURE 2. BiLSTM neural network structure.

Forget gate and Output gate, h_{t-1} is the hidden layer state of the previous time and x_t is the current input.

The calculation formulas are as follows:

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

$$\sigma f_t = \left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{2}$$

$$\sigma o_t = \left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{3}$$

$$h_t = o_t \odot tanh(c_t) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{5}$$

$$\tilde{c}_t = tanh\,(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{6}$$

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{7}$$

$$tanh(x) = \frac{\mathrm{e}^{x} - \mathrm{e}^{-x}}{\mathrm{e}^{x} + \mathrm{e}^{-x}} \tag{8}$$

where W_i, W_f and W_o respectively represent the weight of three gate connections, b is the offset, σ and tanh are the activation functions.

B. BiLSTM NEURAL NETWORK

Since LSTM can only learn the above information of time series, BiLSTM makes a further improvement on the basis of LSTM, which is composed of a forward LSTM network and a reverse LSTM network, introducing the context information of time series. BiLSTM neural network structure is shown in Fig. 2.

$x_1, x_2, \ldots, x_t$ are the input sequence, $\vec{h}_t$ and $\overleftarrow{h}_t$ are the forward and reverse outputs calculated at each moment respectively, and then the forward and reverse outputs are calculated to obtain the final output y_t. Taking the forward output $\vec{h}_t$ at time t as an example, the calculation formulas of forward and backward directions are consistent with LSTM, that is, through "(1)" to "(8)," the forward and reverse temporary cell states $\vec{\tilde{c}}_t$ and $\overleftarrow{\tilde{c}}$, input gate $\vec{l}_t$ and $\overleftarrow{l}_t$, forget gate $\vec{f}_t$ and $\overleftarrow{f}_t$, output gate $\vec{o}_t$ and $\overleftarrow{o}_t$ can be calculated respectively.

The final output y_t at time t is:

$$y_t = [\vec{h}_t, \overleftarrow{h}_t] \tag{9}$$

From the above equation, we can calculate the output at each moment, and then get the final output $Y = [h_0, h_1, \ldots, h_t]$.

III. PROPOSED METHOD

A. PSO ALGORITHM

PSO algorithm is a group intelligence optimization algorithm, which was first proposed by Kennedy and Eberhart in 1995 to simulate the foraging behavior of birds. Each particle in the particle swarm represents a possible solution. Within a given search range, the particle adjusts its speed and position by comparing individual extremum and global extremum for optimization until it finds the optimal solution satisfying the termination conditions. The formula is as follows:

$$V_{i,j}^{t+1} = \omega V_{i,j}^{t} + c_1 r_1 \left(y_{i,j}^{t} - x_{i,j}^{t}\right) + c_2 r_2 \left(\hat{y}_{i}^{t} - x_{i,j}^{t}\right) \tag{10}$$

$$X_{i,j}^{t+1} = X_{i,j}^{t} + V_{i,j}^{t+1} \tag{11}$$

t is the number of iterations, $V_{i,j}^{t}$, $x_{i,j}^{t}$ are the velocity and position of particle i in the j dimensional at the time of iteration t, ω is the inertia weight, c_1 and c_2 are learning factors, $y_{i,j}^{t}$ is the individual extremum point in the t-th iteration of the particle swarm, $\hat{y}_{i}^{t}$ is the global extremum of the particle swarm, r_1 and r_2 are random numbers uniformly distributed in the interval $[0, 1]$, $V_{i,j}^{t} \in [-V_{max}, V_{max}]$, V_{max} is the set constant.

Because of its simple operation and fast convergence speed, PSO has been widely used in many fields such as function optimization and image processing. However, the PSO algorithm has problems such as premature convergence and prone to fall into local extremum. For the problem of PSO, many papers have proposed a lot of methods to optimize PSO [25]–[28], such as introducing new parameters, optimizing initial parameters and search strategy, combining PSO with other algorithms and so on. Based on this, this paper proposes a new strategy of position update and parameter adjustment to improve the performance of the PSO algorithm.

B. ADPSO ALGORITHM

ADPSO is a dynamic particle swarm optimization algorithm proposed in this paper that can adjust parameters adaptively. This algorithm combines conventional particle swarm optimization algorithm and dynamic spatial search strategy, making the position oscillation process of particles in the iterative search process more active, and solving the problem that the PSO algorithm is easy to fall into the local extreme value. In addition, adaptive adjustment of learning factor parameters can be realized to balance local search ability and global search ability.

1) DYNAMIC

In the PSO algorithm, the current position of the particle is mainly associated with two factors. one is the previous position of the particle, and the other is the particle's current velocity. For particle search, it belongs to a dynamic nonlinear process, and there is no fixed trajectory. Therefore, in order to further improve particle search activity and further improve the possibility of finding the best position, the formula of particle position is optimized and adjusted in this paper, specifically as follows:

$$V_{i,j}^{t+1} = \omega V_{i,j}^{t} + c_1 r_1 \left(y_{i,j}^{t} - x_{i,j}^{t}\right) + c_2 r_2 \left(\hat{y}_i^t - x_{i,j}^{t}\right) \tag{12}$$

$$X_{i,j}^{t+1} = X_{i,j}^{t} + V_{i,j}^{t+1} \times ln(e^{\frac{t}{\mathrm{t_{max}}}} - 1) \tag{13}$$

where $\mathrm{t_{max}}$ is the maximum number of iterations, $t \in [-\mathrm{t_{max}}, \mathrm{t_{max}}]$. In the above equation, the logarithmic function is introduced to make the position of each particle oscillate, dynamically adjust the search process of the particle, and effectively guide the particle to find the optimal solution.

During particle operation, when a particle is out of the search range, the boundary position value is usually assigned to the particle, and after this processing, all the out-bound particles will gather at the boundary, at which point if there is a local optimal in the boundary area, the particles will easily fall into the local extreme point and unable to find global optimal solution, thus affect the global search ability of the algorithm. In this paper, the algorithm is improved as follows for the case of particle out of bounds:

When $X_{i,j}^{t+1} > \mathrm{X_{max}}$:

$$X_{i,j}^{t+1} = \mathrm{X_{max}} * \left(1 - \frac{rand\left(X_{i,j}^{t}, \mathrm{X_{max}}\right)}{\mathrm{X_{max}}}\right) \tag{14}$$

When $X_{i,j}^{t+1} < -\mathrm{X_{max}}$:

$$X_{i,j}^{t+1} = -\mathrm{X_{max}} * \left(1 + \frac{rand\left(-\mathrm{X_{max}}, X_{i,j}^{t}\right)}{\mathrm{X_{max}}}\right) \tag{15}$$

After the operation of the above formula, the out-of-bounds particles will not gather in the boundary region but distribute near the boundary, which not only makes the particles in the feasible space, but also overcomes the defect that the algorithm is easy to fall into the local optimal at the boundary, and improves the global search ability of the algorithm to some extent.

2) ADJUSTMENT OF INERTIAL WEIGHT

The inertia weight ω has a great impact on the performance of the PSO algorithm. Since the search state of particles in different periods should be different, strong search ability is needed in the early stage to quickly search for the best, and fine selection is required in the late stage to ensure the search accuracy:

$$\omega^t = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \frac{\mathrm{t_{max}} - t}{\mathrm{t_{max}} - 1} \tag{16}$$

where $\omega_{\max}$ and $\omega_{\min}$ are the upper and lower limits of the preset inertia weight. In this paper, $\omega_{\max}$ is set to 3 and $\omega_{\min}$ is set to 1, t and $\mathrm{t_{max}}$ have the same meanings as in "(11)."

It can be seen from "(16)" above that, in the early stage of particle swarm iteration, ω is larger and more inclined to global search, which is convenient for rapid search, while in the late stage, ω is smaller, which is suitable for local search, and particle swarm can search more carefully, which is conducive to particle search for the optimal solution.

3) ADAPTIVE LEARNING FACTOR

In the conventional PSO algorithm, $\mathrm{c_1}$ and $\mathrm{c_2}$ are the fixed values set. This paper proposes a method that the learning factor can be adaptively adjusted with the change of inertia weight, so as to adjust the self-learning ability and group learning ability required by particles in different search periods, and then adjust the performance of the particle in finding the optimal solution. Specific adjustments are as follows:

$$c_1 = \mathrm{c_{start}} + (\mathrm{c_{start}} - \mathrm{c_{end}}) \cos(e^{2-\omega} - 1) \tag{17}$$

$$c_2 = \mathrm{c_{start}} - (\mathrm{c_{start}} - \mathrm{c_{end}}) \cos(e^{2-\omega} - 1) \tag{18}$$

where $\mathrm{c_{start}}$ represents the initial value of the learning factor c_1 and c_2, $\mathrm{c_{end}}$ is the termination value, and ω represents the inertia weight. As can be seen from the above two equations, when the initial ω decreases from 3 to 2, $\cos(e^{2-\omega}-1)$ increases, c_1 also increases gradually, while c_2 decreases gradually, indicating that the social information communication ability of the particle is weak and the self-adjustment ability is relatively strong. At the later stage of the search, ω decreases from 2 to 1, and the $\cos(e^{2-\omega}-1)$ decreases gradually, so $\mathrm{c_1}$ decreases and $\mathrm{c_2}$ increases, indicating that the self-learning ability of the particle is gradually weakened and the social learning ability is enhanced.

4) TIME COMPLEXITY ANALYSIS

Similar to the original particle swarm optimization algorithm, each iteration of particle optimization must go through the steps of fitness function value calculation, particle individual and population global optimal solution selection, particle velocity and position update. Assuming that the maximum number of iterations is L, the particle swarm size is M, and the problem size is N, the time complexity of the algorithm can be expressed as:

$$\begin{aligned} O(L, M, N) &= L * \Big(O(M * N) + O(1) + O(M) \\ &\quad + O\left(2M * N^2 + M * N^2\right)\Big) \\ &= L * M * O(N^2) \end{aligned} \tag{19}$$

IV. SYSTEM MODELING

A. ADPSO OPTIMIZE BiLSTM NEURAL NETWORK

The main idea of this paper is to use the ADPSO algorithm to optimize the hyperparameters of the BiLSTM neural network, find the optimal hidden layer nodes of the BiLSTM neural network through the ADPSO algorithm, and then build a better neural network model to predict the ship motion attitude. The flowchart of ADPSO optimizing BiLSTM hyperparameters is shown in Fig. 3.

The specific steps for ADPSO to optimize BiLSTM parameters are as follows:

1) Initialize particle swarm parameters and BiLSTM network structure. Particle swarm parameters mainly include the particle swarm size, number of iterations, learning factors, and the value range of particle location and velocity. The initialization of the BiLSTM network structure mainly refers to the determination of the number of hidden layers in the network.
2) Determine the evaluation function of the particles. The fitness function of particles is defined as:

$$fit_{MSE} = \frac{1}{\text{n}} \sum_{i=1}^{\text{n}} (\hat{y}_i - y_i)^2 \tag{20}$$

In the formula, n represents the particle swarm size, Y_i is the predicted value, and y_i is the actual value.
3) Calculate the fitness value of each particle according to the above formula.
4) Update the individual optimal position of the particles and the global optimal position of the particle swarm, and update the speed and position of the particles themselves.
5) Determine whether the iteration end condition is reached (maximum number of iterations or a good enough position), if it is reached, it ends, otherwise, go to step 3 to continue iteration.

B. PREDICTION PROCESS OF SHIP MOTION ATTITUDE BASED ON ADPSO-BiLSTM

This article is based on the ADPSO-BiLSTM network model to predict the measured ship motion attitude data. The specific process of prediction is shown in Fig. 4.

Firstly, collect the measured ship motion data and preprocess it. Select part of the motion data as the training set and part as the test set. Input the training set and use the ADPSO algorithm to optimize the hyperparameters of the hidden layer nodes of the BiLSTM neural network. Enter the test data to make a prediction and get the prediction result.

V. SIMULATION RESULTS AND ANALYSIS

A. ADPSO ALGORITHM PERFORMANCE ANALYSIS

The ADPSO algorithm proposed in this paper is mainly for the problem that the PSO algorithm is easy to fall into local extremes. Therefore, the commonly used test functions will be selected for testing and compared with the PSO algorithm and QPSO (Quantum Particle Swarm) algorithm [29]. Table 1 shows the selected test functions.

In this paper, the particle swarm size of each algorithm is set as 40, and the maximum number of iterations is 5000, with each algorithm running 100 times. Fig. 5, Fig. 7, Fig. 9 and Fig. 11 are three-dimensional graphs of the test functions, Fig. 6, Fig. 8, Fig. 10 and Fig. 12 are representative pictures of the selected operation results, which are respectively comparison graphs of the optimization iteration results of different algorithms for the test function.

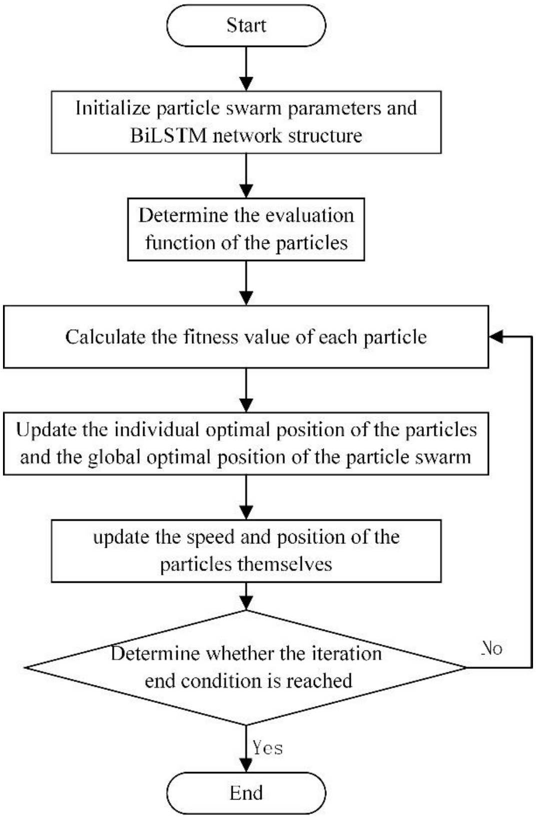
Start
Initialize particle swarm parameters and BiLSTM network structure
Determine the evaluation function of the particles
Calculate the fitness value of each particle
Update the individual optimal position of the particles and the global optimal position of the particle swarm
update the speed and position of the particles themselves
Determine whether the iteration end condition is reached
No
Yes
End

FIGURE 3. Parameter optimization flowchart.

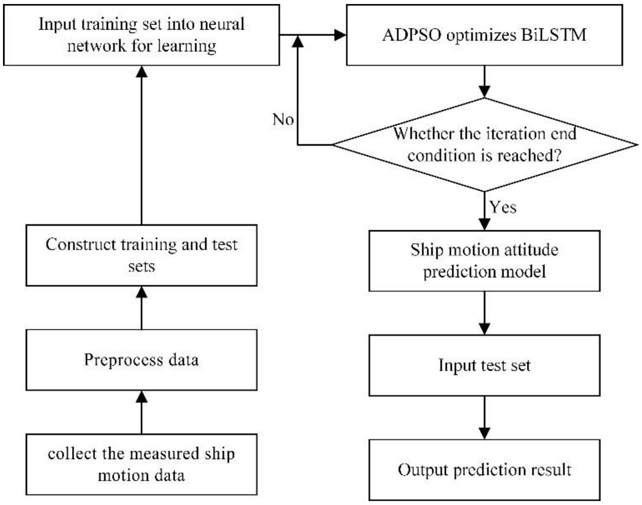
Input training set into neural network for learning
ADPSO optimizes BiLSTM
No
Whether the iteration end condition is reached?
Yes
Construct training and test sets
Ship motion attitude prediction model
Preprocess data
Input test set
collect the measured ship motion data
Output prediction result

FIGURE 4. Prediction process.

Fig. 5 shows that the Holder table test function is a multi-peak function with many local extremum points. Experimental results in Fig. 6 show that the PSO algorithm often falls into local extremum, while the other two algorithms quickly find the global optimal value.

TABLE 1. Test functions information.

Name	Test Function	Dimension	Range	Optimal Value
Holder table	$f_1(x) = -\lvert \sin(x_1)\cos(x_2)\, exp\left(\left\lvert 1-\frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert\right)\rvert$	2	[−10,10]	-19.2085
Rastrign	$f_2(x) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$	30	[−5.12,5.12]	0
Beale	$f_3(x) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$	2	[−4.5,4.5]	0
Sphere	$f_4(x) = \sum_{i=1}^{D} x_i^2$	20	[−5.12,5.12]	0

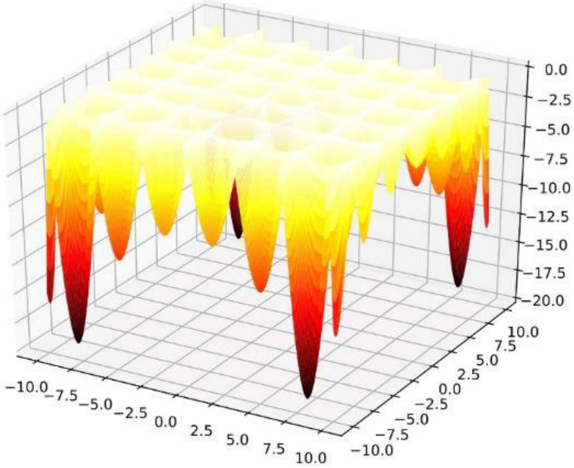

FIGURE 5. Three-dimensional graph of the Holder table test function.

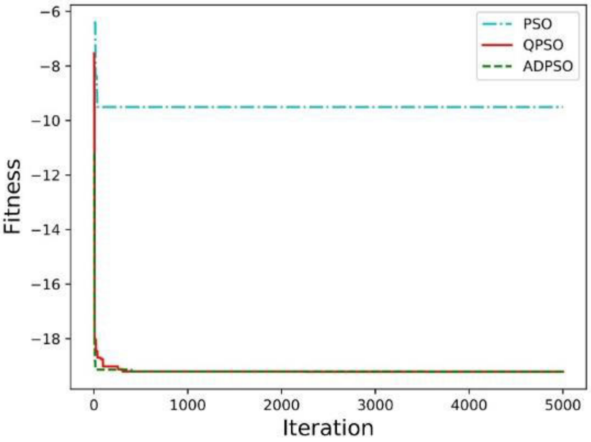

FIGURE 6. Optimization iteration results of the Holder table test function.

Fig. 7 shows that the Rastrign test function is also a multi-peak function. It can be seen from Fig. 8 that the PSO algorithm and the QPSO algorithm fall into local extreme points in the process of optimization.

The Beale test function is a unimodal function with a global minimum of 0. Compared with Holder table and Rastrign test functions, the PSO algorithm can find the global optimal solution in the Beale test function in a better way, but sometimes it still falls into the local value.

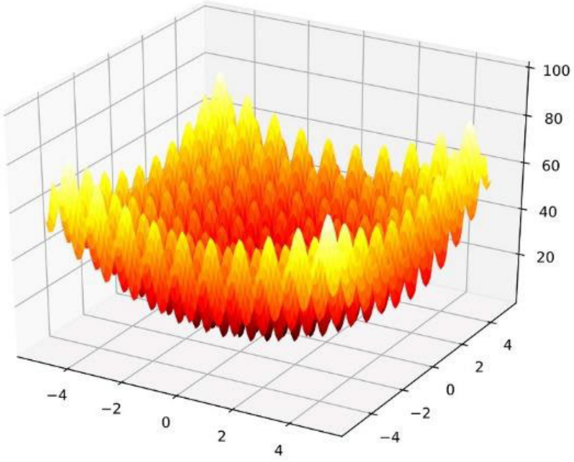

FIGURE 7. Three-dimensional graph of the Rastrign test function.

From the formula in Table 1 and three-dimensional diagram of the sphere test function, we can easily see that the Sphere test function has only one global minimum value of 0. Compared with the other two algorithms, ADPSO finds the global optimal solution quickly.

It can be seen from Fig. 6, Fig. 8, Fig. 10, and Fig. 12 that the PSO algorithm is prone to fall into the local extreme value in the test functions. Among the above four single-peak and multi-peak test functions, compared with the other algorithms, ADPSO shows a good optimization ability and quickly finds the global optimal value. The ADPSO algorithm finds the global optimal solution in the four test functions in the least number of iterations, and all of them do not exceed 200 times.

Table 2 shows the stability of the three algorithms, that is, the success rate of the algorithm in finding the optimal value of the test function (the percentage (%) of finding the optimal value).

TABLE 2. **Comparison of algorithm stability.**

Global optimization success rate / (%)	PSO	QPSO	ADPSO
f_1	68	100	100
f_2	76	84	100
f_3	97	100	100
f_4	88	92	100

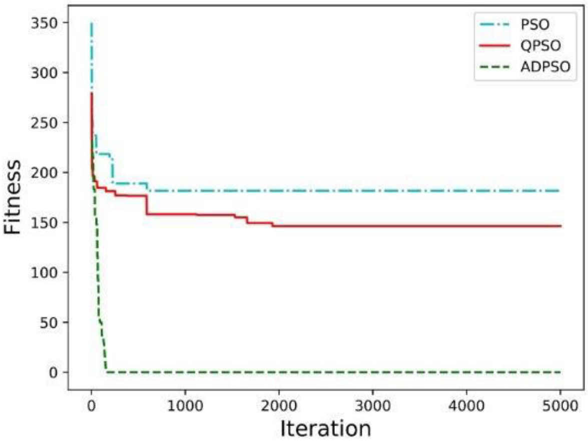

FIGURE 8. **Optimization iteration results of the Rastrign test function.**

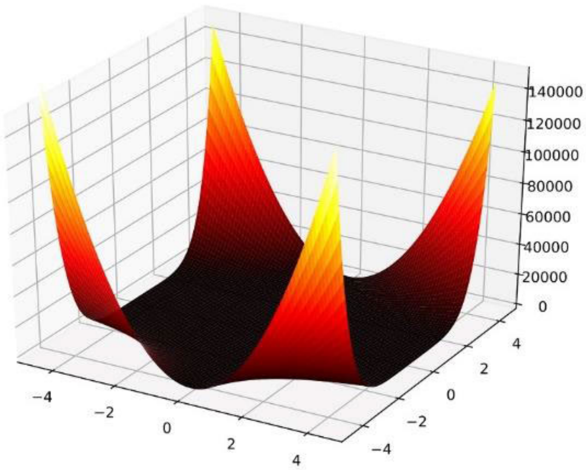

FIGURE 9. **Three-dimensional graph of the Beale test function.**

The stability of the algorithm is very important. If the PSO particle swarm algorithm is used to optimize the parameters of the neural network, there will be a certain probability of falling into the local extreme point during the optimization process, which will affect the accuracy of the prediction experiment. The above pictures and tables prove that ADPSO solves the problem that is easy to fall into a local optimum, obtains a higher iteration success rate, and has good stability. ADPSO achieved a good global optimization success rate in all test functions. It can be seen that the improved algorithm reduces the probability of falling into the local optimal value by up to 34 percentage points, which proves that the improved algorithm is significantly more robust.

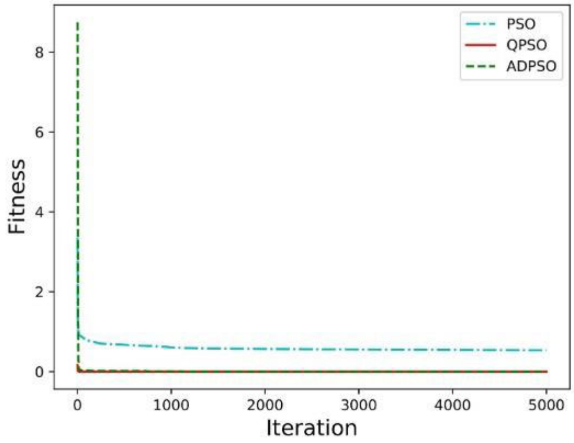

FIGURE 10. **Optimization iteration results of the Beale test function.**

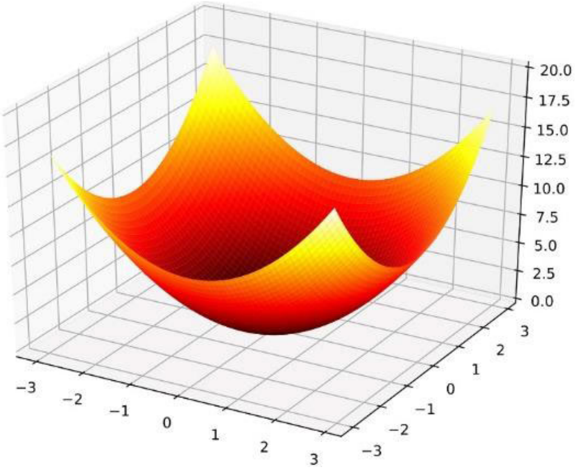

FIGURE 11. **Three-dimensional graph of the Sphere test function.**

B. EXPERIMENT OF SHIP MOTION ATTITUDE PREDICTION

This article collects the measured motion data of a ship, and mainly predicts the heave displacement, roll angle and pitch angle in the motion attitude. In order to verify the effectiveness of the BiLSTM neural network optimized by the ADPSO algorithm (ADPSO-BiLSTM), the paper uses LSTM neural network, BiLSTM neural network, and BiLSTM network optimized by standard PSO algorithm (PSO-BiLSTM) for comparison experiments. In order to ensure the fairness of the experiment, the parameters of the four models compared in this paper are as consistent as possible. For example, the network structure parameters of LSTM and BiLSTM are consistent, and the parameters of PSO and ADPSO algorithms are also of the same value. The specific parameter setting is as follows:

1) Particle swarm size. In this paper, the number of particles is finally set to be 40 after the actual simulation experiment on different particle numbers.
2) Particle swarm dimension. In this paper, the particle swarm optimization algorithm is used to find the optimal number of nodes in the hidden layer of the neural network, so the particle swarm dimension corresponds to the number of hidden layers. Corresponding to the number of hidden layers of the neural network, this paper sets the particle swarm dimension to 2 when constructing the actual experimental model, which is to find the optimal number of hidden layer nodes of the two hidden layers.
3) Initial particle position, initial velocity, velocity range and position range. In this paper, the range of particle positions is set to 1 to 30, the maximum speed of particles is set to 1, and the initial position and speed of particles are set to random values within the range.
4) Inertia weight and learning factor parameters. The parameter settings in the two algorithms are different. The parameters in the PSO algorithm are fixed, while the parameters in the ADPSO algorithm change with the number of iterations. In the PSO algorithm, the weight parameter ω is set to 1, and the learning factors c_1 and c_2 take the value 2. In the ADPSO algorithm, $\omega_{\max}$ is equal to 3, $\omega_{\min}$ is 1, the initial learning factor c_{start} is set to 2, and c_{end} is set to 1.
5) Maximum number of iterations. In this paper, the maximum number of iterations is set to 100.
6) Fitness function. The fitness functions of both algorithms are set as the mean square error function.

Before the experiment, the collected ship motion data is normalized, such as "(21)." The ship motion data collected in this paper is actually measured when the ship is sailing at sea. The heave displacement is measured by a laser ranging sensor. The roll and pitch angles are collected by an electronic inclinometer. The sampling time of the data set exceeds 1000s, the collection interval is 0.176s, the collection frequency is 6 times per second, and there are more than 6000 sets of data in total. This paper sets 98% of the experimental data as the training set and 2% as the test set.

$$m_i = \frac{x_i - \mathrm{x}_{\min}}{\mathrm{x}_{\max} - \mathrm{x}_{\min}} \tag{21}$$

In the above formula, x_i is the input data ($i = 1, 2, \ldots n$), $\mathrm{x}_{\max}$ and $\mathrm{x}_{\min}$ are the maximum and minimum values in the data, and m_i is the normalized data.

This paper mainly selects RMSE (Root Mean Square Error) and MAPE (Mean Absolute Percentage Errors) as the evaluation indicators of the prediction results. The calculation formulas are as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{\mathrm{n}} \left(\hat{y}_i - y_i\right)^2}{\mathrm{n}}} \tag{22}$$

$$MAPE = \frac{1}{\mathrm{n}} \sum_{i=1}^{\mathrm{n}} \left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{23}$$

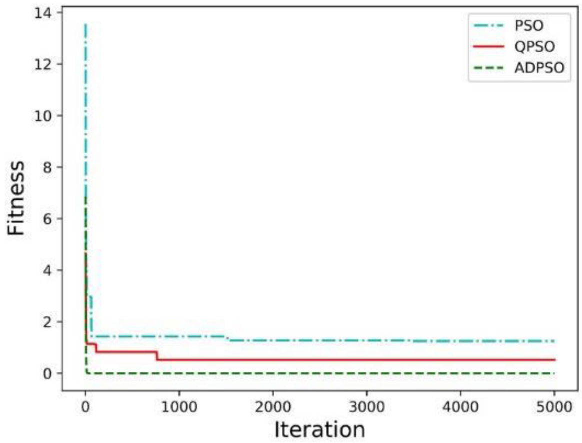

FIGURE 12. Optimization iteration results of the Sphere test function.

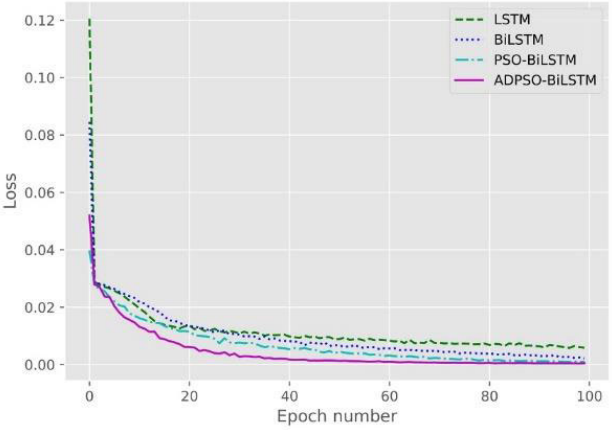

FIGURE 13. Iterative results of loss functions for four models.

where $\hat{y}_i$ is the predicted value, y_i is the actual value, and n is the number of predictions. The smaller the value of the evaluation index in the above formula, the more accurate the prediction result and the better the prediction effect.

1) HEAVE DISPLACEMENT PREDICTION

Four neural network models were used to predict the ship heave displacement. Fig. 13 shows the iterative results of the loss function of the four models. Fig. 14 and Fig. 15 show the comparison between the heave displacement prediction curve and the true curve. Fig. 16 shows the comparison between the prediction errors. Table 3 shows the two evaluation index values of the prediction results of the four models.

It can be seen from Fig. 13 that the loss functions of the four models decrease with the number of iterations while training the models, and the performance of the ADPSO-BiLSTM model is the most prominent.

Fig. 14 and Fig. 15 show the comparison between the predicted value of the heave displacement and the true value. The data was normalized before the experiment, so the predicted value is also the normalized value, and then the real predicted value was obtained after the denormalization. These

TABLE 3. Evaluation indexes of heave displacement prediction results.

	LSTM	BiLSTM	PSO-BiLSTM	ADPSO-BiLSTM
RMSE	0.162489	0.087512	0.066444	0.049804
MAPE	0.483738	0.276900	0.228500	0.183593

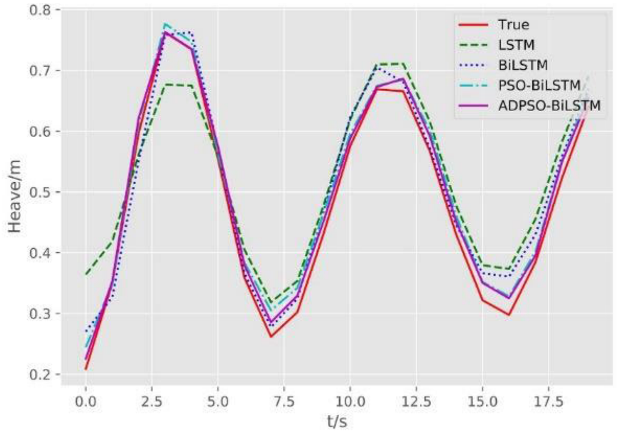

FIGURE 14. Heave displacement prediction results (normalized data).

two figures are the normalized predicted values and the real predicted values respectively. The figure clearly shows the performance of the four models in the prediction of the ship's heave displacement. They all maintain the same trend with the real movement, but by comparison, the LSTM is quite different from the true value, while the predicted value of ADPSO-BiLSTM model is the best fit with the true value.

The vertical coordinate of Fig. 16 is the difference between the true value and the predicted value. From Fig. 16, the prediction accuracy of the four models can be seen more clearly. Except for the LSTM model, the other three models predicted the heave displacement with small error.

The different colored lines in Fig. 13 to Fig. 16 represent the predictive performance of the four models. Fig. 13 is a comparison of the loss function changing with the number of iterations in the four models. We observe that ADPSO-BiLSTM has the fastest convergence speed, and the final result is very close to zero. Fig. 14 and Fig. 15 show the prediction results of the heave displacement obtained in the four models. Fig. 14 uses the predicted normalized standard data, and Fig. 15 is the real data obtained after denormalization. Fig. 16 is a comparison chart of the error between the predicted value and the true value. Table 3 shows the evaluation index values of the prediction results of the four models. The above results show that the predicted values of the four models have fluctuated over time, but the ADPSO-BiLSTM neural network model proposed in this paper has fewer errors than the other three models, the value of the obtained evaluation index is smaller, and it fits better with the real value, and has better prediction results. Compared with the LSTM, ADPSO-BiLSTM on index RMSE and MAPE respectively decreased 0.112685 and 0.300145.

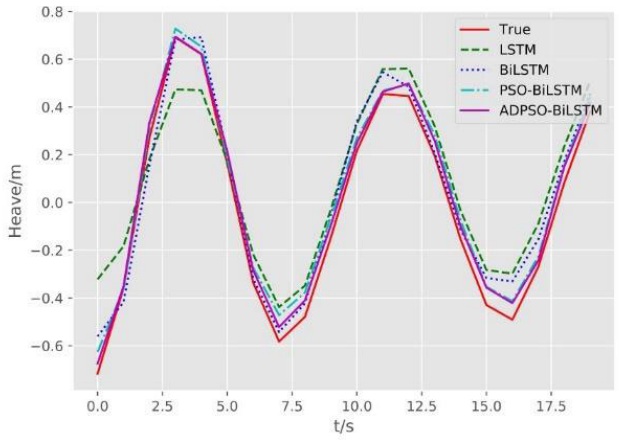

FIGURE 15. Heave displacement prediction results (real data).

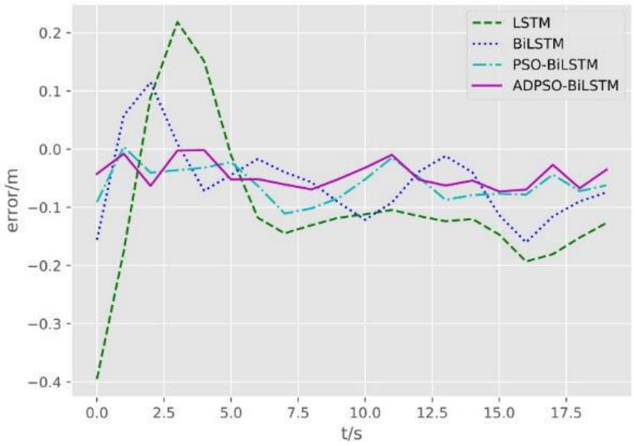

FIGURE 16. Heave displacement prediction error.

2) ROLL ANGLE PREDICTION

Four neural network models were used to predict the ship roll angle data. Fig. 17 shows the iterative results of the loss function of the four models, Fig. 18 and Fig. 19 are comparison charts of the roll angle prediction curve and the true curve, and Fig. 20 is a comparison chart of the prediction error. Table 4 shows the four evaluation index values of the prediction results of the four models.

Fig. 17 shows the change process of the loss functions of the four models when training the model, and the ADPSO-BiLSTM model has the fastest convergence when training.

Compared with heave displacement, roll motion requires prediction angle, and the maximum roll angle in the dataset can reach 10 degree, so roll motion has a greater impact on ship performance. As can be seen from Fig. 18 and Fig. 19, the roll angle prediction of the four models is consistent with the trend of real roll motion. Compared with the other three models, the ADPSO-BiLSTM model has a smaller gap

TABLE 4. Evaluation indexes of rolling angle prediction results.

	LSTM	BiLSTM	PSO-BiLSTM	ADPSO-BiLSTM
RMSE	0.882616	0.810308	0.560505	0.478694
MAPE	0.853155	0.796305	0.630343	0.433547

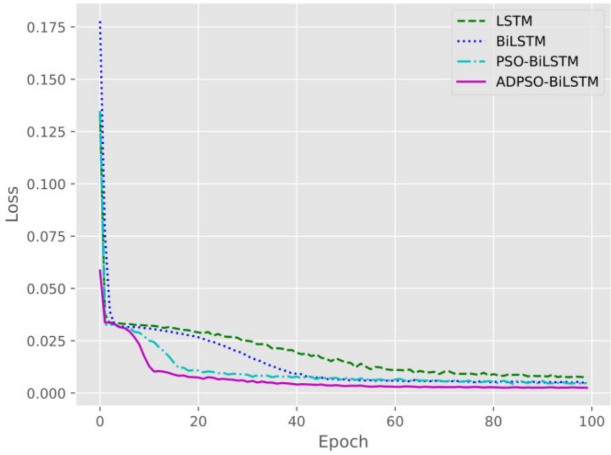

FIGURE 17. Iterative results of loss functions for four models.

between the predicted value and the true value within the predicted duration.

Fig. 17 shows that with the increase of iteration times, the loss function values of the four models tend to be stable, while ADPSO-BiLSTM converges the fastest. As can be seen from Fig. 18, Fig. 19 and Fig. 20, among the four models, ADPSO-BiLSTM neural network model has a fewer prediction error and better prediction accuracy, while the LSTM model has the biggest gap with the real value. The data in Table 4 shows that from LSTM to ADPSO-BiLSTM, the index values predicted by the four models are getting smaller and smaller, indicating that the prediction accuracy is gradually increasing. Compared with the LSTM, ADPSO-BiLSTM on index RMSE and MAPE decreased 0.403922, 0.419608, almost half as much.

3) PITCH ANGLE PREDICTION

Four network models were used to predict the ship pitch angle data. Fig. 21 shows the iterative results of the loss function in the four models, Fig. 22 and Fig. 23 are comparison diagrams of the pitch angle prediction curve and the true curve, and Fig. 24 is a comparison diagram of the prediction error. Table 5 shows the four evaluation index values of the prediction results of the four models.

In the prediction of pitch motion, Fig. 17 shows the change process of the loss function of the four models during the training of the model. It shows that the model optimized by the intelligent algorithm can achieve less loss function value than the other two neural network models and reach a stable state faster.

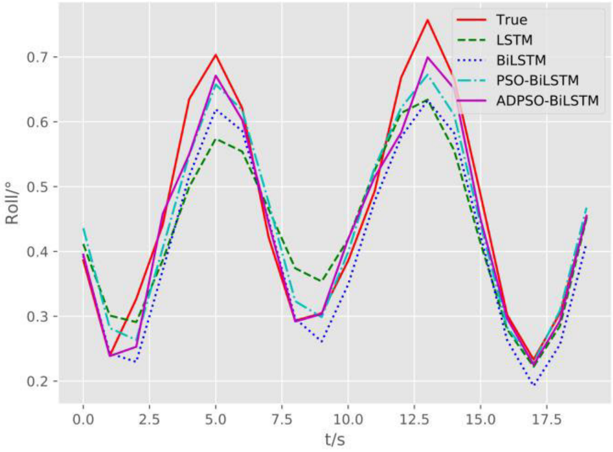

FIGURE 18. Rolling angle prediction result (normalized data).

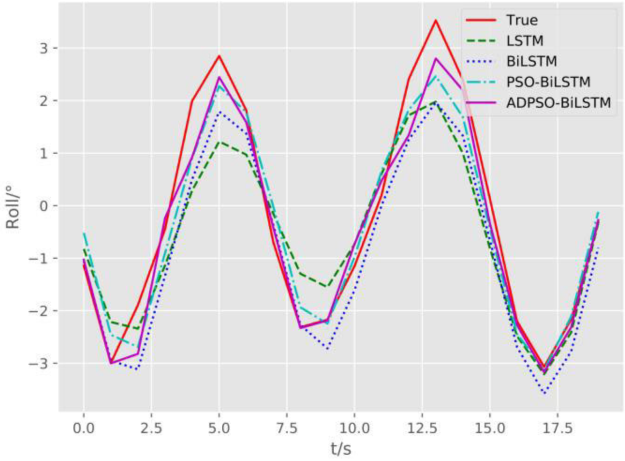

FIGURE 19. Rolling angle prediction result (real data).

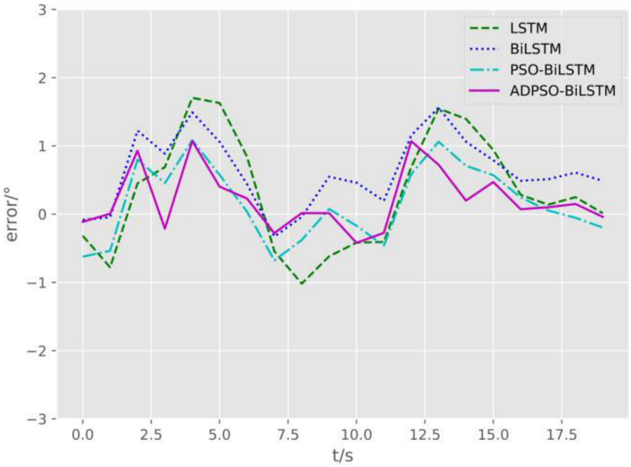

FIGURE 20. Rolling angle prediction error.

As can be seen from Fig. 22 and Fig. 23, the amplitude of pitch is significantly smaller than that of roll. The pitch angle prediction of ADPSO-BiLSTM model is better fitted

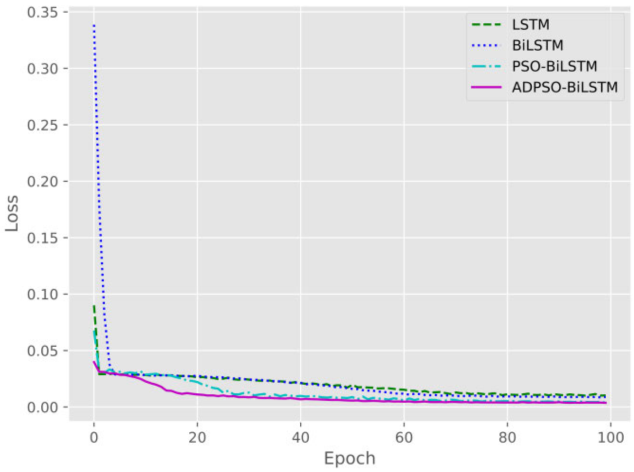

FIGURE 21. Iterative results of loss functions for four models.

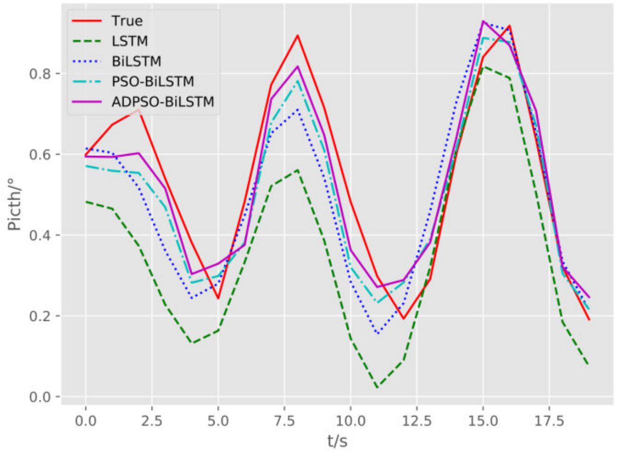

FIGURE 22. Pitch angle prediction result (normalized data).

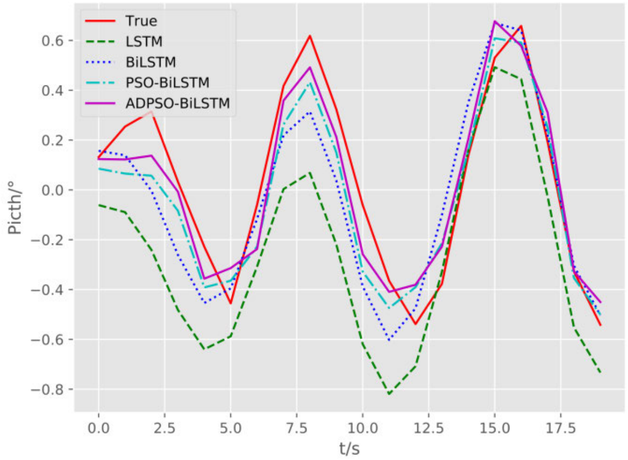

FIGURE 23. Pitch angle prediction result (real data).

to the real pitch motion curve, which also proves the effectiveness of ADPSO algorithm to optimize neural network parameters.

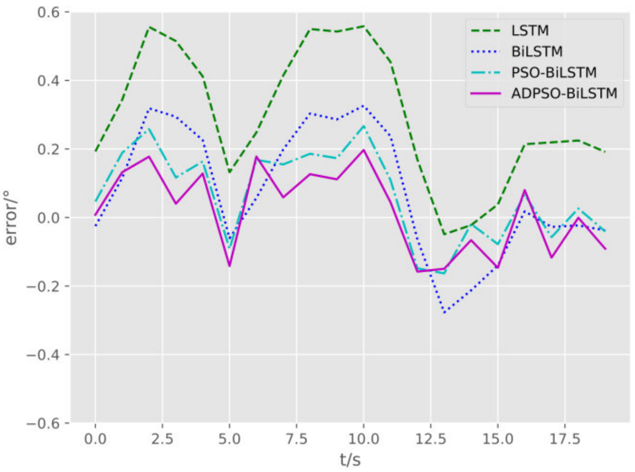

FIGURE 24. Pitch angle prediction error.

TABLE 5. Evaluation indexes of the pitch angle prediction results.

	LSTM	BiLSTM	PSO-BiLSTM	ADPSO-BiLSTM
RMSE	0.352050	0.198043	0.144586	0.121107
MAPE	2.170007	1.167142	0.829227	0.630618

The above results show that in the prediction of pitch angle, ADPSO-BiLSTM and PSO-BiLSTM perform better than the other two models. Fig. 24 shows that the LSTM neural network model has the largest prediction error. Moreover, it can be seen from Fig. 21 that the two network models optimized by the algorithm converge faster and the value obtained by the loss function is small. Compared with the other three prediction models, ADPSO-BiLSTM neural network model has fewer prediction errors and better performance in terms of prediction accuracy. Compared with the LSTM, ADPSO-BiLSTM on index RMSE and MAPE reduced almost two thirds, 0.230943 and 1.539389 respectively.

From the prediction results of the four models, it can be seen that the ADPSO algorithm improves the global search capability of the particle swarm algorithm. In the process of optimizing the hyperparameters of the BiLSTM neural network model, it can generate the parameters with higher accuracy and more optimized models make the ADPSO-BiLSTM network model have higher prediction performance in the prediction of ship's heave displacement, roll angle and pitch angle. The results show that the ADPSO-BiLSTM neural network model has a good prediction performance for the ship motion data set, but its prediction effect in other fields needs to be further verified.

VI. CONCLUSION

In this paper, an improved particle swarm optimization algorithm, ADPSO, is proposed, which can adjust parameters adaptively and dynamically. The ADPSO algorithm is used to optimize the number of hidden nodes of the BiLSTM neural network, and an optimized ship motion attitude prediction model is obtained, which can effectively improve the prediction accuracy of the model in ship motion attitude prediction. The experimental results show that compared with LSTM, BiLSTM, and PSO-BiLSTM neural network models, ADPSO-BiLSTM neural network model can better fit the data in the ship motion attitude prediction and has better prediction performance.

REFERENCES

[1] Q. U. Hao, R. Guo, and X. Ding, "Design of the deck longitudinal motion compensation for carrier landing," *Aeronaut. Sci. Technol.*, vol. 27, no. 12, pp. 13–17, 2016.

[2] P. Kaplan, "A study of prediction techniques for aircraft carrier motions at sea," *J. Hydronautics*, vol. 3, no. 3, pp. 121–131, Jul. 1969.

[3] B. Zeng and S. Liu, "A self-adaptive intelligence gray prediction model with the optimal fractional order accumulating operator and its application," *Math. Methods Appl. Sci.*, vol. 40, no. 18, pp. 7843–7857, Dec. 2017.

[4] U. D. Nielsen, A. H. Brodtkorb, and J. J. Jensen, "Response predictions using the observed autocorrelation function," *Mar. Struct.*, vol. 58, pp. 31–52, Mar. 2018.

[5] I. Yumori, "Real time prediction of ship response to ocean waves using time series analysis," in *Proc. OCEANS*, 1981, pp. 1082–1089.

[6] G. Yang, Q. M. Jie, and N. Q. Tao, "Prediction of ship motion attitude based on BP network," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 1596–1600.

[7] G. D. Wang, B. Han, and W. Y. Sun, "Short-term prediction of ship motion based on LSTM," *Ship Sci. Technol.*, vol. 39, no. 13, pp. 69–72, 2017.

[8] X. Peng, B. Zhang, and H. Zhou, "An improved particle swarm optimization algorithm applied to long short-term memory neural network for ship motion attitude prediction," *Trans. Inst. Meas. Control*, vol. 41, no. 15, pp. 4462–4471, Nov. 2019.

[9] Z. Wang and Y. Lou, "Hydrological time series forecast model based on wavelet de-noising and ARIMA-LSTM," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Mar. 2019, pp. 1697–1701.

[10] S. S. Vaitheeswaran and V. R. Ventrapragada, "Wind power pattern prediction in time series measuremnt data for wind energy prediction modelling using LSTM-GA networks," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–5.

[11] Z. Tan and P. Pan, "Network fault prediction based on CNN-LSTM hybrid neural network," in *Proc. Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE)*, Jul. 2019, pp. 486–490.

[12] X. Song, J. Huang, and D. Song, "Air quality prediction based on LSTM-Kalmanmodel," in *Proc. IEEE 8th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, May 2019, pp. 695–699.

[13] T. Li, M. Hua, and X. Wu, "A hybrid CNN-LSTM model for forecasting particulate matter (PM2.5)," *IEEE Access*, vol. 8, pp. 26933–26940, 2020.

[14] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] P. Hu, J. Tong, J. Wang, Y. Yang, and L. D. Oliveira Turci, "A hybrid model based on CNN and bi-LSTM for urban water demand prediction," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2019, pp. 1088–1094.

[17] J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu, "Deep recurrent models with fast-forward connections for neural machine translation," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 371–383, Dec. 2016.

[18] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks," Jul. 2017, *arXiv:1707.06799*. [Online]. Available: http://arxiv.org/abs/1707.06799

[19] R. Ghaeini, S. A. Hasan, V. Datla, J. Liu, K. Lee, A. Qadir, Y. Ling, A. Prakash, X. Z. Fern, and O. Farri, "DR-BiLSTM: Dependent reading bidirectional LSTM for natural language inference," Feb. 2018, *arXiv:1802.05577*. [Online]. Available: http://arxiv.org/abs/1802.05577

[20] H. Moalla, W. Elloumi, and A. M. Alimi, "H-PSO-LSTM: Hybrid LSTM trained by PSO for online handwriter identification," in *Proc. ICONIP*. Cham, Switzerland: Springer, 2017, pp. 41–50.

[21] G. U. Xingjian, L. Zhao, M. Jin, Y. Liu, and C. Liu, "Research on short-term prediction of typical trial sea environment in China based on LSTM neural network," *Shipbuilding China*, vol. 58, no. 4, pp. 100–107, 2017.

[22] L. Troiano, E. M. Villa, and V. Loia, "Replicating a trading strategy by means of LSTM for financial industry applications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3226–3234, Jul. 2018.

[23] A. ElSaid, F. El Jamiy, J. Higgins, B. Wild, and T. Desell, "Optimizing long short-term memory recurrent neural networks using ant colony optimization to predict turbine engine vibration," *Appl. Soft Comput.*, vol. 73, pp. 969–991, Dec. 2018.

[24] Y. Yao, L. Han, and J. Wang, "LSTM-PSO: Long short-term memory ship motion prediction based on particle swarm optimization," in *Proc. IEEE CSAA Guid., Navigat. Control Conf. (CGNCC)*, Aug. 2018, pp. 1–5.

[25] T. Zhongda, L. Shujiang, W. Yanhong, and W. Xiangdong, "SVM predictive control for calcination zone temperature in lime rotary kiln with improved PSO algorithm," *Trans. Inst. Meas. Control*, vol. 40, no. 10, pp. 3134–3146, Jun. 2018.

[26] H. Wang, Z. Cui, H. Sun, S. Rahnamayan, and X.-S. Yang, "Randomly attracted firefly algorithm with neighborhood search and dynamic parameter adjustment mechanism," *Soft Comput.*, vol. 21, no. 18, pp. 5325–5339, Sep. 2017.

[27] M. Ghasemi, J. Aghaei, and M. Hadipour, "New self-organising hierarchical PSO with jumping time-varying acceleration coefficients," *Electron. Lett.*, vol. 53, no. 20, pp. 1360–1362, Sep. 2017.

[28] M. S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, and G. Pasi, "Fuzzy self-tuning PSO: A settings-free algorithm for global optimization," *Swarm Evol. Comput.*, vol. 39, pp. 70–85, Apr. 2018.

[29] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proc. Congr. Evol. Comput.*, vol. 1, Jun. 2004, pp. 325–331.

GUOYIN ZHANG received the M.S. degree from the College of Computer Science and Technology, Harbin Engineering University, China, in 1988, the Ph.D. degree from the College of Automation, Harbin Engineering University, China, in 1999. In 2001, he was a Visiting Scholar with the Bauman Moscow State Technical University, Russia. He is currently a Professor with the College of Computer Science and Technology, Harbin Engineering University. His research interests include machine learning and intelligent perception and control.

FENG TAN received the B.S. degree from the College of Shipbuilding Engineering, Harbin Engineering University, Harbin, China, in 2014, where she is currently pursuing the M.S. degree in computer science and technology. Her research interests include ship motion and machine learning.

YANXIA WU received the M.S. and Ph.D. degrees from the College of Computer Science and Technology, Harbin Engineering University, China, in 2005 and 2008, respectively. She is currently an Associate Professor with the College of Computer Science and Technology, Harbin Engineering University. Her research interests include pattern recognition, machine learning, and computer architecture.