Research Article

Online Prediction of Ship Roll Motion in Waves Based on Auto-Moving Grid Search-Least Square Support Vector Machine

Chang-Zhou Xu¹ and Zao-Jian Zou ²¹,²

¹School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
²State Key Laboratory of Ocean Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Correspondence should be addressed to Zao-Jian Zou; zjzou@sjtu.edu.cn

Received 16 August 2020; Revised 5 December 2020; Accepted 15 December 2020; Published 12 January 2021

A novel method based on auto-moving grid search-least square support vector machine (AGS-LSSVM) is proposed for online predicting ship roll motion in waves. To verify the method, simulation data are used, which are obtained by solving the second-order nonlinear differential equation of ship roll motion using the fourth-order Runge–Kutta method, while the Pierson–Moskowitz spectrum (P–M spectrum) is used to simulate the irregular waves. Combining the sliding time window with the least square support vector machine (LS-SVM), the samples in the time window are used to train the LS-SVM model, and the model hyperparameters are optimized online by the auto-moving grid search (AGS) method. The trained model is used to predict the roll motion in the next 30 seconds, and the prediction results are compared with the simulation data. It is shown that the AGS-LSSVM is an effective method for online predicting ship roll motion in waves.

1. Introduction

Prediction of ship motion at sea is important for ship’s safety and efficient operation. Among various modes of ship motion induced by waves, roll motion is the most important one. Due to the nonlinearity of the restoring moment, the damping moment, and the wave exciting moment, ship roll motion in heavy seas is a typical nonlinear motion. Especially under severe sea conditions, strong nonlinear roll motion will seriously threaten the ship’s navigation safety and degrade the ship’s maneuverability. Prediction of ship roll motion is of great significance, which can not only provide a guarantee for marine operations such as cargo lightening and landing of shipborne aircraft under wind and wave conditions but also help to improve the safety of navigation and avoid ship capsizing. Therefore, a study in this field has received widespread attention and various methods have been applied, such as AR (AutoRegressive) models [1], hybrid models combining deep learning and ARIMA (AutoRegressive Integrated Moving Average) [2], local linear wavelet neural networks [3], gray topological methods combining gray systems theory [4], least square support vector machines [5], and gray sequential limit learning machines [6]. As research progresses, the applicability and accuracy of ship roll motion prediction are being improved.

For ship motion prediction, much research has been carried out. The main prediction methods can be divided into three categories, that is, the linear methods, the nonlinear methods, and the hybrid methods of the first two. The main linear methods include the Kalman filter method, bow wave method, AR method, AutoRegressive Moving Average (ARMA) method, and ARIMA method. The commonly used nonlinear methods include Neural Network (NN) and support vector machine (SVM) methods.

Currently, Kalman filtering is the most widely used filtering method and has been well applied in many fields such as communication, guidance, and control. Richter et al.
[7] used the Kalman filter to estimate the ship heave motion in real time. Lin et al. [8] used the Kalman filter and an improved AR model to predict the ship motion. The advantage of the Kalman filter method is that it can estimate the state of a dynamic system from noisy data, and the computation is simple. However, the premise of the known system state equation limits the application of this method to a certain extent, especially for ships sailing at sea. The environmental disturbance makes it difficult to obtain accurate state equations of ship motion. The bow wave method uses wave measurement information at a certain distance from the ship’s bow and the historical data of ship motion to predict the ship’s motion at subsequent moments. However, in the actual ship navigation, the wave information is difficult to measure and the cost is high. The classic AR, ARMA, and ARIMA methods are also widely used in online prediction of ship motion. Wu et al. [9] made a short-term prediction of ship motion based on the improved AR model. The modeling methods of AR, ARMA, and ARIMA models are simple, and the computation burden is small, but they are mainly suitable for modeling and prediction of linear systems; for the problem of strongly nonlinear ship roll motion, these methods still have certain limitations.

In recent years, with the development of artificial intelligence technology, machine learning methods such as NN and SVM have been increasingly widely used in various fields. Huang et al. [10] used a coarse and fine-tuning fixed grid wavelet network to predict ship roll motion. Yin et al. [11] used RBF networks to conduct online prediction of roll motion during ship maneuvering. Yang et al. [12] proposed an online least square SVM (LS-SVM) method combined with the EMD (Empirical Mode Decomposition) for ship roll prediction. Duan et al. [13] proposed a hybrid AR-EMD-SVR (Support Vector Regression) model for the prediction of nonlinear and nonstationary ship motion. As an excellent machine learning method, SVM has rigorous derivation based on mathematical theory and can obtain the optimal solution in the case of limited samples. It has an excellent performance in nonlinear function fitting, as well as parameter identification and prediction of ship nonlinear motion.

For machine learning algorithms, there are usually two kinds of parameters, that is, model parameters and hyperparameters. The model parameters do not need to be set in advance and can be learned by the model, while the hyperparameters need to be set in advance. For different problems, different data sets, and different models, the hyperparameters are different. Like other machine learning methods, the performance of SVM depends on the selection of its hyperparameters [14]. Different data distributions often have different hyperparameter combinations.

At present, for the prediction of ship roll motion, offline training and then online prediction are usually implemented. The hyperparameters are selected based on experiences in the offline training phase or selected by an optimization algorithm, such as grid search, random search, genetic algorithm, particle swarm optimization, and Bayesian optimization. The problem of offline training is that it does not consider the complex and changeable sea conditions in actual navigation. This means that the optimized hyperparameters for the offline data may not be suitable for the online data under different sea conditions.

In order to meet the requirements of online prediction, this paper proposes an effective auto-moving grid search (AGS) method to innovate in the online hyperparameter optimization for LS-SVM. In the AGS method, the offline grid search algorithm is evolved into an online one that can be automatically adjusted by controlling the relative proportion of the grid rather than the actual grid values. This gives the proposed prediction method the features of online updating the sample set, online learning, and real-time predicting.

The rest of the paper is organized as follows: Section 2 describes the principles of the algorithm, including LS-SVM, time windows, and AGS; Section 3 presents the data preparation, that is, the acquisition of ship roll motion data by simulation; in Section 4, the prediction results are presented and compared; and Section 5 provides a summary of the innovations and shortcomings of the methodology presented in this paper.

2. Online Predictions

2.1. LS-SVM. LS-SVM is a variant of SVM. By selecting the square loss function, although the LS-SVM loses the important feature of sparseness, it converts the solution of the quadratic optimization problem into the solution of linear equations, which greatly speeds up the computation, and does not affect the accuracy of the computation [15]. In recent years, LS-SVM has been used in many studies such as image classification [16], control of nonlinear systems [17], short-term wind speed prediction [18], and freight volume prediction in logistics systems [19].

Consider a multiple-input single-output system, the set of training sample pairs is \(\{(x_i, y_i)\}_{i=1}^l, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}\), where \(R\) is the Euclidean space, \(m\) is the dimension of the input in a single sample pair, and \(l\) is the total number of sample pairs. LS-SVM maps the input data into a high-dimensional space through nonlinear mapping, thereby transforming the nonlinear regression problem into a linear regression problem. The regression form is expressed as

\[
y(x) = w^T \varphi(x) + b,
\]

where \(x\) is the input, \(y(x)\) is the output, \(\varphi(x)\) is the high-dimensional nonlinear mapping used; and \(w, b\) are the weight vector and bias vector to be solved.

The solution of LS-SVM is transformed into the following optimization problem:

\[
\min_{w, b} \frac{1}{2} w^T \mathbf{w} + \frac{C}{2} \sum_{i=1}^{l} e_i^2
\]

s.t. \(y_i = w^T \varphi(x_i) + b + e_i, i = 1, 2, \ldots, l\),

where \(e_i\) is the regression error and \(C\) is the penalty parameter.
Using structural risk minimization, the Lagrange function of the objective function and constraint conditions is given as

\[ L(w, b, e, \alpha) = f(w, e) - \sum_{i=1}^{L} \alpha_i [w^T \varphi(x_i) + b + e_i - y_i]. \tag{3} \]

Taking the partial derivative with respect to the variables \( w, b, e, \alpha \) in equation (3) and setting the derivatives to zero, the following equation is obtained:

\[
\begin{align*}
\frac{\partial}{\partial w} L &= 0 \rightarrow w = \sum_{i}^{L} \alpha_i \varphi(x_i), \\
\frac{\partial}{\partial b} L &= 0 \rightarrow \sum_{i}^{L} \alpha_i = 0, \\
\frac{\partial}{\partial e_i} L &= 0 \rightarrow e_i = C \alpha_i, \\
\frac{\partial}{\partial \alpha_i} L &= 0 \rightarrow w^T \varphi(x_i) + b + e_i - y_i = 0.
\end{align*}
\tag{4}
\]

After eliminating \( e_i \) and \( w \), the following equation is obtained:

\[
\begin{bmatrix}
0 \\
I_i \\
U_i + C^{-1} I_i \\
\end{bmatrix} \begin{bmatrix}
b \\
\alpha \\
y \\
\end{bmatrix} = \begin{bmatrix}
0 \\
\end{bmatrix}.
\tag{5}
\]

where \( I_i = [1, 1, \ldots, 1]^T, U_{ij} = (\varphi(x_i)^T \varphi(x_j)) = K(x_i, x_j) \) \((i, j = 1, 2, \ldots, L)\) \((K(x_i, x_j)\) is the kernel function\), and \( \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_l]^T, y = [y_1, y_2, \ldots, y_l]^T \). By introducing the kernel function, it is avoided to solve the specific high-dimensional nonlinear mapping \( \varphi(x) \); one only needs to express its inner product through the kernel function.

One of the advantages of SVM is to introduce a kernel function, thereby transforming the complex nonlinear mapping into a relatively simple kernel function. The commonly used kernel functions include linear kernel, polynomial kernel, and Gaussian kernel. For nonlinear problems, Gaussian kernel function is generally used [20], and its expression is as follows:

\[ K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right). \tag{6} \]

where \( \sigma \) is the parameter of the kernel function. After determining the kernel function, the parameters \( \alpha \) and \( b \) in (5) are solved to obtain the following regression estimation function:

\[ y = \sum_{i}^{l} \alpha_i K(x_i, x) + b. \tag{7} \]

2.2. Online Learning Algorithm. The motion of a ship in waves is a dynamic process that constantly changes over time, during which new sample data are continuously generated. If all samples are used for SVM training, the algorithm will inevitably become slower and slower. In addition, the future state has little correlation with the old samples but has a great correlation with the new samples. Therefore, the online learning of SVM can be decomposed into a learning process of adding one sample (incremental learning) and eliminating one sample (decremental learning). This paper uses the method of sliding time window to update the samples, that is, when new samples are added to the training sample set, the oldest samples in the training sample set are deleted to keep the total number of samples in the time window constant. This can not only learn the new features of the system in time but also maintain the stability of the algorithm in terms of time consumption.

The samples in the time window are combined according to Table 1 to form a training sample set used to train the model online. After the model training is completed, the k-step motion prediction of future values is realized using the trained model according to Table 2.

2.3. Auto-Moving Grid Search Algorithm. In machine learning algorithms, adjusting the model hyperparameters is an important task. For the LS-SVM using Gaussian kernel, there are two hyperparameters that need to be adjusted: the penalty parameter \( C \) and the kernel parameter \( \sigma \). The penalty parameter represents the intensity of the penalty for errors. The larger the value, the smaller the tolerance for errors, and the more likely it is to be affected by noise, which may result in overfitting. The kernel parameter \( \sigma \) affects the complexity of mapping the low-dimensional samples to the high-dimensional space. The smaller the value of this kernel parameter, the higher the dimensionality of the mapping; and it is also prone to overfitting, which reduces the generalization ability. Thus, the hyperparameters should neither be too large nor too small; choosing a suitable hyperparameter combination is extremely important to the performance of SVM.

For the hyperparameter optimization problem, the commonly used methods include grid search [21], random search [22], genetic algorithm [23], particle swarm optimization [24], and Bayesian optimization [25]. Grid search is an exhaustive search method. Its core is to list all combinations of parameters, try every possibility, and select the best hyperparameter combination. Compared with the grid search, the random search does not try all parameter values but selects a fixed number of parameter settings from a specified distribution. Its theoretical basis is that if the random sample point set is large enough, then the global optimal solution or its approximate value can be found. Random search is faster than grid search, but its
disadvantage is that its results cannot be guaranteed and are not stable enough. The genetic algorithm is a method to search for the optimal solution by simulating the natural evolution process. Particle swarm optimization originated from the study on predation behavior of birds, and its basic idea is to find the optimal solution through cooperation and information sharing between individuals. The main idea of Bayesian optimization is to use the Gaussian process, consider the previous parameter information, and continuously update the prior. All of the methods mentioned above can be used for offline hyperparameter optimization of SVM. However, for the problem of online prediction of ship motion, there is a limitation to give prediction results within a limited time interval; and it is needed to update the hyperparameters online. Moreover, the running time of the offline optimization methods mentioned above is usually in hundreds of seconds [26], and if they are applied directly to the online case, they will not be able to meet the real-time requirements of the online case due to the large time consumption in parameter optimization.

Therefore, it is necessary to improve the hyperparameter optimization methods to make them suitable for online prediction of ship motion. Considering the stable and effective characteristics of the grid search method, this paper proposes an improved grid search scheme, the auto-moving grid search (AGS), which can dynamically adjust the hyperparameters online. Figure 1 shows the flowchart and the main idea of the AGS algorithm.

For the two hyperparameters $C$ and $\sigma$ to be optimized in LS-SVM, the initial parameter combinations $C_0$ and $\sigma_0$ can be selected, and a set of search ratios $K = [r_1, r_2, r_3, \ldots, r_n]$ can be defined at the same time. As the time step progresses, the hyperparameters are continuously updated iteratively. At each time, the selection of hyperparameters is based on the following scheme: assuming the value of the hyperparameter to be optimized is $W$ and the search grid is $K$, the possible value $W \times K$ of this round is obtained, which is the search range of this round; the samples in the time window are divided into the training set and validation set with the ratio of 8:2 that is commonly used in machine learning algorithms. Then, multiple models on the training set are trained at the same time, and each hyperparameter value corresponds to a model. After the model training is completed, the accuracy of each model is verified on the validation set, and the model with the smallest error is selected as the predicting model of the current round, and its corresponding hyperparameter is the optimal hyperparameter at this time. Since each search is performed based on the excellent hyperparameters selected in the previous round, the hyperparameters are continuously optimized and improved to meet the needs of online parameter adjustment.

Besides, the setting of the search ratio $K$ follows the principle of density: the grids adjacent to the

---

**Table 1: Training sample set.**

| Sample  | Input | Output |
|---------|-------|--------|
| Sample 1 | $(x_{1t}, x_{1t+1}, \ldots, x_{1t+m-1})$ | $x_{1t+m}$ |
| Sample 2 | $(x_{1t+1}, x_{1t+2}, \ldots, x_{1t+1+m})$ | $x_{1t+m+1}$ |
| ... | ... | ... |
| Sample l | $(x_{1tl-1}, x_{1tl}, \ldots, x_{1tl+m-2})$ | $x_{1tl+m-1}$ |

**Table 2: k-step prediction.**

| Prediction | Input | Output |
|------------|-------|--------|
| Step 1     | $(x_{1t}, x_{1t+1}, \ldots, x_{1t+m-2}, x_{1t+m-1})$ | $\tilde{x}_{1t+m}$ |
| Step 2     | $(x_{1t+1}, x_{1t+2}, \ldots, x_{1t+m-3}, \tilde{x}_{1t+m})$ | $\tilde{x}_{1t+m+1}$ |
| ...        | ...   | ...   |
| Step k     | $(x_{1tk-1}, x_{1tk}, \ldots, x_{1tk+m-3}, \tilde{x}_{1tk+m-2})$ | $\tilde{x}_{1tk+m-1}$ |
hyperparameters selected in the previous round are denser, and the grids far away are sparser. This setting can not only obtain a larger search range but also make minor adjustments to the hyperparameters selected in the previous round.

3. Numerical Simulation of Ship Roll Motion in Irregular Waves

3.1. Nonlinear Ship Roll Motion. The linear roll motion of a ship can be described by a second-order linear differential equation with constant coefficients, and an accurate solution can be obtained by conventional methods. But in fact, the ship motion in heavy seas is nonlinear, and the large-amplitude roll motion is more prominent, which is a typical nonlinear motion problem [10]. The nonlinear roll motion of a ship is affected by many factors. It is generally believed that the nonlinear roll motion is mainly caused by the nonlinearity of the restoring moment, the nonlinearity of the damping moment, and the nonlinearity of the wave exciting moment.

The nonlinearity of the restoring moment is determined by the ship’s static stability curve. In the range of small roll angle, there is a linear relationship between the restoring moment \( M_r \) and the roll angle \( \phi \), that is, \( M_r = D\phi \), where \( D \) is the displacement and \( \phi \) is the transverse metacentric height. As the roll angle increases, the restoring moment exhibits a nonlinearity with the roll angle. The general nonlinear restoring moment can be expressed as

\[
M_r = C_1\phi - C_2\phi^3 - C_3\phi^5,
\]

where \( C_1, C_2, \) and \( C_3 \) are constants.

The widely used models of nonlinear roll damping moment mainly include

\[
M(\phi) = A\phi + B\phi^3,
\]

\[
M(\phi) = A\phi + C\phi^3,
\]

where \( A, B, \) and \( C \) are constants.

When the assumption of small-amplitude waves is not valid, the wave exciting moment has a nonlinear relationship with the roll angle, which will cause the nonlinear roll motion of the ship.

3.2. Irregular Waves and the Wave Exciting Moment. The real ocean waves are random and irregular and cannot be described in a definite form. However, considering the convenience of the study, they can be regarded as a superposition of a certain number of regular waves with different wavelengths, amplitudes, and random phases. In this paper, the Pierson–Moskowitz spectrum (P–M spectrum) is used to simulate the random irregular waves. The expression is as follows:

\[
S_\iota(\omega) = \frac{0.11 \times 10^{-3} g^2}{\omega^5} \exp\left(-\frac{0.74g^4}{U^4\omega^6}\right),
\]

where \( \omega \) is the wave frequency (rad/s); \( g \) is the gravitational acceleration, \( g = 9.81 \text{ m/s}^2 \); and \( U \) is the wind speed (m/s) at a height of 19.5 m above the sea.

When the significant wave height \( h_{1/3} \) is known, the wind speed can be approximately estimated by the following formula:

\[
U = 6.85\sqrt{h_{1/3}}.
\]

Using the wave spectrum, the random wave exciting moment on a ship sailing in irregular waves can be deduced:

\[
M(t) = D\phi \sum_{n=1}^{N} K_n K\frac{\omega_n^2}{g} \left[ 2 \int_{\omega_n}^{\infty} S_\iota(\omega) d\omega \cos\left(\omega_n t + \theta_n\right) \right],
\]

where \( x \) is the angle between the ship’s speed \( V \) and the direction of wave propagation, called encounter angle; \( N \) is the number of random regular waves (cosine waves), \( K_nK \) is the effective wave inclination coefficient, \( \omega_n \) and \( \theta_n \) are the frequency and random phase of the \( i \)-th regular wave, respectively. \( \omega_n = \omega_c \cos(x/\langle x \rangle) \) in deep water.

The irregular waves can be regarded as a random process with an energy spectrum of limited bandwidth. Generally, when the number of cosine wave components \( N \) reaches 30, this requirement can be met; thus, \( N \geq 30 \).

3.3. Simulation Data. In this paper, a maritime patrol boat [27] is taken as the study object. Its main parameters are listed in Table 3. The significant wave height is set as \( h_{1/3} = 1.3 \text{m} \), and the number of cosine waves is taken as \( N = 50 \). According to the data of the P–M spectrum, when the sea state level is low, the energy of the waves is more dispersed, and the simulated frequency band can be set to be relatively wide; on the contrary, if the sea state level is high, the energy of the waves is more concentrated, and the simulated frequency band can be set to be relatively narrower [28]. Taking the simulated frequency range as 0.5–3.0 rad/s, the wave spectral density \( S_\iota(\omega) \) obtained according to the P–M spectrum is shown in Figure 2.

The ship navigating in the waves is affected by the restoring moment, damping moment, and the random wave exciting moment, and they are described by (8), (10), and (13), respectively. The second-order nonlinear differential equation for the ship roll motion can be derived as follows:

\[
\ddot{\phi} + 0.0696\dot{\phi} + 0.08\phi + 1.2499\phi - 1.0057\phi^3 + 0.1012\phi^5 = \frac{M}{I_\phi}.
\]

In order to more comprehensively illustrate the effectiveness of the AGS method, simulation tests are carried out for four encounter angles, namely, 60°, 90°, 120°, and 150°, at the ship speed 18 kn. For equation (14), the analytical solution cannot be obtained, and the fourth-order Runge–Kutta method is used to obtain the numerical solution. The numerical simulation step is 0.01 s, and the total simulation time is 500 s. For the 50000 sets of roll angle data obtained from the simulation, samples are taken every 50, that is, the sampling step is 0.5 s. As an example, the
simulation results under the encounter angle of 120° are presented, as shown in Figure 3.

4. Online Prediction of Roll Motion and Analysis of the Results

In order to verify the effectiveness of the AGS algorithm, the results of two methods for online prediction are compared: one is to use constant hyperparameters and the other is to use those obtained by the AGS algorithm.

In the study, four encounter angles are considered, corresponding to four sets of simulation data. At the same time, four sets of hyperparameters are used for each set of simulation data, making a total of 16 sets of comparison cases. As an example, this section provides a graphical comparison of the prediction results for the 120° encounter angle, as shown in Figures 4–7.

4.1. Online Prediction Using Constant Hyperparameters

Four sets of hyperparameters are used, which are \([C_0 = 50, \sigma_0 = 30], [C_0 = 10, \sigma_0 = 10], [C_0 = 100, \sigma_0 = 100], \) and \([C_0 = 1, \sigma_0 = 50]\). The training sample pairs are formed from the samples in the sliding time window. The input dimension of each sample pair is \(m = 150\), and there are 250 sample pairs in the time window, that is, \(l = 250\). The composition of sample pairs within the time window is shown in Table 4.

All samples in the time window are used to train LS-SVM, and the trained model is applied to predict future roll motion. Multistep forward prediction is carried out in an iterative manner and multiple-step prediction is realized through single-step prediction; that is, after each step of prediction, the output value of the current step is updated and used as the input of the next step to predict the roll motion. As an example, the roll motion of the next 60 steps (30 s) is predicted, and the input and the output of each iteration are given in Table 5.

From the perspective of the overall time span, the longer the prediction time, the worse the overall prediction accuracy. To show the prediction accuracy, the values of roll angle at the next 60th step of each prediction, that is, the prediction values at the next 30 s of every time, are presented in comparison with the simulated ones in Figure 4. Figure 5 shows the error between the predicted value and the simulated value (the simulated value minus the predicted value). Both Figures 4 and 5 are the results for the encounter angle 120°, \(C_0 = 10\), and \(\sigma_0 = 10\).

To further analyze the prediction accuracy, mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), and correlation coefficient (CC) are used to evaluate the prediction results. MAE is the average of the absolute value of the difference between the predicted value (denoted by subscript “predict”) and the simulated value (denoted by subscript “real”). MSE refers to the mean value of the square of the difference between the predicted value and the simulated value. RMSE is the square root of the mean square of the difference between the predicted value and the simulated value. CC is used to measure the correlation between two sets of data, and its value ranges within \((-1, +1)\); \(CC = 0\) means that the two variables are not correlated.
correlated, and a positive value indicates a positive correlation, while a negative value indicates a negative correlation; a larger value indicates a stronger correlation. The MAE, MSE, RMSE, and CC are defined as follows:

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |y_{\text{real}}^{(i)} - y_{\text{predict}}^{(i)}|, \tag{15}
\]

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (y_{\text{real}}^{(i)} - y_{\text{predict}}^{(i)})^2, \tag{16}
\]

\[
RMSE = \left( \frac{1}{n} \sum_{i=1}^{n} (y_{\text{real}}^{(i)} - y_{\text{predict}}^{(i)})^2 \right)^{1/2}, \tag{17}
\]

\[
CC = \frac{\text{Cov}(y_{\text{real}}, y_{\text{predict}})}{\sqrt{\text{Var}(y_{\text{real}}) \cdot \text{Var}(y_{\text{predict}})}}, \tag{18}
\]

where \( n \) is the number of samples.

Substituting the prediction data and simulation data in Figure 4 into equations (15)–(18), the values of MAE, MSE, RMSE, and CC under the encounter angle 120° can be determined. Besides, similar predictions are conducted for encounter angles 60°, 90°, and 150°. The results of four sets of encounter angles and four sets of hyperparameters, that is, a total of 16 sets of cases are presented in Tables 6–9.

4.2. Online Prediction Using the AGS Algorithm. Then, the AGS algorithm is used to predict the roll motion online. The dimension of sample input, the number of sample pairs in the time window, and the initial hyperparameters are consistent with those used in the method with fixed hyperparameters.

For the AGS algorithm, a set of search ratios \( K = [0.002, 0.01, 0.2, 0.6, 1, 2, 5, 100, 2000] \) is chosen; the grid setting follows the principle of sparse on both sides and dense in the middle. Assuming that the excellent hyperparameters in the previous round are \( C_i \) and \( \sigma_i \), it can be determined that the search range of \( C \) in this round is
and the search range of $\sigma$ in this round is $K \times \sigma_i$. The 250 sample pairs in the time window are divided into a training set and a validation set according to the commonly used ratio of 8:2 in machine learning algorithms. Excellent hyperparameters of this round are selected based on the principle of least square error, which are used for prediction.

In the specific implementation, considering that the time complexity of searching for $C$ and $\sigma$ simultaneously is too large (the order of time complexity is $O(n^2)$), a compromise solution is selected: $C$ and $\sigma$ are updated separately and alternately, and only one of them is optimized in each round, reducing the time complexity to $O(n)$. Figure 6 shows the comparison between the simulated value and the predicted value obtained by using the AGS algorithm to optimize the hyperparameters online, and Figure 7 shows the error (the simulated value minus the predicted value). Both Figures 6 and 7 are the results for the encounter angle 120°, $C_0 = 10$, and $\sigma_0 = 10$.

When using the AGS algorithm for prediction, the hyperparameters of LS-SVM are optimized at each time step according to the distribution of samples in the time window, and the hyperparameters obtained from each step of optimization are shown in Figure 8.

Substituting the prediction data and simulation data in Figure 6 into equations (15)–(18), the values of MAE, MSE, RMSE, and CC under the encounter angle 120° can be obtained. In addition, similar predictions are conducted for encounter angles 60°, 90°, and 150°. The results of four sets of encounter angles and four sets of hyperparameters, that is, a total of 16 sets of cases are presented in Tables 6–9.

Comparing the prediction results in Figures 4 and 6, as well as the errors in Figures 5 and 7, it can be seen that the prediction accuracy of the AGS algorithm is higher. Besides, the results in Tables 6–9 show that the AGS algorithm achieves dynamic updating of the hyperparameters online and high prediction accuracy in a variety of cases. In particular, for the last set of cases in Table 9, the AGS algorithm has much higher prediction accuracy than the method with fixed hyperparameters. As can be seen from the comparison of results, the AGS method, which can realize the online adjustment of the hyperparameters, has greater advantages over the method of determining the hyperparameters offline.

Another concern in using the AGS method is the time consumption of updating the hyperparameters. The computations in this paper are carried out on a portable computer.

### Table 4: Sample pairs.

| Sample | Input | Output |
|--------|-------|--------|
| Sample 1 | $(x_i, x_{i+1}, \ldots, x_{i+149})$ | $x_{i+150}$ |
| Sample 2 | $(x_{i+1}, x_{i+2}, \ldots, x_{i+150})$ | $x_{i+151}$ |
| ... | ... | ... |
| Sample 250 | $(x_{i+249}, x_{i+1}, \ldots, x_{i+398})$ | $x_{i+399}$ |

### Table 5: 60-step prediction.

| Prediction | Input | Output |
|------------|-------|--------|
| Step 1 | $(x_i, x_{i+250}, x_{i+251}, \ldots, x_{i+399})$ | $x_{i+400}$ |
| Step 2 | $(x_{i+251}, x_{i+252}, \ldots, x_{i+400})$ | $x_{i+401}$ |
| ... | ... | ... |
| Step 60 | $(x_{i+309}, x_{i+310}, \ldots, x_{i+458})$ | $x_{i+459}$ |
Table 6: Results of 60° encounter angle.

| Hyperparameters | Methods       | MAE    | MSE    | RMSE   | CC      |
|-----------------|---------------|--------|--------|--------|---------|
| $C_0 = 50, \sigma_0 = 30$ | Fixed hyperparameters | 0.007150 | 0.000081 | 0.008980 | 0.954416 |
|                 | AGS algorithm  | 0.006439 | 0.000067 | 0.008163 | 0.964379 |
| $C_0 = 10, \sigma_0 = 10$ | Fixed hyperparameters | 0.007827 | 0.000098 | 0.009935 | 0.944652 |
|                 | AGS algorithm  | 0.006812 | 0.000071 | 0.008435 | 0.958232 |
| $C_0 = 100, \sigma_0 = 100$ | Fixed hyperparameters | 0.008033 | 0.000104 | 0.010238 | 0.942802 |
|                 | AGS algorithm  | 0.006741 | 0.000072 | 0.008531 | 0.956149 |
| $C_0 = 1, \sigma_0 = 50$   | Fixed hyperparameters | 0.021885 | 0.000644 | 0.025391 | 0.900614 |
|                 | AGS algorithm  | 0.005859 | 0.000054 | 0.007361 | 0.970615 |

Table 7: Results of 90° encounter angle.

| Hyperparameters | Methods       | MAE    | MSE    | RMSE   | CC      |
|-----------------|---------------|--------|--------|--------|---------|
| $C_0 = 50, \sigma_0 = 30$ | Fixed hyperparameters | 0.067651 | 0.007432 | 0.086210 | 0.477561 |
|                 | AGS algorithm  | 0.020647 | 0.000661 | 0.025714 | 0.967132 |
| $C_0 = 10, \sigma_0 = 10$ | Fixed hyperparameters | 0.066805 | 0.007461 | 0.086378 | 0.471173 |
|                 | AGS algorithm  | 0.020543 | 0.000680 | 0.026091 | 0.965726 |
| $C_0 = 100, \sigma_0 = 100$ | Fixed hyperparameters | 0.074709 | 0.008463 | 0.091996 | 0.287150 |
|                 | AGS algorithm  | 0.028490 | 0.001327 | 0.036436 | 0.921966 |
| $C_0 = 1, \sigma_0 = 50$   | Fixed hyperparameters | 0.076404 | 0.008688 | 0.093211 | 0.242660 |
|                 | AGS algorithm  | 0.024605 | 0.000927 | 0.030461 | 0.949958 |

Table 8: Results of 120° encounter angle.

| Hyperparameters | Methods       | MAE    | MSE    | RMSE   | CC      |
|-----------------|---------------|--------|--------|--------|---------|
| $C_0 = 50, \sigma_0 = 30$ | Fixed hyperparameters | 0.008323 | 0.000106 | 0.010310 | 0.972782 |
|                 | AGS algorithm  | 0.005382 | 0.000047 | 0.006911 | 0.869301 |
| $C_0 = 10, \sigma_0 = 10$ | Fixed hyperparameters | 0.010207 | 0.000160 | 0.012649 | 0.962128 |
|                 | AGS algorithm  | 0.003996 | 0.000024 | 0.004927 | 0.992759 |
| $C_0 = 100, \sigma_0 = 100$ | Fixed hyperparameters | 0.031013 | 0.001293 | 0.035963 | 0.803701 |
|                 | AGS algorithm  | 0.004081 | 0.000027 | 0.005178 | 0.992098 |
| $C_0 = 1, \sigma_0 = 50$   | Fixed hyperparameters | 0.005761 | 0.000054 | 0.007412 | 0.857779 |
|                 | AGS algorithm  | 0.000841 | 0.000001 | 0.001073 | 0.991677 |

Table 9: Results of 150° encounter angle.

| Hyperparameters | Methods       | MAE    | MSE    | RMSE   | CC      |
|-----------------|---------------|--------|--------|--------|---------|
| $C_0 = 50, \sigma_0 = 30$ | Fixed hyperparameters | 0.005382 | 0.000047 | 0.006911 | 0.869301 |
|                 | AGS algorithm  | 0.000841 | 0.000001 | 0.001073 | 0.991677 |
| $C_0 = 10, \sigma_0 = 10$ | Fixed hyperparameters | 0.005761 | 0.000054 | 0.007412 | 0.857779 |
|                 | AGS algorithm  | 0.000855 | 0.000001 | 0.001060 | 0.991886 |
| $C_0 = 100, \sigma_0 = 100$ | Fixed hyperparameters | 0.005766 | 0.000055 | 0.007417 | 0.871271 |
|                 | AGS algorithm  | 0.007998 | 0.000001 | 0.000993 | 0.992928 |
| $C_0 = 1, \sigma_0 = 50$   | Fixed hyperparameters | 0.006464 | 0.000069 | 0.008324 | 0.125924 |
|                 | AGS algorithm  | 0.000911 | 0.000001 | 0.001129 | 0.990783 |

Figure 8: Parameter optimization results for each step using AGS (encounter angle 120°, $C_0 = 10, \sigma_0 = 10$).
computer equipped with an Intel Core i5-8300H (2.30 GHz) processor. The CPU time consumption of the AGS solution is larger, but still within the sampling interval of 0.5 s. It can meet the timeliness requirements of online predictions.

5. Conclusions

This paper proposes a novel AGS-LSSVM algorithm for the online prediction of ship roll motion in waves. The method is based on LS-SVM, combined with the time window method to update the samples set, and uses the AGS algorithm to update the model hyperparameters online, which can realize the prediction of ship roll motion in the next 30 s.

Considering that the computational time of the methods updating hyperparameters offline often exceeds tens or even hundreds of seconds, which cannot meet the requirement of updating parameters online, this paper makes an effort to overcome this shortcoming of offline algorithms and proposes the AGS-LSSVM algorithm. The design idea of the AGS-LSSVM algorithm is as follows: the system state of the ship at the adjacent moments does not change much; thus, the hyperparameters of the current moment can be adjusted based on the hyperparameters of the previous moment. In this paper, instead of using the fixed value of the grid, a grid ratio is used, and the grid for the current moment is obtained by combining the grid ratio and the previous hyperparameters. Through this improvement, the fixed grid is transformed into the auto-moving grid to realize the online adjustment of hyperparameters, and the searching space of hyperparameters is reduced so that the computational time can meet the timeliness requirement.

To illustrate the effectiveness of the AGS algorithm, 16 cases are selected for prediction and the results are compared with those obtained by the method with fixed hyperparameters. From the comparison of the prediction results and the corresponding values of MAE, MSE, RMSE, and CC indices obtained by the two methods, it can be seen that the AGS-LSSVM algorithm realizes the online update of the hyperparameters and improves the prediction accuracy largely. It is an effective method for the online prediction of ship roll motion.

Although only the ship roll motion is predicted in this paper, the proposed method is universal and can be used to predict other modes of ship motion in waves. It can also be extended to online predicting ship maneuvering motion in waves. Besides, in this paper, the offline grid search solving process is decomposed to obtain an online grid search scheme, and this idea can also be used to improve other offline algorithms to evolve them into online algorithms.

However, in this paper, only the simulation data obtained from the equation of ship roll motion are used, and the prediction ability of the proposed method for ship roll motion at real sea still needs to be verified. In addition, the AGS algorithm used in this paper only optimizes the two hyperparameters in LS-SVM, and there is still a lack of effective optimization methods for the hyperparameters in the sample construction, such as the input order of the sample and the length of the time window. These will be the focus of future study.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 51979165).

References

[1] J. Hua, S. Duan, L. Huang et al., “Scale effects in AR model real-time ship motion prediction,” Ocean Engineering, vol. 203, 2020.
[2] N. Suhermi, S. Suhartono, D. D. Prastyo, and B. Ali, “Roll motion prediction using a hybrid deep learning and ARIMA model,” Procedia Computer Science, vol. 144, pp. 251–258, 2018.
[3] J.-C. Yin, Z.-J. Zou, F. Xu, and N.-N. Wang, “An online ship roll motion prediction model based on opportunistic learning and reinforcement learning,” Ocean Engineering, vol. 160, pp. 138–147, 2018.
[4] Z. Yang, Y. Han, and Y.-H. Zhang, “A hybrid AR-EMD-SVR model for the short-term prediction of nonlinear and non-stationary ship motion,” Journal of Zhejiang University-SCIENCE A, vol. 16, no. 7, pp. 2024–2032, 2015.
[5] J.-C. Yin, Z.-J. Zou, and F. Xu, “On-line prediction of ship roll motion during maneuvering using sequential learning RBF neural networks,” Ocean Engineering, vol. 160, pp. 425–437, 2018.
[6] J.-C. Yin, Z.-J. Zou, and F. Xu, “An online prediction method of ship rolling motion based on decomposition strategy (in Chinese),” Journal of Ship Mechanics, vol. 22, no. 8, pp. 915–925, 2018.
[7] J.-C. Yin, Z.-J. Zou, and F. Xu, “On-line prediction of ship roll motion during maneuvering using sequential learning RBF neural networks,” Ocean Engineering, vol. 160, pp. 425–437, 2018.
[8] Z.-J. Zou, and F. Xu, “An online prediction method of ship rolling motion based on decomposition strategy (in Chinese),” Journal of Ship Mechanics, vol. 22, no. 8, pp. 915–925, 2018.
[9] Z.-J. Zou, and F. Xu, “On-line prediction of ship roll motion during maneuvering using sequential learning RBF neural networks,” Ocean Engineering, vol. 160, pp. 425–437, 2018.
[10] Z.-J. Zou, and F. Xu, “On-line prediction of ship roll motion during maneuvering using sequential learning RBF neural networks,” Ocean Engineering, vol. 160, pp. 425–437, 2018.
[11] Z.-J. Zou, and F. Xu, “On-line prediction of ship roll motion during maneuvering using sequential learning RBF neural networks,” Ocean Engineering, vol. 160, pp. 425–437, 2018.
[12] Z.-J. Zou, and F. Xu, “On-line prediction of ship roll motion during maneuvering using sequential learning RBF neural networks,” Ocean Engineering, vol. 160, pp. 425–437, 2018.
[14] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, “Choosing multiple parameters for support vector machines,” Machine Learning, vol. 46, no. 1/3, pp. 131–159, 2002.

[15] J. A. K. Suykens and J. Vandewalle, “Least squares support vector machines classifiers,” Neural Processing Letters, vol. 9, no. 3, pp. 293–300, 1999.

[16] J. Wang, W. Shao, and J. Kim, “Combining MF-DFA and LSSVM for retina images classification,” Biomedical Signal Processing and Control, vol. 60, p. 101943, 2020.

[17] Z. D. Tian, S. J. Li, Y. H. Wang et al., “Mixed-kernel LSSVM predictive control based on improved free search algorithm for nonlinear systems,” Transactions of the Institute of Measurement and Control, vol. 40, no. 16, pp. 4382–4396, 2018.

[18] Z. Tian, “Short-term wind speed prediction based on LMD and improved FA optimized combined kernel function LSSVM,” Engineering Applications of Artificial Intelligence, vol. 91, p. 103573, 2020.

[19] S. Yin, Y. Jiang, Y. Tian, and O. Kaynak, “A data-driven fuzzy information granulation approach for freight volume forecasting,” IEEE Transactions on Industrial Electronics, vol. 64, no. 2, pp. 1447–1456, 2017.

[20] A. J. Smola, B. Schölkopf, and K.-R. Müller, “The connection between regularization operators and support vector kernels,” Neural Networks, vol. 11, no. 4, pp. 637–649, 1998.

[21] X. L. Wang and Z. B. Li, “Identifying the parameters of the kernel function in support vector machines based on the grid-search method,” Journal of Ocean University of China (Natural Science), vol. 35, no. 5, pp. 859–862, 2005.

[22] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” Journal of Machine Learning Research, vol. 13, no. 1, pp. 281–305, 2012.

[23] B. Üstün, W. J. Melssen, M. Oudenhuijzen et al., “Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization,” Analytica Chimica Acta, vol. 544, no. 1-2, pp. 292–305, 2005.

[24] S.-W. Lin, K.-C. Ying, S.-C. Chen, and Z.-J. Lee, “Particle swarm optimization for parameter determination and feature selection of support vector machines,” Expert Systems with Applications, vol. 35, no. 4, pp. 1817–1824, 2008.

[25] J. Wu, X. Y. Chen, H. Zhang et al., “Hyperparameter optimization for machine learning models based on Bayesian optimization,” Journal of Electronic Science and Technology, vol. 17, no. 1, pp. 26–40, 2019.

[26] Z. Tian, “Backtracking search optimization algorithm-based least square support vector machine and its applications,” Engineering Applications of Artificial Intelligence, vol. 94, 2020.

[27] L. Zhou and F. Chen, “Chaotic motion of the parametrically excited roll motion for a class of ships in regular longitudinal waves,” Ocean Engineering, vol. 195, 2019.

[28] B. W. Tian, “Research on mathematical model of ship rolling motion in irregular waves,” Master thesis, Dalian Maritime University, London, UK, 2013.