Nonlinear Dyn (2022) 107:2447–2467
https://doi.org/10.1007/s11071-021-07139-y

ORIGINAL PAPER

A ship motion forecasting approach based on empirical mode decomposition method hybrid deep learning network and quantum butterfly optimization algorithm

Ming-Wei Li · Dong-Yang Xu · Jing Geng · Wei-Chiang Hong

Received: 9 June 2021 / Accepted: 7 December 2021 / Published online: 19 January 2022

Abstract Ship motion (SHM) forecasting value is an important parameter for ship navigation and operation. However, due to the coupling effect of wind, wave, and current, its time series has strong nonlinear characteristics, so it is a great challenge to obtain accurate forecasting results. Therefore, considering the strong nonlinear of SHM time series, firstly, this paper decomposes the original time series into multiple intrinsic mode functions (IMF) using empirical mode decomposition (EMD) technology and then establishes a hybrid deep learning network for each IMF based on convolutional neural network (CNN) and gated recurrent unit (GRU) according to the characteristics of SHM time series. On this basis, the EMD-CNN-GRU (ECG) hybrid forecasting model of SHM is constructed by integrating a component forecasting model. Secondly, considering the difficulty of hyper-parameters selection of ECG model, this paper improves the butterfly optimization algorithm (BOA) based on quantum theory, designs the quantum coding rules of butterfly spatial position, establishes the optimization process of butterfly algorithm based on quantum coding, and then proposes the quantum butterfly optimization algorithm (QBOA). Finally, a hybrid forecasting approach integrating ECG and QBOA is proposed, namely ECG & QBOA. To evaluate the feasibility and performance of the proposed approach. A prediction experiment was carried out with the SHM data of a real ship. The results indicate that, compared with the other comparison models selected in this paper, ECG-based models have significant higher forecasting accuracy (with MAPE values of 10.86% and 12.69% in two experiments, respectively, and with significant accuracy improvement of at least 10% than other compared models), and the QBOA has obtained more appropriate hyper-parameters combination of ECG model.

Keywords Ship motion forecasting · Deep learning model · Empirical mode decomposition · Quantum computation · Butterfly optimization algorithm

Abbreviations

Ship motion (SHM)	Abbreviation of ship six degree of freedom motion
Empirical mode decomposition (EMD)	A signal decomposition technique
Intrinsic mode function (IMF)	The name of the sequence after EMD decomposition

M.-W. Li · D.-Y. Xu · J. Geng
College of Shipbuilding Engineering, Harbin Engineering University, Harbin, Heilongjiang 150001, China

W.-C. Hong (✉)
Department of Information Management, Asia Eastern University of Science and Technology, 58, Sec. 2, Sichuan Rd., Panchiao District, New Taipei 22064, Taiwan
e-mail: samuelsonhong@gmail.com

Convolutional neural network (CNN)	A deep learning network
Gate recurrent unit (GRU)	A deep learning network
Long short-term memory (LSTM)	A deep learning network
Depth neural network (DNN)	A deep learning network
Butterfly optimization algorithm (BOA)	A parameter optimization algorithm
Quantum butterfly optimization algorithm (QBOA)	An improved BOA
EMD-CNN-GRU (ECG)	Abbreviation of EMD-CNN-GRU hybrid model
Artificial neural network (ANN)	Traditional neural network
Autoregressive moving average (ARMA)	Time series prediction model
Autoregressive integrated moving average (ARIMA)	Time series prediction model
Support vector machine (SVM)	Time series forecasting method
Mean absolute percentage error (MAPE)	Prediction and evaluation index
Root mean square error (RMSE)	Prediction and evaluation index

Variables

$X(t)$	Original ship motion time series
$X_{\max}(t)$	An upper envelope sequence composed of the maximum of $X(t)$
$X_{\min}(t)$	A lower envelope sequence composed of the minimum of $X(t)$
$m(t)$	The sequence of the average values of $X_{\max}(t)$ and $X_{\min}(t)$
IMF(t)	The sequence of Intrinsic Mode Function
$r_n(t)$	The residual sequence
x^k	x^k Represents a feature map of the input tensor of the layer k, which is a one-dimensional tensor
w^k	Filters weight of layer k
b^k	Bias terms
C	The size of filters
D	The depth of the feature map
m	The size of the pooling
s	The step of pooling
ϕ	Activation function
x_t	The input sequence of GRU in the tth time step
h_t	Hidden layer output in the tth time step
$\tilde{h}_t$	Candidate state in the tth time step
z_t	Update gate
r_t	Reset gate
W_r, W_z, W_t and W	Weight parameters
f	f Stands for flavor intensity
c	c Is sensory modal
a	a Is the power component
I	I Is the stimulus intensity related to the fitness value
$g_\theta *$	$g*$ Represents the optimal butterfly position found in the current iteration
r	r Is a random number
P	P Is switch probability
P_r	P_R is a random number
μ and v	μ And v represent the probability amplitude of the basic state
x_j	x_j Represents the position of the ith butterfly
$x^{ij}{}_{\max}$	$x^{ij}{}_{\mathrm{Max}}$ represents the upper search limit of x_{ij}
$x^{ij}{}_{\min}$	$x^{ij}{}_{\mathrm{Min}}$ represents the lower search limit of x_{ij}
θ	θ Represents phase
$\Delta\theta$	$\Delta\theta$ Represents phase increment
d_i	The ith number in the time series
D_i	Normalization result of the ith number
$d_{\max}$	Maximum in time series
$D_{\min}$	The minimum value in time series
$\hat{d}$	The predicted value of the model
f_{fitness}	Fitness function value of the algorithm
L_t	Model loss on training data set
L_v	Model loss on validation data set

1 Introduction

Affected by the coupling effect of wind, wave, and current, the ship presents different motions in six degrees of freedom: roll, pitch, yaw, surge, sway and heave pitch, which is called six degrees of freedom motion of the ship and collectively referred to as ship motion. SHM forecasting is an important technology to assist ship navigation control. The development of modern intelligent ships puts forward higher requirements for SHM forecasting technology. However, affected by the coupling effect of wind, wave, and current, the six degrees of freedom motion of ships are complex and changeable, so it is difficult to obtain accurate prediction results. Therefore, it is necessary to study a more accurate SHM forecasting approach.

As early as the last century, scholars have researched the forecasting approach of SHM. Wiener et al. [1] proposed a statistical prediction approach based on historical motion data, which could meet the requirements of short-term prediction to a certain extent. Bates et al. [2] carried out SHM forecasting based on a statistical prediction approach, and the results showed that with the increase of time, the error gradually increased. Kaplan et al. [3] proposed a convolution prediction method for SHM based on the measured wave height and ship response kernel function. However, it is difficult to obtain the wave height and kernel function in practical application. Sidar and Doolin [4] applied Kalman filtering technology to the real-time prediction of SHM. Triantafyllou and Bodson [5] used the Kalman filtering method to predict ship roll and pitch, verified the feasibility of the Kalman filtering method, and pointed out the application limitations of this method. Given the difficulties in the realization of prediction methods based on SHM state, some scholars began to study prediction methods that do not depend on the SHM state, such as the time series method, grey prediction method, and artificial neural network (ANN) method. Yumori [6] used the time series method to establish the autoregressive moving average (ARMA) model for SHM prediction, and the results show that the ARMA model can predict the phase and amplitude in advance in 8 s. Zhao et al. [7] proposed a large SHM prediction method based on the ARMA model, and the experimental results show that if the wave motion law at 1L (L is the captain) in the forward direction of the bow can be observed, the accuracy of ARMA is higher than that of AR method. However, since the strong nonlinear characteristics of the SHM and ARMA model is based on the linear theory, it is defective to use the ARMA model to simulate the SHM mechanism. The grey prediction method has the advantages of simple operation and low information requirement. For example, Sun and Shen [8] established grey metabolism SHM prediction model based on grey system theory, and the numerical verification results show that this method is simple and reliable. Yin et al. [9] proposed a sequential grey prediction approach based on the online sequential extreme learning machine and applies it to ship rolling prediction, and the experimental results show that the proposed approach is effective in dealing with an uncertain time-varying nonlinear system. However, this method is not suitable for time series with severe fluctuations, so it is difficult to meet the requirements of high accuracy. Aiming at the chaos characteristics and cycle variability of SHM time series, the author of this paper proposed a new dynamic seasonal robust v-support vector regression forecasting model based on a chaos system reconstruction method and a dynamic seasonal adjustment mechanism, and the analysis results show that this method has higher accuracy than the traditional AR method [10]. In the subsequent study, the author proposed a new approach for SHM forecasting by integrating the periodogram estimation method, least squares support vector regression, and chaotic cloud particle swarm optimization, and the numerical experiments show that this method is superior to the periodogram estimation method and BPNN [11]. Research results of references [10] and [11] indicate that SHM forecasting based on a support vector machine is feasible and can obtain satisfactory accuracy. Although fast prediction can be realized based on a support vector machine (SVM), to obtain high-precision prediction results, it is necessary to improve the original SVM reasonably, and the generalization performance of the method needs to be studied. The neural network method has a strong adaptive ability for nonlinear data. For example, Khan [11, 12] applied the ANN to the prediction of SHM and proposed two kinds of weights training methods of ANN. The results show that ANN can predict the SHM in real-time. Although ANN is powerful in dealing with nonlinear problems, it relies on samples. Too much or too little data will expose its inherent defects [10].

In recent years, with the development of deep learning, the neural network model has been expanded to provide more choice space for time series prediction. Kuremoto et al. [13] used Deep Belief Network (DBN) to predict time series and proved that DBN is superior to the traditional ANN and autoregressive integrated moving average (ARIMA) model. Akita et al. [14] used long short-term memory (LSTM) network to predict financial time series and verified the method on real data sets. Chen et al. [15] proposed a new wind speed time series prediction method, which achieved good prediction results by integrating the LSTM network, support vector compression machine, and extreme optimization algorithm. Zhang et al. [15] proposed a new time series prediction method by combining filtering cycle decomposition with GRU. In the field of SHM prediction, Suhermi et al. [16] fully considers the advantages of the ARIMA model and Depth Neural Network (DNN) and proposed a ship roll prediction model based on DNN-ARIMA. The experimental results show that the hybrid model has a better ability to capture nonlinear models than traditional models. Wang et al. [17] proposed a ship attitude prediction method based on an input delay neural network, and the simulation results show that the method can effectively improve the accuracy of ship attitude prediction. Peng et al. [18] used the LSTM network to predict ship attitude, which verified the feasibility of using a recurrent neural network to predict SHM. Zhang et al. [19] established a rolling motion prediction model for unmanned surface vehicles (USV) by combining CNN and LSTM, and the experimental results show that the extraction of time series features of SHM by CNN is helpful to improve the prediction accuracy of the model. Liu et al. [20] proposed an LSTM input vector space optimization method based on the implicit correlation in the sequence of SHM records and explored the relationship between the input dimension and the accuracy of SHM prediction. Daesoo and Seung [21] Jae introduces RNN into a dynamic positioning system, which effectively improves the accuracy of ship motion prediction. Wang et al. [22] proposed a ship roll angle prediction method based on bidirectional long short-term memory network and temporal pattern attention mechanism, and the results show that the model has higher accuracy compared with the LSTM model and SVM model.

When faced with dynamically evolving nonlinear ship motion time series, the prediction accuracy of a single model is often lower than that of a hybrid model [23]. The time series of SHM is rich in information. How to accurately identify this information is related to the accuracy of the forecast model. As a method to deal with nonlinear and non-stationary time series, EMD is widely used in power load forecasting [24–26], rainfall prediction [27–29], and wind speed prediction [30, 31]. The practice shows that EMD has good adaptability for time series, and combining it with other prediction models can effectively improve the adaptability to different time series data and improve the prediction accuracy of the model [32]. In the field of SHM prediction, Zhou and Shi [33] proposed a least square support vector machines SHM prediction approach based on EMD and test the effectiveness of the proposed approach by real SHM data. Huang et al. [23] established an extended AR model, namely EMD-AR, for non-stationary SHM prediction and compared it with AR and SVM models to verify that AR-EMD is effective in handling non-stationarity SHM data. Duan et al. [34] proposed a short-term forecast model of nonlinear and non-stationary SHM by combining AR-EMD and SVR, and the prediction results show that the SVM model is improved after the AR-EMD technique is used. Nie et al. [35] studied the influence of the boundary effect of EMD on the prediction accuracy of the model and proposed an MSEMD-SVR model based on the mirror symmetry method, and the rolling and pitching SHM data are used to confirm the validity of the mirror symmetry method. The above research proves that EMD is effective in dealing with nonlinear and non-stationary SHM time series.

Considering the advantages of the deep learning model in prediction accuracy and the effectiveness of the EMD method in separating information, this paper first establishes CNN-GRU deep learning model and then proposes a new SHM prediction model, EMD-CNN-GRU, based on EMD technology.

Although the deep learning model has a strong learning ability and can simulate the mechanism of SHM, the selection of model hyper-parameters has a great influence on prediction accuracy [20]. Therefore, it is worth trying to study the reasonable method of optimizing hyper-parameters. Generally speaking, the adjustment of the hyper-parameters of the deep learning model needs to be completed by a human, and this process requires a lot of practical experience. How to select reasonable hyper-parameters quickly has been a problem in academic circles. At present, some scholars have used intelligent optimization algorithms to optimize the hyper-parameters for the deep learning model. For example, Kuremoto et al. [13] used particle swarm optimization (PSO) to optimize the DBN model, Peng et al. [18] used PSO to optimize the LSTM model, and Rasdi Rere et al. [36] used simulated annealing algorithm to optimize the CNN model. Arora and Singh [37] proposed a new natural heuristic optimization algorithm, butterfly optimization algorithm, to solve optimization problems. This algorithm is applied to three classical engineering problems, and the results show that BOA is more effective than the comparison algorithm. Aiming at the defects of BOA, Arora, and Singh [38] improved BOA based on chaotic maps to improve the chaotic mapping performance of the algorithm. Although chaos disturbance enhances the ability of BOA, whether it is suitable for optimizing the deep learning model remains to be studied. Therefore, given the difficulty of proposed ECG hyper-parameters selection, aiming at the defects of the standard BOA, quantum computing is introduced to search the optimal position of the butterfly through the quantum revolving gate, and the QBOA is proposed for model hyper-parameters optimization in this paper.

Consequently, coupling EMD, CNN, GRU, and QBOA, a hybrid forecasting approach, namely ECG&QBOA, is proposed. In which, EMD is used to decompose the original time series into multiple IMF components, CNN and GRU are selected to extract the spatial and temporal features of each IMF sequence, and QBOA is adopted to optimize the hyper-parameters of the ECG model. To test the rationality of ECG model structure and the superiority of prediction, this paper selects seven different models to carry out two experiments. At the same time, Inclined Planes System Optimization (IPO) [39], Grey Wolf Optimization (GWO) [40], and BOA are selected as comparative algorithms to test the feasibility of QBOA.

The remainder rest of this paper is organized as follows. The ECG hybrid forecasting model is proposed in Sect. 2. Section 3 provides the detailed designs of QBOA and its optimizing process. The calculation process of the ECG&QBOA hybrid forecasting approach is described in Sect. 4. The numerical testing example is illustrated in Sect. 5. The conclusions are briefed in Sect. 6.

2 The construction of ECG hybrid forecasting model

2.1 Time series decomposition of SHM based on EMD

Due to the influence of complex factors such as wind, wave, and current, the six degrees of freedom motion of the ship show highly nonlinear and non-stationary characteristics. How to deal with the SHM time series effectively has a direct impact on the prediction accuracy of the prediction model. EMD technology can decompose complex time series data into a finite number of IMF. IMF components are oscillatory functions with time-varying frequency, which can reflect the local characteristics of non-stationary time series data. Therefore, the time series of SHM is decomposed into multiple IMF components by EMD, which reduces the difficulty of learning the features of the deep learning model. The IMF has two requirements, the first is that the mean value of the envelope is zero at any time, and the other is that there is only one extreme point between adjacent zero-crossing points. The specific calculation flow of EMD is as follows:

Step 1 All the extremum points of the original data $X(t)$ are found to form the local maximum sequence $X_{\max}(t)$ and the local minimum sequence $X_{\min}(t)$. According to the cubic spline interpolation, all the maximum values are connected into the upper envelope, and the minimum values are connected into the lower envelope.

Step 2 Calculate the average value $m_1(t)$ of the upper envelope and the lower envelope according to Eq. (1) and calculate $h_1(t)$ according to Eq. (2). If $h_1(t)$ meets the two requirements of IMF, it can be considered as the first component.

$$m_1(t) = \frac{X_{\max}(t) + X_{\min}(t)}{2} \tag{1}$$

$$h_1(t) = X(t) - m_l(t) \tag{2}$$

Step 3 If the IMF component requirements are not met, $h_1(t)$ is used as the original data. Repeat *Step 1* and *Step 2* for K times until the $h_1{}^K(t)$ meeting the IMF requirements, and the $h_1{}^K(t)$ is IMF_1. At the same time, the residual sequence is obtained by $r_1(t) = X(t) - h_1{}^K(t)$.

Step 4 Repeat the above steps until $r_n(t)$ is a monotone function or less than a predetermined error.

In this case, IMF components and a residual component can be obtained, and $X(t)$ can be expressed as shown in Eq. (3).

$$X(t) = \sum_{i=1}^{n} IMF_i(t) + r_n(t) \tag{3}$$

2.2 The construction of IMF component forecasting network based on CNN-GRU

The original SHM time series has obvious non-stationary and nonlinear characteristics, and it is difficult to completely extract its rich information by directly using feature extraction methods to operate it, which will also cause the final forecast accuracy to decrease. After EMD decomposition, the original time series becomes a linear superposition of multiple IMFs. The complexity of the sequence is reduced to make the features extracted by the deep learning method more representative. Considering the different advantages of CNN and GRU in processing time series, this paper combines CNN-GRU to build a forecast network for IMF components. In the CNN-GRU network, CNN is used to extract the spatial features of IMF components, and GRU is selected to extract the temporal features of IMF components. The IMF component forecast based on CNN-GRU is shown in Fig. 1, and the calculation process is as follows.

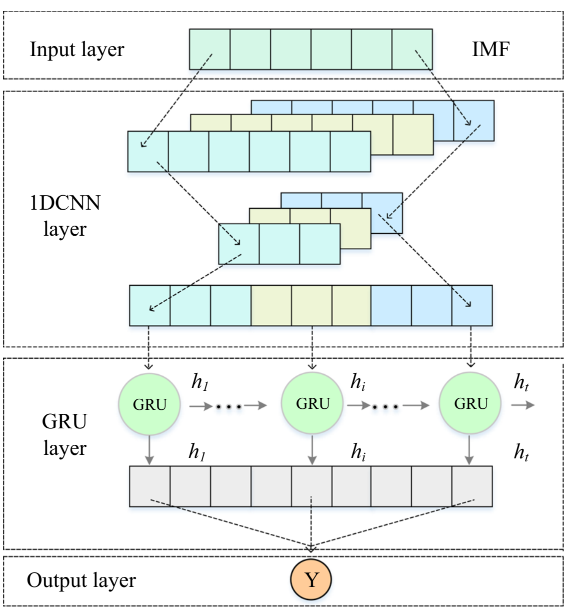


Fig. 1 Calculation diagram of IMF component forecasting network

Step 1 IMF sequence is processed by input layer, which makes it conform to CNN input type.

Step 2 Spatial feature of the IMF sequence is extracted by the CNN convolution layer, the maximum pooling layer is used to downsampling to reduce the data dimension, and the full connection layer is used to flatten.

Step 3 The flattened sequence is extracted by GRU to obtain the output sequence at different times of the hidden layer.

Step 4 The hidden output sequence is mapped to the result through the full connection layer and output through the output layer.

The above is the calculation flowchart of the CNN-GRU network, including two parts: CNN and GRU calculation. The convolution calculation of CNN is shown in Eq. (4), where x_{ij}^k represents the value of the jth dimension on the ith feature map of layer k, w_{ipq}^k is the filter weight value of layer k, b_i^k is bias, C represents the size of the filter, and D represents the depth of the feature map. Max pooling is the operation of taking a fixed number of data on the *i-th* feature map xi for downsampling each time. The pooling output of the *n-th* dimension is shown in Eq. (5), where m represents the size of the pooling, s is the pooling step size. The relationship between the dimension D_b before pooling and the dimension D_a after pooling is shown in Eq. (6).

$$x_{ij}^k = \phi\left(\sum_{p=1}^{D_{k-1}} \sum_{q=1}^{C} w_{ipq}^k \cdot x_{p,j+q}^{k-1} + b_i^k\right) \tag{4}$$

$$x'_{in} = max\left[x_{i,j}, x_{i,j+1}, \ldots x_{i,j+m}\right]\Big|_{j=(n-1)s+1} \tag{5}$$

$$D_a = (D_b - m)/s + 1 \tag{6}$$

The output of the pooling layer is a two-dimensional tensor composed of different one-dimensional characteristic diagrams. To meet the input requirements of GRU, it is flattened into a one-dimensional sequence. The calculation of GRU is shown in Eq. (7) to Eq. (10), where h_t is the hidden state at time t, x_t is the GRU input at time t, $\tilde{h}_t$ is the candidate state, W_z, W_r, and W are the weight matrix to be trained, ϕ is the activation function, z_t and r_t represents the update gate and reset gate, respectively. The output layer of GRU is a two-dimensional tensor composed of hidden outputs at different times $[h_1, h_2, \ldots, h_t]$. The mapping is compressed into the dimension of the output layer through the weight matrix of the full connection layer, and its calculation is shown in Eq. (11).

$$z_t = \phi(W_z \cdot [h_{t-1}, x_t]) \tag{7}$$

$$r_t = \phi(W_r \cdot [h_{t-1}, x_t]) \tag{8}$$

$$\tilde{h_t} = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t} \tag{10}$$

$$Y = [h_1, h_2, \ldots, h_t] \cdot W_t \tag{11}$$

2.3 Hybrid forecasting model based on EMD-CNN-GRU

In Sect. 2.2, the original SHM time series is decomposed into multiple IMF components and a residual by EMD. At the same time, an independent CNN-GRU forecasting network is established for each decomposed series. Considering that the nonlinear characteristics of each component sequence are different, the IMF independent forecasting models with different structures are established and finally integrated into the ECG hybrid forecasting model. Each independent model serves only one component, and the final forecasting result is obtained by adding the outputs of each model. The ECG hybrid forecasting model is shown in Fig. 2.

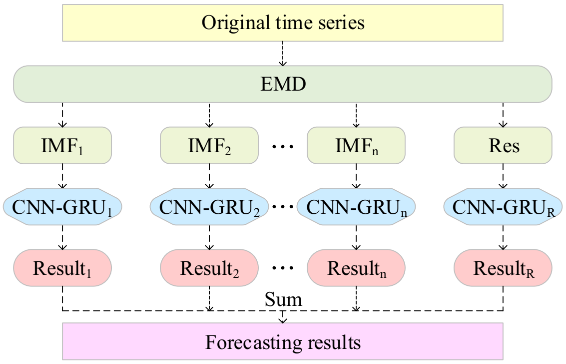


Fig. 2 Structural diagram of ECG hybrid forecasting model

2.4 Model loss function and training algorithm

Considering that the IMF component is a continuous sequence, mean square error (MSE) is selected as the loss function of the CNN-GRU model. The MSE calculation is shown in Eq. (12), where x_i is the real value and $\hat{x}_i$ is the predicted value. Adam [41] algorithm is selected as a model training algorithm.

$$Loss = \frac{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}{n} \tag{12}$$

Although the ECG hybrid forecasting model has been established in this section, the complex hyper-parameters of the model have a great impact on its forecasting accuracy. How to select the hyper-parameters reasonably, such as the number of filters, the size of filters, and the number of GRU hidden layer nodes, is related to the prediction performance of the model. Therefore, a new optimization method of hyper-parameters of the ECG model is established in Sect. 3.

3 The proposal of the quantum butterfly optimization algorithm

3.1 Standard BOA

BOA is a new optimization algorithm to simulate butterfly foraging behavior, which mainly includes three stages: initialization phase, iteration phase, and final phase [37]. It has been widely concerned since it was put forward and has been successfully applied in many engineering fields [38, 42, 43]. Therefore, this paper attempts to apply it to optimize ECG hyper-parameters. The butterfly is the search agent of BOA, which produces some intensity of fragrance related to its fitness. The calculation of fragrance is shown in Eq. (13), where f is flavor intensity, c is sensory modal, I is the stimulus intensity related to the fitness value, and a is the power component. Generally, a and c take the number between 0 and 1.

$$f = cI^a \tag{13}$$

In the iteration stage, the two key points of the algorithm are global search and local search. In the global search phase, the butterfly moves toward the optimal butterfly, and its mathematical expression is shown in Eq. (14), where x_i^t is the position of the t-th iteration of the i-th butterfly, g^* represents the optimal butterfly position found in the current iteration, f_i^t is the fragrance of the i-th butterfly, and r is a random number between 0 and 1. In the local search stage, the butterfly searches randomly, and the mathematical expression is shown in Eq. (15), where x_j^t and x_k^t are two butterflies randomly selected from the butterfly population. In the process of butterfly foraging, global and local searches will occur. Therefore, a switch probability P is set to decide whether to perform a global search or a local search. Each iteration randomly generates a random number P_r between 0 and 1 and compares it with P to determine whether to perform a global search or local search.

$$x_i^{t+1} = x_i^t + (r^2 \times g^* - x_i^t) \times f_i^t \tag{14}$$

$$x_i^{t+1} = x_i^t + (r^2 \times x_j^t - x_k^t) \times f_i^t \tag{15}$$

Like the conventional optimization algorithm, BOA also faces two problems: slow convergence speed and ease to fall into local optimum [38]. The optimization of the deep learning model is special as the discrete solution space and high complexity constraints bring difficulties to the optimal solution. To improve the performance of the algorithm, quantum computing is used to expand the ergodicity of the search and improve the original BOA algorithm.

3.2 The improvement of BOA based on quantum theory

3.2.1 *Quantum coding of butterflies*

In quantum computation, qubits are used to represent the basic states of microscopic particles. The state of qubit at any time can be represented by the basic state, which is called superposition state [44]. It is shown in Eq. (16), where μ and v represent the probability amplitude of the basic state, meeting $|\mu|^2+|v|^2=1$. For the convenience of calculation, this paper directly uses the probability amplitude of the qubit as the butterfly's current position code. The i-th butterfly qubit code Q_i can be expressed by Eq. (17), where N is the dimension of the position vector.

$$|\varphi\rangle = \mu|0\rangle + v|1\rangle \tag{16}$$

$$Q_i = \begin{bmatrix} \mu_{i1} & \mu_{i2} & \dots & \mu_{iN} \\ v_{i1} & v_{i2} & \dots & v_{iN} \end{bmatrix} = \begin{bmatrix} \cos\theta_{i1} \cos\theta_{i2} \cdots \cos\theta_{iN} \\ \sin\theta_{i1} \sin\theta_{i2} \cdots \sin\theta_{iN} \end{bmatrix} \tag{17}$$

3.2.2 *Butterfly position space mapping*

Suppose the position vector $x_i = [x_{i1}, x_{i2}, \ldots, x_{ij}, x_{iN}]$, and the upper and lower of x_{ij} are $x^{ij}_{\max}$ and $x^{ij}_{\min}$, respectively. The qubit code is mapped to the butterfly position code, and each butterfly changes from one position code to two position codes, as shown in Eqs. (18) and (19).

$$x_{ij}^{|0\rangle} = \mu_{ij}\left(x^{ij}_{\max} - x^{ij}_{\min}\right) + x^{ij}_{\min} \tag{18}$$

$$x_{ij}^{|1\rangle} = v_{ij}\left(x^{ij}_{\max} - x^{ij}_{\min}\right) + x^{ij}_{\min} \tag{19}$$

3.2.3 *Butterfly quantum state update*

The quantum revolving gate is used to make the update of the butterfly's position change into the update of the butterfly's quantum probability amplitude. The operation of the quantum revolving gate is shown in Eq. (20), where θ_{ij}^t represents the phase of the j-th position of the i-th butterfly in the t-th iteration, and $\Delta\theta_{ij}^t$ is the phase increment.

$$\begin{bmatrix} \cos\theta_{ij}^{t+1} \\ \sin\theta_{ij}^{t+1} \end{bmatrix} = \begin{bmatrix} \cos\left(\theta_{ij}^t + \Delta\theta_{ij}^{t+1}\right) \\ \sin\left(\theta_{ij}^t + \Delta\theta_{ij}^{t+1}\right) \end{bmatrix} = \begin{bmatrix} \cos\theta_{ij}^t \\ \sin\theta_{ij}^t \end{bmatrix} \cdot \begin{bmatrix} \cos\Delta\theta_{ij}^{t+1} & -\sin\Delta\theta_{ij}^{t+1} \\ \sin\Delta\theta_{ij}^{t+1} & \cos\Delta\theta_{ij}^{t+1} \end{bmatrix} \tag{20}$$

The calculation of phase increment is the core of the quantum revolving gate. Equation (14) and Eq. (15) are used to calculate the phase increment $\Delta\theta_{ij}^t$, and the improved calculation equations are shown in Eq. (21) and Eq. (22), where g_θ^* is the phase vector of the optimal butterfly in the current iteration, θ_j^t and θ_k^t are the phase vector of two butterflies randomly selected.

$$\Delta\theta_i^{t+1} = \Delta\theta_i^t + (r^2 \times g_\theta^* - \theta_i^t) \times f_i^t \tag{21}$$

$$\Delta\theta_i^{t+1} = \Delta\theta_i^t + (r^2 \times \theta_j^t - \theta_k^t) \times f_i^t \tag{22}$$

3.2.4 *Butterfly qubit mutation*

To prevent BOA from prematurely converging and falling into local optimum, a quantum not gate is introduced to realize qubit mutation and increase the diversity of the butterfly population. Firstly, initialize the mutation probability P_m, and then each butterfly randomly generates a random number P_r between 0 and 1 during calculation. If $P_m < P_r$, the mutation operation will not be carried out. On the contrary, half of the qubits will be selected for not gate mutation operation according to Eq. (23).

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_{ij} \\ \sin\theta_{ij} \end{bmatrix} = \begin{bmatrix} \sin\theta_{ij} \\ \cos\theta_{ij} \end{bmatrix} \tag{23}$$

3.3 The calculation flow of QBOA

The improved QBOA transforms the optimization of the butterfly's spatial position into the optimization of the butterfly's qubit. The flowchart is shown in Fig. 3, and the calculation steps are as follows:

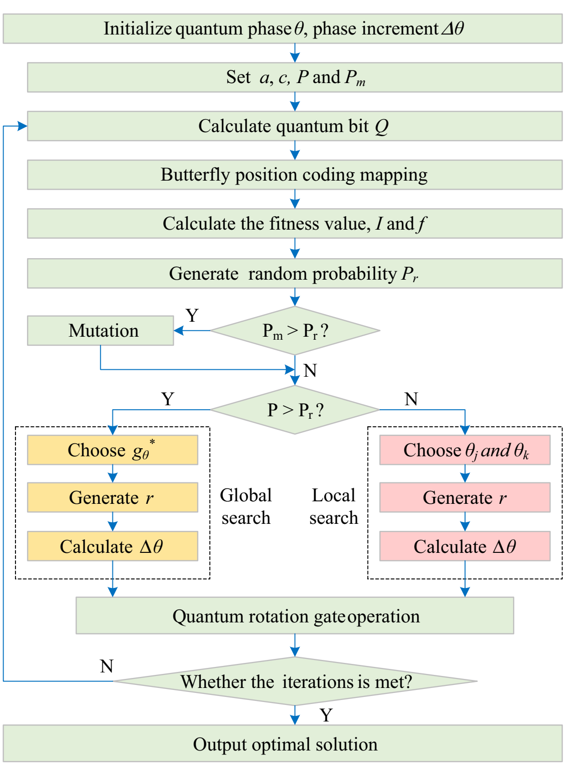


Fig. 3 The calculation flowchart of QBOA

Step 1 The quantum phase θ of the butterfly population is initialized randomly, and the phase increment $\Delta\theta$ is initialized to 0.

Step 2 Set power exponent a, sensory modality c, switch probability P, and mutation probability P_m.

Step 3 Calculate the butterfly population qubit Q according to Eq. (17).

Step 4 According to Eq. (18) and Eq. (19), the butterfly qubits are mapped into the butterfly position space coding.

Step 5 Calculate the fitness value and then calculate stimulus intensity I and fragrance f.

Step 6 Generate a random probability P_r. if $P_m > P_r$, perform the not gate mutation operation according to Eq. (23), and then skip to Step 7, otherwise, skip to Step 7 directly.

Step 7 If $P > P_r$, execute Step 8, otherwise execute Step 9.

Step 8 Select the optimal phase $g_\theta{}^*$ of the butterfly individual in the current iteration, randomly generate r, calculate $\Delta\theta$ according to Eq. (21), and then skip to Step 10.

Step 9 Randomly select two butterflies θ_j and θ_k from the current butterfly group, randomly generate r, calculate $\Delta\theta$ according to Eq. (22), and then skip to Step10.

Step 10 Update the butterfly qubits by quantum revolving gate according to Eq. (20) and then judge whether the termination condition is met, if so, skip to Step 11, otherwise skip to Step 3.

Step 11 Output the optimal butterfly position.

4 The ECG&QBOA hybrid forecasting approach

4.1 Data processing

Data processing mainly includes data normalization processing and network training format processing. Normalization processing is to compress the data to 0 to 1, which prevents the difference between the data is too large and speeds up the network training speed. The normalization calculation is shown in Eq. (24), where d_i is the i-th data in the time series, $d_{\max}$ and $d_{\min}$ represent the maximum and minimum values in the time series, respectively. Network training format processing means that different forecast models adopt different data application methods. At the beginning of training, the data should be divided to meet the training requirements of different models.

$$D_i = \frac{d_i - d_{\min}}{d_{\max} - d_{\min}} \tag{24}$$

4.2 Forecasting process of ECG&QBOA hybrid forecasting approach

The ECG&QBOA hybrid forecasting approach of SHM is mainly composed of two parts: the first part is the ECG model, which is used to predict SHM. The second part is the QBOA algorithm, which is used to optimize the hyper-parameters of the ECG model. The original SHM time series is decomposed by EMD into multiple series data. ECG&QBOA approach is multi-line parallel, and each series adopts the same forecasting method. Therefore, this paper takes IMF_1 training as an example to illustrate the steps of a single branch. The single branch flowchart of the ECG&QBOA hybrid forecasting approach is shown in Fig. 4, and the steps are as follows:

Step 1 Data processing. The IMF_1 sequence is normalized to be between 0 and 1. Then, the format is processed to meet the training requirements of the ECG model. Finally, the data are divided into training data, validation data, and test data, which are reserved for subsequent use.

Step 2 Initialization of model parameters. Set the fixed parameters of the ECG model, such as the number of convolution layers, pooling layer, and activation function. Set the initial parameters of QBOA, and then initialize the quantum parameters randomly.

Step 3 Solution space transformation. The quantum probability amplitude is calculated, and then the qubit coding is mapped to the network hyperparametric solution space coding and assigned to a new model.

Step 4 QBOA calculation. According to the input dimension, obtain training and validation data in the corresponding format, and use the training data to train the inner model of QBOA until it meets the requirement of training termination conditions. Then, the training and validation loss is got and fed back to QBOA as the fitness value, and the quantum probability amplitude is updated according to the QBOA flow until it meets the requirement of the iteration number. Finally, the optimal qubit is mapped into the solution space form of hyper-parameters and output.

Step 5 Training and validation of the optimal model. The optimal hyper-parameters are assigned to a new model, and the model is trained based on the training data. After the training epochs are met, judge whether the model is overfitting according to the validation data. If so, fine-tune the model, then regenerate the model and repeat the above training steps until it meets the requirements. In this case, the forecasting model is obtained.

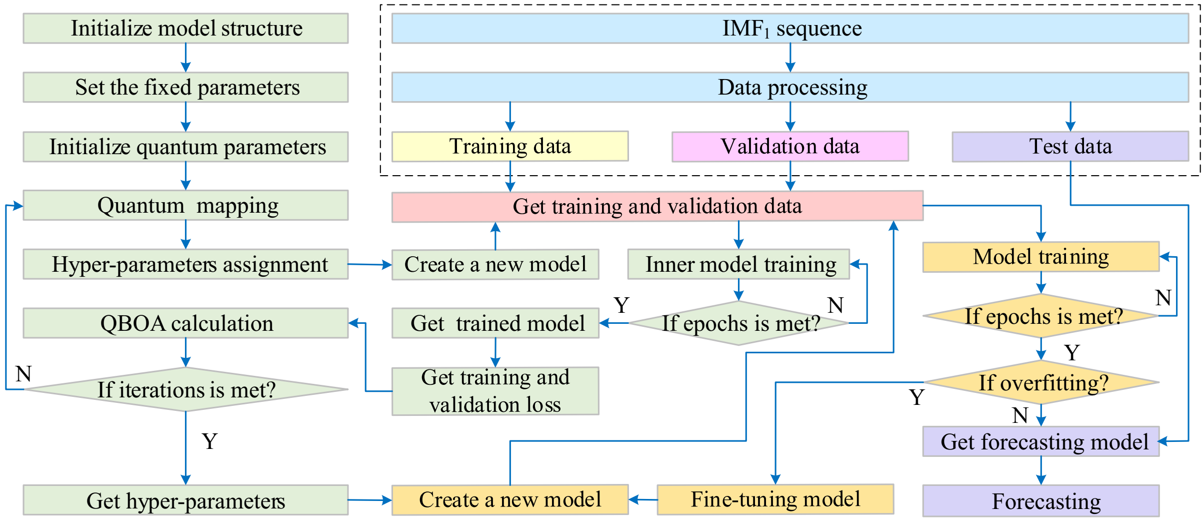


Fig. 4 The flowchart of ECG&QBOA hybrid forecasting approach for IMF_1

Step 6 Model forecasting. Based on the test data, the forecasting experiment is carried out by using the model obtained in Step 5.

4.3 ECG&QBOA parameters setting

Through experiments, when the ECG model is optimized by QBOA, the selection of optimal parameters is shown in Table 1.

5 Example analysis

5.1 Experimental data

5.1.1 Original time series of SHM

To test the superiority of the ECG model and the feasibility of QBOA, the heave motion data of floating production storage and offloading (FPSO) is used for numerical experiments. For the deep learning model, too little data will lead to a poor fitting effect, while too much data will waste computing resources. Therefore, it is necessary to choose the length of time series of example analysis reasonably. Because the purpose is to determine the length of the experimental time series, there is no need to optimize the hyper-parameters of the model. Before the numerical experiment, three random ECG models were selected to determine the length. Data series with different lengths were selected for 10 forecasting experiments, and finally, take the predicted mean absolute percentage error (MAPE) value as the index. The average value of MAPE is shown in Table 2. The curve of the mean value of MAPE with time series length is shown in Fig. 5. From Table 2 and Fig. 5, it can be seen that when the experimental sample time series is too short, the forecasting accuracy is volatile. When the length of the sample series reaches more than 2000, the trend of prediction error tends to be flat. Therefore, taking the forecasting accuracy and computing resources into account, this paper selects the data series with the length of 2000.

The time series data of SHM used in the experiment are shown in Fig. 6, where training data is used to train the ECG model, validation data is used to test whether the model has overfitting or underfitting and test data is used to test the performance of the final model.

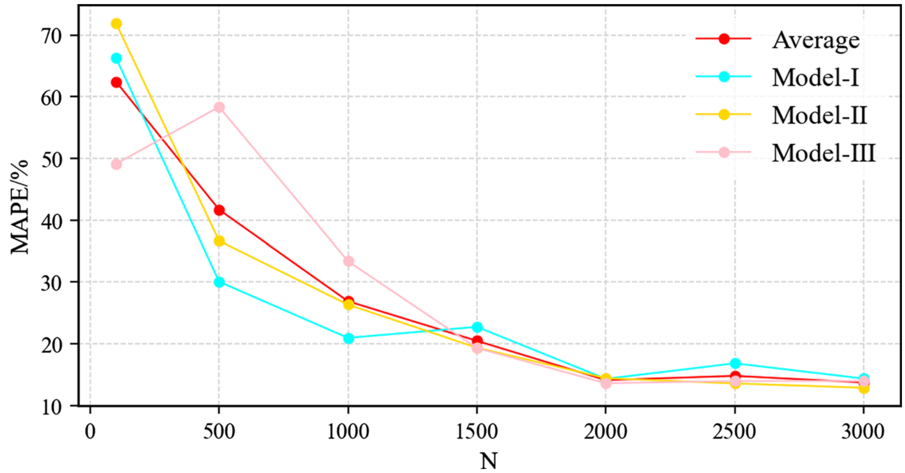
Table 1 The parameters setting and description of ECG model and QBOA algorithm

Object	Variables	Description
CNN	w^k	It is initialized to a small number randomly and finally obtained by model training
	b^k	It is initialized to a small number randomly and finally obtained by model training
	C	C is the hyperparameter to be optimized, which is obtained by QBOA
	D	The value of D is equal to the number of filters and is obtained by QBOA
	m	m is the hyperparameter to be optimized, which is obtained by QBOA
	s	Generally speaking, the pool step size is 1 or 2, and **1** is taken in this paper
	ϕ	The activation function of CNN and GRU is "**sigmoid**" and the activation function of the output layer is "**tanh**"
GRU	W_r, W_z, W_t, and W	Weight parameters are initialized to a small number randomly and finally obtained by model training
QBOA	c	The range of c is [0,1], and **0.8** is taken in this paper
	a	The range of a is [0,1], and **0.9** is taken in this paper
	r	The range of r is [0,1], and **1.0** is taken in this paper
	P	The range of P is [0,1], and **0.5** is taken in this paper
	P_r	P_r is a random number between 0 and 1
	θ	Initialize to a random smaller number
	$\Delta\theta$	Initialize to **0**

Table 2 MAPE value of forecasting model

Time series length	Input dimension	100	500	1000	1500	2000	2500	3000
Model I accuracy mean/%	10	66.32	30.06	20.96	22.75	14.32	16.84	14.36
Model II accuracy mean/%	20	71.93	36.66	26.35	19.37	14.43	13.58	12.87
Model III accuracy mean/%	30	49.11	58.34	33.38	19.31	13.62	13.97	13.97
Average value		62.45	41.69	26.90	20.48	14.12	14.80	13.73

Fig. 5 The curve of the mean value of MAPE under different length data sets



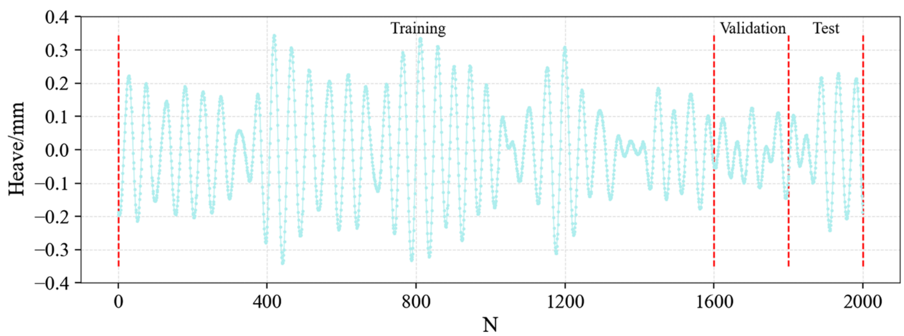


Fig. 6 Original time series of SHM

5.1.2 *Decomposed time series*

The data in Fig. 6 decomposed into seven IMF components and one residual component by EMD, as shown in Fig. 7.

5.2 Accuracy evaluation index and comparison models

5.2.1 *Accuracy evaluation index*

Mean absolute percentage error (MAPE) and root means square error (RMSE) [18] are used as the evaluation indexes of the forecasting results. The calculation is shown in Eqs. (25) and (26).

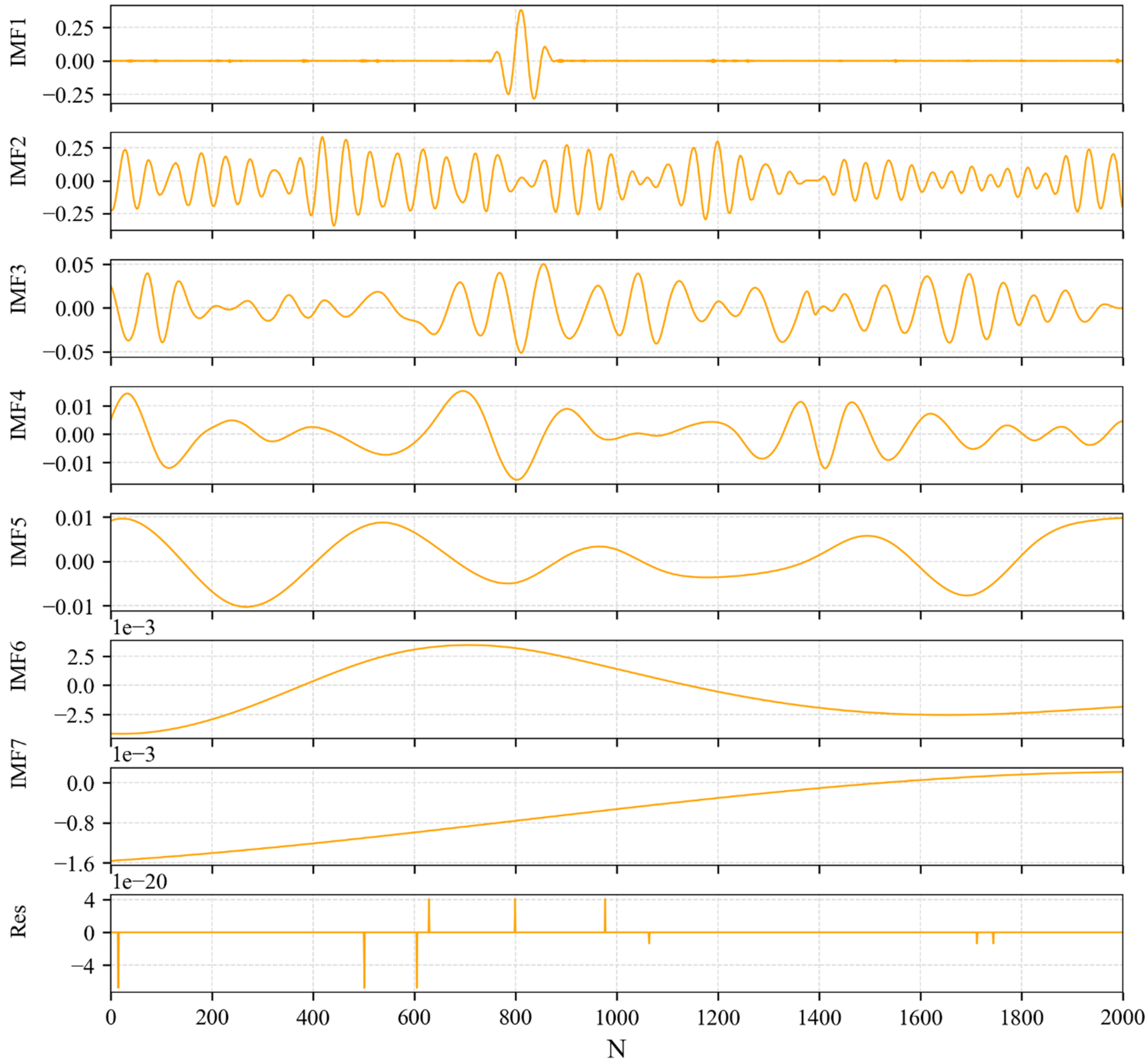


Fig. 7 The SHM time series set after EMD decomposition

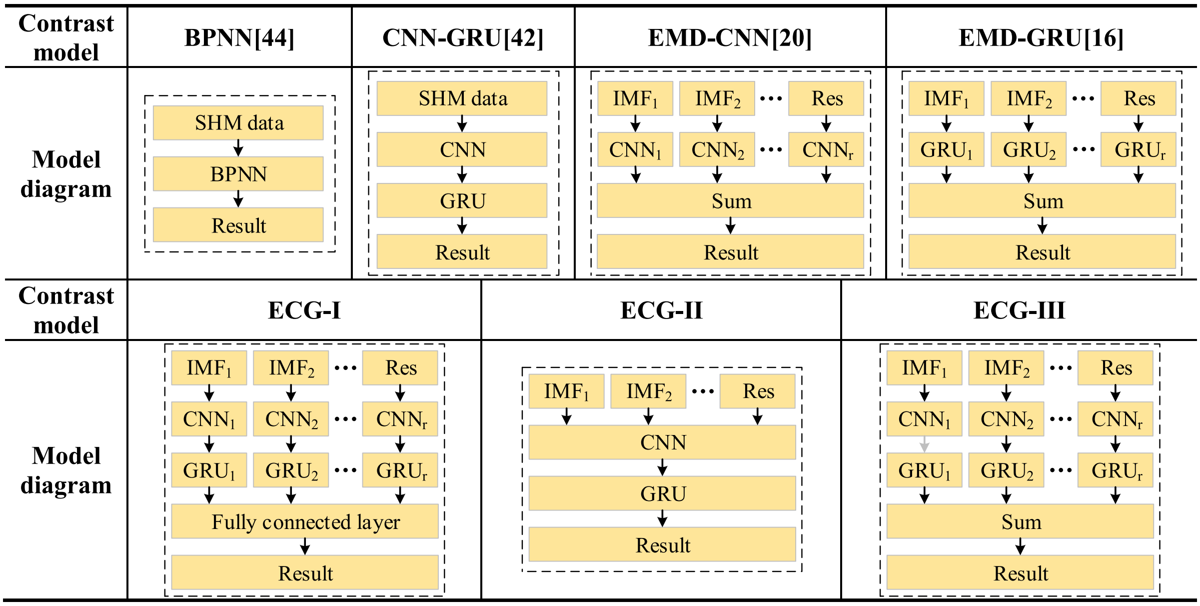
$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{d}(i) - d(i)}{d(i)}\right| \times 100\% \tag{25}$$

$$RMSE = \sqrt{\frac{\sum_{j=1}^{n}\left(d_j - \hat{d}_j\right)^2}{n}} \tag{26}$$

5.2.2 *The choice of comparison models*

To test the superiority of the ECG model structure described in this paper, seven kinds of models are selected for comparison, and the model diagram is shown in Table 3. To distinguish three kinds of ECG models, each model is assigned a number. CNN-GRU is a forecasting model which directly uses the original time series of SHM; EMD-CNN, EMD-GRU, and ECG-III are the component parallel forecasting models, whose final forecasting values are got by summing the component results; ECG-I is a forecasting model with nonlinear mapping of all connected layers; ECG-II is a forecasting model based on the assumption that the component features are not independent and it is coupled with convolution. In this model, each IMF component needs to set the same convolution hyper-

Table 3 Diagram of contrast model



parameters. The rationality of the ECG- III model structure is tested by two comparative experiments, and the results will be shown in the following section.

5.3 Forecasting performance analysis

5.3.1 Forecasting experiment I

The purpose of the experiment I is to compare CNN-GRU, ECG-I, ECG-II, and ECG-III to select the most reasonable structure of the ECG model. In the three ECG models, the same hyper-parameters are set for different IMF components of each model. Taking the influence of random initialization into account, the optimal value and mean value of forecasting accuracy are used to evaluate the forecasting ability of the model. Firstly, the algorithm is used to optimize the model to obtain suitable hyper-parameters. Secondly, each model performs 10 forecasting experiments. Finally, the forecasting accuracy of the four models is shown in Table 4. The forecasting results of the four models in the experiment I are shown in Fig. 8.

It can be seen from Table 4 that in terms of mean value, the ECG-III model with performance criteria (MAPE = 12.69%, RMSE = 0.0063), outperforms the CNN-GRU model with performance criteria (MAPE = 17.16%, RMSE = 0.0073), ECG-I model with performance criteria (MAPE = 17.31%, RMSE = 0.0071) and ECG-II model with performance criteria (MAPE = 15.33%, RMSE = 0.0065), and the optimal value is the same, which indicates that the structure of ECG-III is better than the other two when it is used to SHM forecasting. The ECG-I using the fully connected layer is an automatically trained model. Unlike ECG-III, it can automatically consider the influence of the component forecasting results on the final result. However, it is difficult to train, which makes it difficult to achieve satisfactory results. To ensure the feasibility of convolution operation, the ECG-II model forces the convolution kernel size of each component to be the same, which also leads to the imprecise feature extraction of components. Although it can also achieve high accuracy, its model generalization performance is weak. Therefore, the structure of ECG-III is adopted in this paper.

5.3.2 Forecasting experiment II

The purpose of the forecasting experiment II is to compare the optimal structure of the ECG model with other models to test the ECG-III forecasting performance. In this section, the branch forecasting model of each IMF component is optimized, different convolution hyper-parameters for different IMF components of the ECG-III model are set, and finally, the overall model is established by integrating the component model. Because of the fixed structure of CNN-GRU, no additional experiment is needed this time, but the data in Sect. 5.3.1 are directly used as the basis for comparison. The comparison chart of forecasting results is shown in Fig. 9, and the forecasting accuracy of the experimental model is shown in Table 5.

Figure 9 shows the comparison curves of the forecasting results of several models used in experiment II. It can be seen that the fitting effect of the ECG-III model on the whole data is the best, this is because EMD makes the spatial features extracted by CNN and the temporal features extracted by GRU more representative, and the ECG-III model makes full use of the advantages of the hybrid model compared with EMD-CNN and EMD-GRU.

The data in Table 5 show that the performance of BPNN used in this paper is weaker than that of other models. This is because the feature extraction ability of the traditional neural network model is weaker than that of the deep learning model, and the poor feature representation leads to the larger error of final forecasting results. The comparison between CNN-GRU and ECG-III shows that after EMD, the performance criteria of the model change from MAPE = 17.16%, RMSE = 0.0073 to MAPE = 10.86%, RMSE = 0.0062. This is because EMD makes the original time series into multiple simple series which is more conducive to feature extraction [28]. Comparing EMD-CNN with performance criteria (MAPE = 13.68%, RMSE = 0.0069), EMD-GRU with performance criteria (MAPE = 17.63%, RMSE = 0.0069) and ECG-III with performance criteria (MAPE = 10.86%, RMSE = 0.0062), it can be seen that the accuracy of the integrated forecasting model is higher than that of the single model. In addition, the accuracy of ECG-III with performance criteria (MAPE = 10.86%, RMSE = 0.0062) in the experiment II is higher than that in experiment I (MAPE = 12.69%, RMSE = 0.0063), which indicates that the features of each IMF component are different.

Table 4 Forecasting accuracy of experiment I

Model	MAPE/%	RMSE/mm	Model	MAPE/%	RMSE/mm
CNN-GRU	20.55	0.0077	**ECG-I**	21.73	0.0074
	14.62	0.0070		21.16	0.0075
	13.30	0.0076		17.82	0.0068
	18.06	0.0076		14.11	0.0063
	20.48	0.0077		21.34	0.0074
	14.24	0.0060		**12.29**	**0.0060**
	17.37	0.0064		14.29	0.0061
	18.60	0.0071		17.61	0.0072
	21.79	0.0090		20.20	0.0099
	12.57	**0.0067**		12.59	0.0068
Average value	**17.16**	**0.0073**	**Average value**	**17.31**	**0.0071**
ECG-II	13.55	0.0057	**ECG-III**	14.62	0.0068
	12.43	0.0062		10.04	0.0063
	15.36	0.0069		11.18	0.0058
	19.00	0.0069		**9.25**	**0.0050**
	12.85	0.0066		16.81	0.0068
	10.20	**0.0055**		11.42	0.0064
	14.13	0.0062		15.53	0.0078
	12.00	0.0057		12.28	0.0056
	23.10	0.0074		12.53	0.0054
	20.68	0.0076		13.21	0.0072
Average value	**15.33**	**0.0065**	**Average value**	**12.69**	**0.0063**

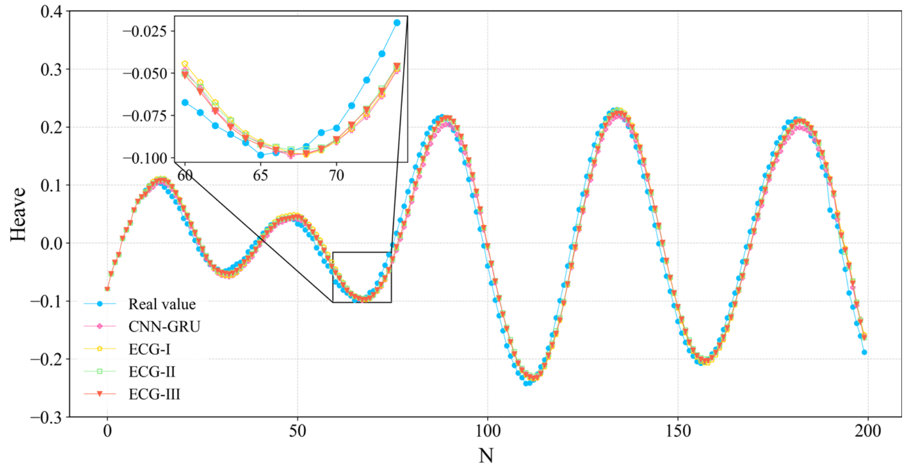


Fig. 8 Comparison figure of different forecasting models in experiment I

It is significant to adopt an independent component model to improve forecasting accuracy. To visually show the improvement of accuracy, this paper calculates the relative accuracy improvement of average MAPE and RMSE of different models, as shown in Table 6.

5.3.3 Model hyper-parameters setting

The hyper-parameters of the optimal model are listed in Table 7. The hyper-parameters descriptions are as follows: filters represent the number of convolution kernels, kernel size represents the size of the convolution kernels, pool size represents the size of the pooling layer, hidden units represent the number of hidden nodes, and input units represent the length of the input time series. The output node of all models is uniformly set to one. 0 means there is no corresponding structure in Table 7.

5.4 Hyper-parameters optimizing performance analysis

5.4.1 Fitness function

The loss of the model decreases gradually with the training process, which shows that the model performs well on the training set. To make the fitness function more representative, this paper takes the performance of the model in the validation data into consideration and selects the fitness function as shown in Eq. (19), where L_t represents the loss of the model in the training data and L_V represents the loss in the validation data.

$$f_{\text{fitness}} = L_{\text{t}} + L_{\text{v}} \tag{19}$$

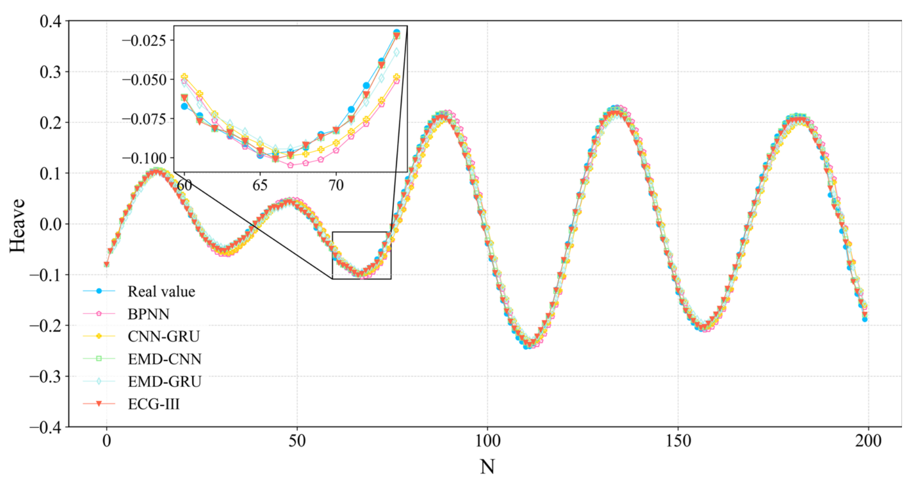


Fig. 9 Comparison figure of different forecasting models in experiment II

Table 5 Forecasting accuracy of experiment II

Model	MAPE/%	RMSE/mm	Model	MAPE/%	RMSE/mm
BPNN	19.76	0.0082	**EMD-GRU**	19.14	0.0068
	25.15	0.0088		16.69	0.0059
	20.09	0.0077		15.51	0.0069
	17.65	0.0074		21.87	0.0074
	22.33	0.0077		**15.29**	**0.0060**
	16.81	0.0068		15.91	0.0071
	23.78	0.0081		17.09	0.0065
	16.81	0.0071		16.39	0.0069
	19.21	0.0086		21.16	0.0084
	16.13	**0.0071**		17.21	0.0075
Average value	**19.77**	**0.0078**	**Average value**	**17.63**	**0.0069**
EMD-CNN	11.63	0.0064	**ECG-III**	11.17	0.0066
	14.32	0.0070		9.01	0.0053
	16.01	0.0089		**7.67**	**0.0049**
	13.02	0.0070		10.32	0.0054
	10.02	**0.0051**		12.39	0.0063
	19.97	0.0105		12.73	0.0074
	12.07	0.0063		9.74	0.0057
	15.16	0.0061		11.88	0.0062
	12.87	0.0060		13.56	0.0071
	11.69	0.0058		10.13	0.0068
Average value	**13.68**	**0.0069**	**Average value**	**10.86**	**0.0062**

5.4.2 Performance analysis of contrast algorithm

To test the performance of QBOA in optimizing the ECG-III model, IPO [39], GWO [40], and BOA [37] are used to perform comparative experiments. The same population, the initial population position, and the number of iterations is set to ensure the credibility of the comparison. At the same time, considering the random factors of the bionic algorithm, each algorithm is used to optimize for 10 times, and the average optimization curve is taken as the final comparison curve. Since the ECG-III model has eight component models, the fitness curves of IMF_1 and IMF_2 component models with iteration times are selected as examples to illustrate the optimization performance of QBOA. The optimization performance comparison is shown in Fig. 10, where (a) represents the optimization curve of IMF_1 and (b) represents the optimization curve of IMF_2.

It can be seen from Fig. 10 that the performance of QBOA is better than the other three algorithms when optimizing the ECG-III model, and QBOA obtains more appropriate hyper-parameters combination in 200 iterations. This is because quantum computing expands the search space of the original butterfly algorithm to improve the global search ability of the algorithm, and the butterfly qubit mutation operation enhances the ability of the algorithm to jump out of the local optimum. There is a subtle difference between (a) and (b) by carefully observing Fig. 10. In Fig. 10a, the convergence speed of GWO is faster than that of QBOA in the early stage of iteration and the 80th iteration. However, in Fig. 10b, the convergence rate of the QBOA algorithm is always the fastest. By analyzing the algorithm structure and calculation process, the following two reasons are obtained: the first is that global search and random search of QBOA are determined by random probability, and there may be more random search in the early stage of the algorithm, the other is that butterfly qubit mutation operation makes the QBOA sacrifice part of the

Table 6 Comparison of accuracy improvement effects

Contrast model	Accuracy improvement of MAPE	Accuracy improvement of RMSE
ECG-III *vs* BPNN	45.07%	20.51%
ECG-III *vs* EMD-GRU	38.40%	10.14%
ECG-III *vs* EMD-CNN	20.61%	10.14%
ECG-III *vs* CNN-GRU	36.71%	15.07%

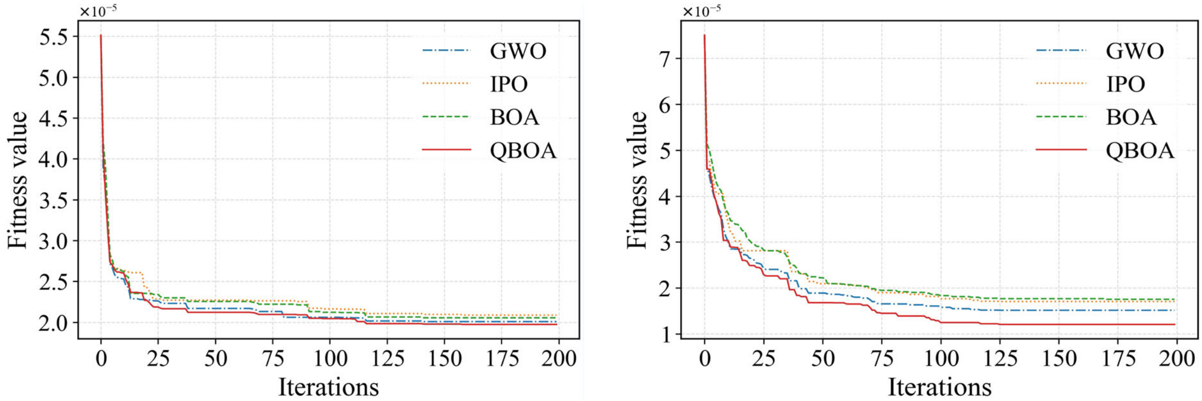
Table 7 The hyper-parameters setting of optimal models

	Model		Filters	Kernel size	Pool size	Hidden units	Input units
Experiment I	CNN-GRU		11	18	3	18	9
	ECG-I		10	15	2	16	8
	ECG-II		10	14	3	18	9
	ECG-III		9	18	2	15	9
Experiment II	BPNN		0	0	0	26	21
	EMD-CNN	IMF_1	14	16	2	0	14
		IMF_2	13	5	2	0	5
		IMF_3	8	10	4	0	10
		IMF_4	14	6	2	0	16
		IMF_5	11	9	5	0	16
		IMF_6	18	8	2	0	12
		IMF_7	6	4	4	0	20
		Res	12	13	3	0	15
	EMD-GRU	IMF_1	0	0	0	17	14
		IMF_2	0	0	0	17	2
		IMF_3	0	0	0	14	8
		IMF_4	0	0	0	13	16
		IMF_5	0	0	0	7	20
		IMF_6	0	0	0	13	6
		IMF_7	0	0	0	18	20
		Res	0	0	0	17	14
	ECG-III	IMF_1	16	17	2	17	14
		IMF_2	20	5	2	17	2
		IMF_3	8	9	6	14	8
		IMF_4	14	6	2	13	16
		IMF_5	10	8	5	7	20
		IMF_6	20	6	2	13	6
		IMF_7	5	3	4	18	20
		Res	15	17	3	16	11

convergence rate. Although qubit mutation operation may affect the early convergence rate of QBOA, it can improve the later search ability of the algorithm. Therefore, the loss of convergence rate is acceptable compared with getting the optimal results.

6 Conclusion

SHM forecasting is an important technology to assist the normal navigation of ships. It is of great practical significance to study the high-precision SHM



(a) The optimization curve of IMF_1 **(b)** The optimization curve of IMF_2

Fig. 10 Comparison of the algorithm optimization performance

forecasting approach. This paper proposes a new hybrid forecasting approach of SHM, namely ECG&QBOA. Considering the strong nonlinearity of the original SHM time series, EMD technology is used to decompose the original series into simple IMF series in the ECG model. Given the spatial attributes and periodic characteristics of the time series, CNN and GRU are integrated to extract IMF component features to establish a component forecasting model. Finally, the ECG hybrid forecasting model is constructed by integrating multiple component forecasting models. In addition, aiming at selecting the hyper-parameters of the deep learning model reasonably, QBOA is proposed by introducing quantum computation and adopted to automatically select the hyper-parameters of the model. To test the proposed approach, the heave motion time series of an FPSO is used to perform an example analysis. The results show that the proposed ECG-III model structure is better than the other two model structure, and its forecasting accuracy is also the highest in the contrast model. The results of optimization performance analysis show that QBOA has better searchability than GWO, IPO, and BOA in optimizing the ECG-III model, and an appropriate combination of hyper-parameters of ECG-III is obtained by QBOA. Consequently, the ECG&QBOA hybrid forecasting approach proposed in this paper is an effective attempt to develop high-precision SHM forecasting approach.

According to the different features of IMF components, this paper sets the same deep learning model structure to study how to select appropriate hyper-parameters to improve forecasting accuracy. However, there are large differences between IMF components. Therefore, whether setting different deep learning model structures of each IMF component will further improve the forecasting accuracy is worth studying in the future.

Acknowledgements The work is supported by the following project grants, National Key Research and Development Program of China (2019YFB1504403); High-tech Ship Technology Project (MC-202030-H04); Heilongjiang Excellent Youth Fund Project (YQ2021E015); National Natural Science Foundation of China (No.51509056); and Ministry of Science and Technology, Taiwan (MOST 110-2410-H-161-001).

Data Availability The datasets generated during and analyzed during the current study are available from the corresponding author on reasonable request.

Declaration

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Wiener, N.: Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering, Applications. MIT Press, Cambridge MA, 10–14, 1949. https://ieeexplore.ieee.org/book/6267356
2. Bates, M.R., Bock, D.H., Powell, F.D.: Analog computer applications in predictor design. IRE Trans. Electron. Comput. **6**, 143–153 (1957). https://doi.org/10.1109/TEC.1957.5222011
3. Kaplan, P.: A study of forecasting techniques for aircraft carrier motions at sea. J. Hydronaut. **3**, 121–131 (1968). https://doi.org/10.2514/3.62814

4. Sidar, M., Doolin, B.: On the feasibility of real-time forecasting of aircraft carrier motion at sea. IEEE Trans. Autom. Control **28**, 350–356 (1983). https://doi.org/10.1109/TAC.1983.1103227
5. Triantafyllou, M.S., Bodson, M.: Real time forecasting of marine vessel motions, using kalman filtering techniques. In: Offshore Technology Conference, 1982, pp. 159–173. https://doi.org/10.4043/4388-MS
6. Yumori, I.: Real time forecasting of ship response to ocean waves using time series analysis. In: Proceeding of OCEANS 81, 16–18 Sept. Boston, MA, USA. https://doi.org/10.1109/OCEANS.1981.1151574 (1981)
7. Zhao, X.R., Peng, X.Y., Lu, S.P., Wei, W.N.: Extreme short forecasting of big ship motion having wave survey. J. Ship Mech. **7**, 39–44 (2003). https://doi.org/10.3969/j.issn.1007-7294.2003.02.005
8. Sun, L.H., Shen, J.H.: Application of the Grey topological method to predict the effects of ship pitching. J. Mar. Sci. Appl. **7**, 292–296 (2008). https://doi.org/10.1007/s11804-008-7111-z
9. Yin, J.C., Zhou, Z.D., Xu, F., Wang, N.N.: Online ship roll motion forecasting based on grey sequential extreme learning machine. Neurocomputing **129**, 168–174 (2014)
10. Li, M.W., Geng, J., Han, D.F., Zheng, T.J.: Ship motion prediction using dynamic seasonal RvSVR with phase space reconstruction and the chaos adaptive efficient FOA. Neurocomputing **174**, 661–680 (2016). https://doi.org/10.1016/j.neucom.2015.09.089
11. Li, M.W., Geng, J., Hong, W.C., Zhang, L.D.: Periodogram estimation based on LSSVR-CCPSO compensation for forecasting ship motion. Nonlinear Dyn. **97**, 2579–2594 (2019). https://doi.org/10.1007/s11071-019-05149-5
12. Khan, A., Bil, C., Marion, K.: Theory and application of artificial neural networks for the real time forecasting of ship motion. In: Khosla, R., Howlett, R.J., Jain, L.C. (Eds), Knowledge-Based Intelligent Information and Engineering Systems (KES 2005), Lecture Notes in Computer Science, vol. 3681, pp. 1064–1069. Springer, Berlin. https://doi.org/10.1007/11552413_151 (2005)
13. Kuremoto, T., Kimura, S., Kobayashi, K., Obayashi, M.: Time series forecasting using a deep belief network with restricted Boltzmann machines. Neurocomputing **137**, 47–56 (2014). https://doi.org/10.1016/j.neucom.2013.03.047
14. Akita, R., Yoshihara, A., Matsubara, T., Uehara, K.: Deep learning for stock forecasting using numerical and textual information. In: Proceeding of IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, 26–29 June 2016, pp. 1–6. https://doi.org/10.1109/ICIS.2016.7550882
15. Chen, J., Zeng, G., Zhou, W., Du, W., Lu, K.: Wind speed forecasting using nonlinear-learning ensemble of deep learning time series forecasting and extremal optimization. Energy Convers. Manag. **165**, 681–695 (2018). https://doi.org/10.1016/j.enconman.2018.03.098
16. Suhermi, N., Suhartono, D.D., Prastyo, B.: Ali, Roll motion forecasting using a hybrid deep learning and ARIMA model. Proc. Comput. Sci. **144**, 251–258 (2018). https://doi.org/10.1016/j.procs.2018.10.526
17. Wang, Y., Soltani, M., Hussain, D.M.A.: Ship attitude forecasting based on Input Delay Neural Network and measurements of gyroscopes. In: Proceedings of the 2017 American Control Conference (ACC), pp. 4901–4907. https://doi.org/10.23919/ACC.2017.7963714 (2017)
18. Peng, X., Zhang, B., Zhou, H.: An improved particle swarm optimization algorithm applied to long short-term memory neural network for ship motion attitude forecasting. Trans. Inst. Meas. Control. **41**, 4462–4471 (2019). https://doi.org/10.1177/0142331219860731
19. Zhang, W., Wu, P., Peng, Y., Liu, D.: Roll motion forecasting of unmanned surface vehicle based on coupled CNN and LSTM. Future Int. **11**, 243 (2019)
20. Liu, Y.H., Duan, W.Y., Huang, L.M., Duan, S.L., Ma, X.W.: The input vector space optimization for LSTM deep learning model in real-time forecasting of ship motions. Ocean Eng. **213**, 107681 (2020). https://doi.org/10.1016/j.oceaneng.2020.107681
21. Lee, D., Lee, S.: Motion predictive control for DPS using predicted drifted ship position based on deep learning and replay buffer. Int. J. Naval Arch. Ocean Eng. **12**, 768–783 (2020). https://doi.org/10.1016/j.ijnaoe.2020.09.004
22. Wang, Y., Wang, H., Zou, D., Fu, H.: Ship roll prediction algorithm based on Bi-LSTM-TPA combined model. J. Mar. Sci. Eng. **9**(4), 384 (2020). https://doi.org/10.3390/jmse9040387
23. Huang, L.M., Duan, W.Y., Han, Y., Yu, D.H.: A hybrid AR-EMD-SVR model for the short-term prediction of nonlinear and non-stationary ship motion. J. Ship Mech. **19**, 1033–1049 (2015). https://doi.org/10.3969/j.issn.1007-7294.2015.09.002
24. Fan, G.F., Peng, L.L., Hong, W.C., Sun, F.: Electric load forecasting by the SVR model with differential empirical mode decomposition and auto regression. Neurocomputing **173**, 958–970 (2016)
25. Wang, X.P., Wang, Y.Q.: A Hybrid Model of EMD and PSO-SVR for short-term load forecasting in residential quarters. Math. Probl. Eng. Article ID: 9895639. https://www.hindawi.com/journals/mpe/2016/9895639/ (2016)
26. He, K.J., Wang, H.Q., Du, J.Z., Zou, Y.C.: Forecasting electricity market risk using empirical mode decomposition (EMD)—based multiscale methodology. Energies **9**, 931 (2016)
27. Bi, S.B., Bi, S.G., Chen, X., Ji, H., Yin, L.: A climate forecasting method based on EMD and ensemble forecasting technique. Asia-Pacific J. Atmos. Sci. **54**, 611–622 (2018). https://doi.org/10.1007/s13143-018-0078-z
28. X.X. Liu, A.B. Zhang, C.M. Shi, H.F. Wang, Filtering and multi-scale RBF forecasting model of rainfall based on EMD method, In: Proceeding of 2009 First International Conference on Information Science and Engineering (CISE 2009), Nanjing, China, 26–28 Dec. 2009, Accession Number: 11281301. https://ieeexplore.ieee.org/document/5455288
29. Xiang, Y., Guo, L., He, L.H., Xia, S.L.: Wang, Wang, A SVR–ANN combined model based on ensemble EMD for rainfall forecasting. Appl. Soft Comput. **73**, 874–883 (2018)
30. Zhang, C., Wei, H.K., Zhao, J.S., Liu, T.H., Zhu, T.T., Zhang, K.J.: Short-term wind speed forecasting using empirical mode decomposition and feature selection. Renew. Energy **96**, 727–737 (2016)
31. Kang, A.Q., Tang, Q.X., Yuan, X.H., Lei, X.H., Yuan, Y.B.: Short-term wind speed forecasting using EEMD-LSSVM

model. Adv. Meteorol. (2017). https://doi.org/10.1155/2017/6856139
32. Hong, W.C., Li, M.W., Geng, J., Zhang, Y.: Novel chaotic bat algorithm for forecasting complex motion of floating platforms. Appl. Math. Model. **72**, 425–443 (2019). https://doi.org/10.1016/j.apm2019.03.031
33. Zhou, B., Shi, A.G.: Empirical mode decomposition based LSSVM for ship motion forecasting. In: Guo, C., Hou, Z.G., Zeng, Z. (Eds) 2013 International Symposium on Neural Networks (ISNN 2013) Advances in Neural Networks, Lecture Notes in Computer Science, vol. 7951, pp. 319–325. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-39065-4_39
34. Duan, W.Y., Huang, L.M., Han, Y., Zhang, Y.H., Huang, S.: A hybrid AR-EMD-SVR model for the short-term forecasting of nonlinear and non-stationary ship motion. J. Zhejiang Univ. Sci. A **16**, 562–576 (2015). https://doi.org/10.1631/jzus.A1500040
35. Nie, Z.H., Shen, F., Xu, D.J., Li, Q.H.: An EMD-SVR model for short-term forecasting of ship motion using mirror symmetry and SVR algorithms to eliminate EMD boundary effect. Ocean Eng. **217**, 107927 (2020). https://doi.org/10.1016/j.oceaneng.2020.107927
36. Rere, L.M.R., Fanany, M.I., Arymurthy, A.M.: Simulated annealing algorithm for deep learning. Proc. Comput. Sci. **72**, 137–144 (2015). https://doi.org/10.1016/j.procs.2015.12.114
37. Arora, S., Singh, S.: Butterfly optimization algorithm: a novel approach for global optimization. Soft. Comput. **23**, 715–734 (2019). https://doi.org/10.1007/s00500-018-3102-4
38. Arora, S., Singh, S.: An improved butterfly optimization algorithm with chaos. J. Intell. Fuzzy Syst. **32**, 1079–1088 (2017). https://doi.org/10.3233/JIFS-16798
39. Mohammadi, A., Hamid Zahiri, S.: Inclined planes system optimization algorithm for IIR system identification. Int. J. Mach. Learn. Cybern. **9**, 541–558 (2018). https://doi.org/10.1007/s13042-016-0588-x
40. Mirjalili, S., Mohammad Mirjalili, S., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014). https://doi.org/10.1016/j.advengsoft.2013.12.007
41. Fei, Z., Wu, Z., Xiao, Y., He, W.: A new short-arc fitting method with high precision using Adam optimization algorithm. Optik **212**, 164788 (2020). https://doi.org/10.1016/j.ijleo.2020.164788
42. Arora, S., Singh, S., Yetilmezsoy, K.: A modified butterfly optimization algorithm for mechanical design optimization problems. J. Braz. Soc. Mech. Sci. Eng. **40**, 21 (2018). https://doi.org/10.1007/s40430-017-0927-1
43. Arora, S., Anand, P.: Learning automata-based butterfly optimization algorithm for engineering design problems. Int. J. Comput. Mater. Sci. Eng. **7**, 1850021 (2018). https://doi.org/10.1142/S2047684118500215
44. Han, K.H., Kim, J.H.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans. Evol. Comput. **6**, 580–593 (2002). https://doi.org/10.1109/TEVC.2002.804320
45. Zhang, X., Shen, F., Zhao, J., Yang, G.: Time series forecasting using GRU neural network with multi-lag after decomposition. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S. (Eds) Neural Information Processing (ICONIP 2017), Lecture Notes in Computer Science, vol. 10638, pp. 523–532. Springer, Cham. https://link.springer.com/chapter/10.1007%2F978-3-319-70139-4_53 (2017)
46. Luo, L.: Network text sentiment analysis method combining LDA text representation and GRU-CNN. Pers. Ubiquit. Comput. **23**, 405–412 (2019). https://doi.org/10.1007/s00779-018-1183-9
47. Jin, C., Jin, S., Qin, L.: Attribute selection method based on a hybrid BPNN and PSO algorithms. Appl. Soft Comput. **12**, 2147–2155 (2012). https://doi.org/10.1016/j.asoc.2012.03.015

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.