Online Multiple Outputs Least-Squares Support Vector Regression Model of Ship Trajectory Prediction Based on Automatic Information System Data and Selection Mechanism

JIAO LIU, GUOYOU SHI, AND KAIGE ZHU

Key Laboratory of Navigation Safety Guarantee of Liaoning Province, Navigation College, Dalian Maritime University, Dalian 116026, China

Corresponding author: Guoyou Shi (sgydmu@dlmu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51579025, in part by the Natural Science Foundation of Liaoning Province under Grant 20170540090, and in part by the Fundamental Research Funds for the Central Universities under Grant 3132020134 and Grant 3132020139.

ABSTRACT Existing maritime trajectory prediction models are faced with problems of low accuracy and inability to predict ship tracks in real time. To solve the above problem, an online multiple outputs Least-Squares Support Vector Regression model based on selection mechanism was proposed: (a) converting the traditional Least-Squares Support Vector Regression’s single output to multiple outputs, aiming at the problem that the single-output of the traditional Least-Squares Support Vector Regression model is difficult to apply to complex multiple features prediction scenarios, (b) reducing the high computational complexity of matrix inversion calculations using an iterative solution, in order to solve the problem of poor real-time performance, (c) determining whether to use online model based on the characteristics of different trajectories, and (d) removing initial samples least affecting the model to alleviate the impact of large increases in the number of new samples on computational complexity. The model was simulated using the automatic identification system tracks of Tianjin port in March 2015. The calculation accuracy and efficiency of this model was verified by comparing the predicted results of the proposed model with the recurrent neural network–long short-term memory, back propagation neural network, and traditional Least-Squares Support Vector Regression models. In sum, the proposed model is highly accurate in online and real-time prediction of a target ship’s trajectory when sailing at sea. In particular, it can sustain high prediction accuracy in the case of smaller data samples. The real-time predicted trajectory can assist the generation of ship collision avoidance decision-making.

INDEX TERMS Least-squares support vector regression (LSSVR), online learning, selection mechanism, sliding window, trajectory prediction.

I. INTRODUCTION

The increasing water transportation demand has been significantly linked to rapidly increasing globalization of the world economy. For instance, more than 80% of today’s world merchandise is transported for trade via water routes (which simultaneously requires the transport of massive personnel). Thus, it can be said that water transportation has gradually formed the backbone of international trade. However, increase in water cargo transportation volume, ship density, and complexity of the navigation environment has resulted in certain drawbacks and potential safety hazards. This chapter analyzes the full explanation of background, a detailed explanation of recent findings, the main work and contribution of this article and highlights and originality.

A. BACKGROUND

In the 2019 Annual Overview of Marine Casualties and Incidents [1], the European Maritime Safety Administration claimed that an average of 3239 marine accidents have occurred worldwide every year for the past five years,
with variations from ship capsizing, collision, contacts, fire, and grounding, among others. The most frequent accident among marine accidents is a ship collision, which accounted for 26.3% of the total number. The frequency and serious consequences of collisions have encouraged practitioners and researchers to ascribe significant importance to related research. Various technologies have been developed to prevent ship collisions. In the past five years, the European Maritime Safety Agency has analyzed 4104 marine accidents, attributing 65.8% of accidents to human factors [1]. To reduce the workload of the officer on watch (OOW) and eliminate human factors causing collisions between ships, researchers have directed attention to the development of autonomous navigation systems that can automatically determine collision avoidance solutions and replace human roles. With the rapid development of artificial intelligence technology, the maritime autonomous surface ship (MASS) has received significant attention. The collision avoidance process of MASS includes five parts [2]: an information acquisition module, which contains various sensors, acquires environmental information around the ship and provides support for other modules; a motion prediction module for estimating future trajectory of the target ship; a conflict detection module for detecting collision risks and issuing collision warnings when necessary; a conflict resolution module that determines specific collision avoidance programs; and an execution module to implement solutions. The trajectory prediction algorithm proposed in this study can be used in the motion prediction module. When a ship encounters a potential danger, the predicted target ship’s trajectory is used to determine collision risk and conduct collision detection. Moreover, when the ship wants to determine a conflict resolution, the predicted trajectory can be used to assess collision risk. Collision avoidance decisions depend on the current navigation information of the target ship along with its navigation behavior in the future or at a certain time period in the future, which can be estimated using the trajectory prediction. Such a method can predict and detect different scenarios of ship encounters in advance and generate certain priors for collision avoidance decisions to effectively improve reliability in decision making, reduce collision risk, and ensure navigation safety. Thus, a highly precise and real-time prediction model of ship trajectory can be achieved based on Automatic Identification System (AIS) data.

B. PREVIOUS RESEARCH

Objects used in existing trajectory prediction include robots, pedestrians, balloons, vehicles, aircrafts, and ships. With the increasing number of intelligent autonomous systems in the environment, the ability to perceive, understand, and predict human [3], ship, aircraft, and vehicle behaviors becomes increasingly important. Presently, there are substantial studies on the prediction of vehicle and aircraft trajectories. The application and popularization of the AIS equipment on ships has significantly improved the availability of ship trajectory data such that related research on ship trajectory prediction has become a hotspot. Moreover, owing to the particularity of ship trajectory and a large number of noise points in the ship trajectory data, the prediction of ship trajectory becomes more difficult to organize than that of highways and aircraft trajectories. Different algorithms can be used for trajectory prediction, including traditional mathematical–statistical methods, machine learning methods, heuristic algorithms, and waveform processing algorithms.

Traditional mathematical–statistical methods include the Naive Bayes algorithm, Gaussian process, Markov model, and Gray model, which are currently widely used in the trajectory prediction. Tong et al. [4] improved the traditional Markov model and proposed a ship trajectory prediction method for a curved route based on the Markov chain and Gray model. Cheng et al. [5] comprehensively considered the continuity of a trajectory in time and space and predicted such a trajectory using the improved Markov model, with a prediction accuracy superior than its traditional counterpart. Schreier et al. [6] proposed a trajectory prediction and criticality evaluation model based on an integrated Bayesian method, which is not limited to specific driving situations. Moreover, Qiao et al. [7] proposed a road trajectory prediction model based on the Gaussian mixture model, which overcomes the disadvantages of existing methods.

Meanwhile, the rapid development of machine learning technology has led to the emergence of related algorithms, such as Support Vector Regression (SVR) and Artificial Neural Networks (ANNs), for the gradual application in trajectory prediction. Liu et al. [8] used the Adaptive Chaotic Differential Evolution (ACDE) algorithm to optimize the parameters of the SVR model and proposed a ship trajectory prediction model based on ACDE–SVR and AIS data; however, because the model was offline, it could not be used for real-time prediction. Wang et al. [9] used the motion model as a sequence prediction problem and employed the research progress of deep learning in visual feature extraction and sequence prediction to construct a trajectory predictor based on Recurrent Neural Network–Long Short-Term Memory (RNN–LSTM). Gao et al. [10] proposed a ship trajectory prediction algorithm based on bidirectional RNN–LSTM to solve the problem of generating only offline prediction prevailing in most existing trajectory prediction algorithms and demonstrated an online prediction of ship trajectory. Payeur et al. [11] proposed a method to predict the real-time trajectory of moving objects based on an ANN. Li et al. [12] proposed a human skeleton action recognition method based on the spatio–temporal convolution neural network. Experimental results demonstrated that this method can obtain the same results as the state-of-the-art models in this field.

Heuristic algorithms mainly include certain bionic algorithms, such as genetic algorithm and particle swarm algorithm. Baklacioglu et al. [13] used a genetic algorithm to derive a new aeropropulsive model based on flight manual data of a transport aircraft for accurate trajectory prediction. Yang et al. [14] proposed a prediction algorithm for a local trajectory based on the simulated annealing algorithm and...
particle filtering; however, this algorithm could only predict the local trajectory and could not generate a good prediction for many trajectory groups.

Filtering algorithms are common in the forms of the Kalman and particle filtering algorithms. Perera et al. [15] proposed an extended Kalman filter algorithm to estimate the state of a ship and predict its trajectory. Xu et al. [16] proposed a prediction algorithm of a ship motion trajectory based on improved Kalman filtering and least-squares algorithms to compensate for inaccurate ship trajectory caused by delayed update of AIS data. To overcome the uncertainty and noise of sample data, Park et al. [17] used a combination of a Kalman filter and a neural network to estimate the real-time motion trajectory of an object.

Although the above trajectory prediction methods have demonstrated obvious progress, they were accompanied by a series of shortcomings. First, a ship motion model is required for track prediction based on heuristic algorithm and traditional mathematical–statistical method. Second, the track prediction must satisfy real-time and online prediction requirements to meet decision making for collision avoidance at sea; however, establishing a real-time and accurate ship motion model is often difficult. Moreover, several variables, such as wind, flow, and other environmental factors, complicate the nature of the model, thereby increasing modeling difficulty. Thus, most of these algorithms are only suitable under ideal circumstances. Moreover, the use of track prediction for intelligent collision avoidance at sea requires real-time acquisition of AIS ship data possibly with collision risk for model training. Currently, the amount of sample data that can be acquired is small; thus, the prediction accuracy requirements of the training model using machine learning algorithms, such as a neural network, may not be met. The traditional machine learning algorithm is also a batch offline algorithm, implying that once the model is trained, it cannot be modified. This also implies that the prediction accuracy is significantly reduced when the new data differ from the original one. Moreover, when the new sample data arrive, they can only be merged with the original one to retrain and modify the model. Such a process undoubtedly increases time and space complexity, which makes it difficult to obtain real-time AIS data at sea along with changes in trajectory characteristics.

For better guidance on collision avoidance at sea using the navigation dynamics of the target ship, Liu et al. [8] proposed an offline track prediction model based on ACDE–SVR, which fully utilizes SVR’s advantage of maintaining higher precision in the case of nonlinearity and small samples. However, the model has certain setbacks. First, the SVR model has multiple features inputs and single-feature outputs. When faced with complex systems, a single-output SVR model requires a lengthy training time, which renders it impractical. Second, SVR is a constrained quadratic programming (QP) problem with high computational complexity. Third, it is an offline prediction model, implying that achieving real-time and online prediction of ship trajectory at maritime navigation is not possible. During sailing of a ship on the sea, the training samples are gradually acquired with time and then modeled, after which sudden manipulation instructions may cause the new samples to significantly differ from the trained data. Here, if the previously trained offline model is still used, the accuracy of prediction decreases.

C. THE MAIN WORK AND CONTRIBUTION OF THE ARTICLE

To solve the above problems by improving the prediction accuracy and decreasing the calculation time, this study proposes the Online Multiple Outputs Least-Squares Support Vector Regression model based on Selection Mechanism (SM–OMLSSVR). Unlike the SVR, LSSVR changes the inequality constraint of the original method to an equality constraint that (a) facilitates the solution of Lagrange multiplier, (b) redefines the original solution of the QP problem to a solution of linear equations, (c) reduces the complexity of the problem, and (d) improves the calculation efficiency. Moreover, LSSVR has an improved internal model that displays characteristics of a multiple features output; this allows LSSVR to adapt to situations of predicting the multiple features of the ship’s trajectory during an actual sea voyage. Furthermore, to receive updates regarding the arrival of new samples, online learning is integrated into the LSSVR model, allowing calculation techniques to transform complex matrix inversion problems into iterative solutions, which improves the calculation efficiency. In this model, the SM is used to determine whether the online learning algorithm has been adopted. When the new samples arrive, the original offline MLSSVR model is used to predict and calculate the error. In the case wherein the error does not meet the requirements, the SM–LSSVR model can be used to add the new samples to the training set, thus updating the model. Otherwise, in the case the wherein prediction error meets the requirements, the original offline model continues to predict the next new sample. In the standard SVR algorithm, solving complex QP problems is necessary to obtain the theoretical global optimal solution. Moreover, most Lagrange multipliers are zero so that the final decision function depends only on a small part of the sample data, namely, the Support Vector (SV). Therefore, the solution in the SVR method shows the characteristic of sparsity. In the LSSVR method, because the objective function of the optimization problem uses the error square term and equality constraint, the SVR’s QP problem is converted into a set of linear equations be solved so that the Lagrange multipliers are proportional to the error term. Consequently, the final decision function is related to all samples and the sparsity characteristic of the solution in the SVR method is lost. The problem wherein the LSSVR has no sparsity can be solved using a sliding-window based pruning algorithm. In other words, when the number of samples exceeds the window sample size, a pruning algorithm deletes the initial samples with the least impact on the model. Such a method uses a SM to choose whether to use the online learning
method based on the size of the prediction error, which effectively uses historical training results and avoids repeated sample training. This method also deletes the samples with less impact on the model with time using the pruning algorithm, which reduces the calculation complexity. By adjusting the prediction online model, it can be continuously optimized and its adaptability can be enhanced.

For the traditional LSSVR model, the whole matrix inversion process must consume $O(m^3)$ computational complexity, where $m$ is the number of all samples, because LSSVR model loses the sparsity of the solution. The model proposed herein uses matrix-solving techniques to transform the complex matrix inversion problem into an iterative solution problem. Moreover, the sliding-window-based pruning algorithm can alleviate the sparsity of the solution of traditional LSSVR model to a certain extent. The improved model achieves a significantly reduced time complexity of $O(n)$, where $n$ is the window length. The main contributions of this study are as follows:

(1) A summary of the existing problems of current ship track prediction methods are provided based on the analysis of various trajectory prediction methods and the characteristics of ship trajectory.

(2) The proposed OMLSSVR trajectory prediction model based on AIS data and SM addresses the problems of low accuracy of ship trajectory prediction and inability to predict in real time, combined with the application scenarios for intelligent collision avoidance of ships. This model integrates online learning with the multiple outputs LSSVR model and sets the SM according to the gap between the new samples and original training set, from which it decides whether to use the online learning algorithm. This method can ensure high prediction accuracy under the premise of small samples.

(3) To verify the generalization of the proposed algorithm, six trajectories with different trajectory characteristics are selected for simulation comparison. The simulation results show that the trajectory prediction with different trajectory characteristics can meet the requirements of prediction accuracy. The superiority of the proposed algorithm is verified through simulations with the Back Propagation (BP) neural network, RNN–LSTM, and traditional LSSVR models. The proposed model could maintain high prediction accuracy in the case of small samples as verified by simulation results.

D. HIGHLIGHTS AND ORIGINALITY

The highlights and originality of this article are summarized as follows:

(1) An online multiple outputs Least-Squares Support Vector Regression ship trajectory prediction model based on Automatic Information System data and selection mechanism is proposed;

(2) The proposed model can predict the ship’s future trajectory in real time to assist ship collision avoidance;

(3) In order to verify the efficiency and effectiveness of this model, the prediction results of this model are compared with the more popular RNN-LSTM and BP neural network.

II. ONLINE MULTIPLE OUTPUTS LEAST-SQUARES SUPPORT VECTOR REGRESSION MODEL

Aiming at the shortcomings of the LSSVR algorithm such as high time complexity, difficulty in adapting to complex application scenarios with single output, and only offline prediction, this chapter improves the traditional LSSVR model and proposes an improved OMLSSVR model.

A. THE CLASSIC LEAST-SQUARES SUPPORT VECTOR REGRESSION MODEL

As mentioned above, LSSVR is different from SVR because it changes the inequality constraint into an equality constraint [18]. This model displays certain singularities, which can be resolved by introducing error variables in each sample. Thus, the empirical risk changes from the first power to the second power, thereby reducing the calculation complexity [19], [20]. A training dataset is represented as $(x_i, y_i)$, $i = 1, 2, \ldots, l$, $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, where $x_i$ and $y_i$ are the input and output vectors, respectively. The LSSVR algorithm is described as follows.

The nonlinear mapping $\phi(\cdot)$ is used to map the dataset from the input space to the high-dimensional feature space in which the data points can be fitted using the linear function

$$y(x) = \omega^T \phi(x) + b, \quad (1)$$

where $\omega$ and $b$ are the weight vector and offset, respectively. When comprehensively considering the complexity of the function and the fitting error, the regression prediction problem can be expressed as a constrained optimization problem

$$\min_{\omega, b, e} J(\omega, e) = \frac{1}{2} \|\omega\|^2 + \frac{C}{2} \sum_{i=1}^{l} e_i^2$$

s.t. $y_i = \omega^T \phi(x) + b + e_i, \quad i = 1, 2, \ldots, l, \quad (2)$

where $C$ is the penalty factor used to weigh the relationship between the calculated loss and maximum interval and $e_i$ is the error variable. The problem is changed into an unconstrained problem via the introduction of the Lagrange multiplier $\alpha$ before it is solved. The corresponding Lagrange equation is established as

$$L(\omega, b, e, \alpha) = J(\omega, e) - \sum_{i=1}^{l} \alpha_i \left[ \omega^T \phi(x_i) + b + e_i - y_i \right], \quad (3)$$

where $\alpha_i$ is the Lagrange multiplier for each $x_i$. Considering the derivatives of the variables $\omega, b, e_i$ in (3) and equating these to 0 yields

$$\frac{\partial L}{\partial \omega} = 0 \rightarrow \omega = \sum_{i=1}^{l} \alpha_i \phi(x_i);$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow 0 = \sum_{i=1}^{l} \alpha_i;$$

$$\frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = C e_i;$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow \omega^T \phi(x_i) + b + e_i - y_i = 0 \quad (4)$$
Variables \( e \) and \( \omega \) are eliminated in (4) to yield
\[
\begin{bmatrix}
0 \\
1
\end{bmatrix} = \begin{bmatrix}
1^T \\
K + C^{-1}I
\end{bmatrix}
\begin{bmatrix}
b \\
\alpha
\end{bmatrix} = \begin{bmatrix}
0 \\
1
\end{bmatrix} \begin{bmatrix}
1^T \\
K + C^{-1}I
\end{bmatrix}
\begin{bmatrix}
b \\
\alpha
\end{bmatrix}
\]
\[= \begin{bmatrix}
0 \\
y
\end{bmatrix}, \quad (5)
\]
where \( 1 = (1, \ldots, 1)^T \), \( \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_l)^T \), \( K = \phi(x_i) \cdot \phi(x_j) = K(x_i, x_j) \), \( i, j = 1, 2, \ldots, l \), where \( K(x_i, x_j) \) represents a kernel function that can be directly inner product operated in a high-dimensional feature space, and \( y = (y_1, y_2, \ldots, y_l)^T \). Therefore, for new sample \( x \), the output of the LSSVM model is defined as
\[
f(x) = \omega \phi(x) + b = \sum_{i=1}^{l} \alpha_i K(x_i, x) + b. \quad (6)
\]

### B. Multiple Outputs Least-Squares Support Vector Regression Model

The multiple outputs LSSVR model is an extension of the traditional single-output model. Research on multiple outputs is crucial as many real-life problems contain multiple outputs [21, 22].

The samples are defined as \((x_i, y_i), i = 1, 2, \ldots, l\), \( x_i \in R^m, y_i \in R^n \), where \( x_i \) and \( y_i \) are the input and output vectors, respectively. These samples can be mapped into a high-dimensional feature space by employing a mapping function. Based on the principle of structural risk minimization, LSSVR must solve the planning problem
\[
\min \frac{1}{2} \sum_{j=1}^{k} \| \omega_j \|^2 + \frac{C}{2} \sum_{i=1}^{l} \sum_{j=1}^{k} \epsilon_{ij}^2 + C_0
\]
\[
\text{s.t. } y_{ij} = \omega_j^T \cdot \phi(x_i) + b_j + e_{ij},
\]
\[
\eta_i = \sum_{j=1}^{k} | e_{ij} |, \quad (7)
\]
where \( C \) is the penalty factor of the fitting error in the single-dimension output and \( C_0, \eta_i, \) and \( e \) are the sample’s penalty factor, total fitting error, and output error, respectively. Lagrange multipliers are introduced to construct Lagrange equations to solve the planning problem:
\[
\begin{align*}
L(\omega, b, e, \alpha) &= \frac{1}{2} \sum_{j=1}^{k} \| \omega_j \|^2 + \frac{C}{2} \sum_{i=1}^{l} \sum_{j=1}^{k} \epsilon_{ij}^2 + C_0 \sum_{j=1}^{l} \sum_{j=1}^{k} | e_{ij} | \\
&\quad - \sum_{i=1}^{l} \sum_{j=1}^{k} \alpha_{ij} \left( y_{ij} - \omega_j^T \cdot \phi(x_i) - b_j - e_{ij} \right), \quad (8)
\end{align*}
\]

Similar to the classic model, the derivatives of the variables in (8) are considered and equated to 0:
\[
\frac{\partial L}{\partial \omega_j} = 0 \rightarrow \omega_j = \alpha_{ij} y_{ij} \phi(x_i);
\]
\[
\frac{\partial L}{\partial b_j} = 0 \rightarrow 0 = \sum_{i=1}^{l} \alpha_{ij} y_{ij};
\]
\[
\frac{\partial L}{\partial e_{ij}} = 0 \rightarrow e_{ij} = 0
\]
\[
\frac{\partial L}{\partial \alpha_{ij}} = 0 \rightarrow \alpha_{ij} = \frac{y_{ij}}{\phi(x_i)}
\]

Eliminating \( w \) and \( e \) in (9) yields the matrix
\[
\begin{bmatrix}
0 \\
1
\end{bmatrix} = \begin{bmatrix}
1^T \\
K + C^{-1}I
\end{bmatrix}
\begin{bmatrix}
b \\
\alpha
\end{bmatrix} = \begin{bmatrix}
0 \\
y
\end{bmatrix}, \quad (10)
\]
where \( y^*_j = y_j + \frac{C_0}{l} M_j \), in which \( M_j \) is the column vector of \( l \)-dimension. From (10),
\[
f^{(j)}(x) = \sum_{j=1}^{l} \alpha_{ij} K(x, x_i) + b_j + e_{ij}, \quad j = 1, 2, \ldots, k. \quad (11)
\]

Kernel functions mainly include the linear kernel, polynomial kernel, Gaussian kernel, and sigmoid functions. Among these functions, the Gaussian kernel function is widely used owing to its advantages such as nonlinearity and few parameters. Thus, this function was used in this study for feature mapping:
\[
K(x_1, x_2) = \exp \left( -\frac{\| x_1 - x_2 \|^2}{2\sigma^2} \right), \quad (12)
\]
where \( \sigma \) is the kernel function parameter.

### C. Online Multiple Outputs Least-Squares Support Vector Regression Model

For online learning, the samples increase with time. Online learning is single-incremental or block-incremental. In the former, a sample set \((x_i, y_i), i = 1, 2, \ldots, N\) generates one incremental sample at a time, with time. It can be applied to a time series with equal and moderate time intervals. In the latter, some incremental samples are added to the original dataset each time with time interval \( t \). It is generally used in cases of unequal or equal time intervals with a small time interval. In actual navigation, AIS data are sent at unequal intervals depending on the course and speed of the ship; thus, block-incremental online learning was used in this study. For specific time interval \( t \) (whose value is greater than the AIS data transmission interval), \( m \) new samples are added to the training set as follows.

For the initial sample set of the time series, \( \{x_i(t), y_i(t)\} | i = 1, 2, \ldots, N \} \), where \( x_i(t) \) and \( y_i(t) \) represent all input and output samples, respectively, from the initial recording time to time \( t \). Here, Lagrange multiplier \( \alpha \), offset \( b \), and kernel function matrix \( K(x, x_i) \) in (11) will change with an increment in time; thus, these parameters also become functions of time \( t \) [23]:
\[
\begin{align*}
\alpha(t) &= (\alpha_1, \alpha_2, \ldots, \alpha_l)^T \\
b(t) &= b_t \\
K(t) &= K(x, x_i) \quad i, j = 1, 2, \ldots, t
\end{align*}
\]

Accordingly, the output of the multiple outputs LSSVR in (11) can be redefined as
\[
f^{(j)}(x, t) = \sum_{j=1}^{l} \alpha_{ij}(t) K(x, x_i) + b_j(t), \quad j = 1, 2, \ldots, k. \quad (14)
\]
Simultaneously, (10) can be rewritten in the form
\[
\begin{bmatrix}
0 & 1^T \\
1 & H(t)
\end{bmatrix}
\begin{bmatrix}
b_j(t) \\
\alpha_j(t)
\end{bmatrix}
= \begin{bmatrix}
0 \\
y_j^T(t)
\end{bmatrix}.
\]
(15)

Solving (13) and \( U(t) = H(t)^{-1} \) yields
\[
\begin{cases}
\alpha(t) = U(t) \left[ y^T_j (t) - \frac{1^T U(t) y^T_j (t)}{1^T U(t) 1} \right] \\
b(t) = \frac{1^T U(t) y^T_j (t)}{1^T U(t) 1}.
\end{cases}
\]
(16)

Thus, the key to solve the multiple outputs LSSVR modeling is to determine the inverse of matrix \( H(t) \), i.e., solve matrix \( U(t) \). In the traditional LSSVR, the inverse of matrix \( H(t) \) can be directly solved. However, with progress in time \( t \), more incremental samples are added to the dataset and dimension of the inverse matrix becomes increasingly larger, thereby creating more calculation complexity. According to two lemmas in [24], a matrix technique that avoids direct matrix inversion can be used to subsequently obtain \( U(t) \) by iteration.

Accordingly, from the initial time to time \( t \), the kernel function matrix \( K(t) \) is a square matrix of \( N \times N \) dimension:
\[
K(t) = \begin{bmatrix}
K(x_1, x_1) & \cdots & K(x_N, x_1) \\
\vdots & \ddots & \vdots \\
K(x_1, x_N) & \cdots & K(x_N, x_N)
\end{bmatrix}.
\]
(17)

Based on this, \( H(t) \) is also an \( N \times N \) square matrix:
\[
H(t) = K + C^{-1}I
= \begin{bmatrix}
K(x_1, x_1) + \frac{1}{C} & \cdots & K(x_N, x_1) \\
\vdots & \ddots & \vdots \\
K(x_1, x_N) & \cdots & K(x_N, x_N) + \frac{1}{C}
\end{bmatrix}.
\]
(18)

For \( \{x_{N+i}, y_{N+i}\} | i = 1, 2, \ldots, m \} \), \( m \) samples are added to the initial sample set at time \( t+1 \). Thus, the total number of samples changes from original \( N \) to \( N+m \). At this time, the corresponding kernel function matrix \( K(t) \) will have extra \( m \) rows and \( m \) columns such that it transforms into an \((N+m) \times (N+m)\) square matrix (19), as shown at the bottom of the page. Thus, \( H(t+1) \) is also a square matrix with the same dimension of \((N+m) \times (N+m)\) in (20), as shown at the bottom of the page.

Comparing (18) and (20), \( H(t+1) \) can be written in the block matrix form:
\[
H(t+1) = \begin{bmatrix}
H(t) & V(t+1) \\
V(t+1)^T & Q(t+1)
\end{bmatrix},
\]
where \( V(t+1) \)
\[
= \begin{bmatrix}
K(x_{N+1}, x_1) & \cdots & K(x_{N+m}, x_1) \\
\vdots & \ddots & \vdots \\
K(x_{N+1}, x_N) & \cdots & K(x_{N+m}, x_N)
\end{bmatrix}
\]
and
\[
Q(t+1) = \begin{bmatrix}
K(x_{N+1}, x_{N+1}) + \frac{1}{C} & \cdots & K(x_{N+m}, x_{N+1}) \\
\vdots & \ddots & \vdots \\
K(x_{N+1}, x_{N+m}) & \cdots & K(x_{N+m}, x_{N+m}) + \frac{1}{C}
\end{bmatrix}.
\]
(21)

From the matrix lemma in [21], if the inverse of matrix
\[
A = \begin{bmatrix}
P & Q \\
R & S
\end{bmatrix}
\]
exists, then its inverse \( A^{-1} \) is
\[
A^{-1} = \begin{bmatrix}
P^{-1} & 0 \\
0 & I
\end{bmatrix} + \begin{bmatrix}
-P^{-1}Q \\
I
\end{bmatrix} S^{-1} R^{-1} Q \begin{bmatrix}
-P^{-1}I \\
0
\end{bmatrix},
\]
(22)
which can be manipulated to obtain \( U(t+1) \):
\[
U(t+1) = H(t+1)^{-1} = \begin{bmatrix}
U(t) & 0 \\
0 & I
\end{bmatrix} + \begin{bmatrix}
U(t) \\
0
\end{bmatrix}
\times \begin{bmatrix}
Q(t+1) & -V(t+1)^T U(t) V(t+1) -V(t+1)^T U(t) I
\end{bmatrix},
\]
(23)
where \(0\) represents the zero matrix and \(I\) a unit matrix of \(m \times m\) dimension. An iteration can be performed in (21) to obtain \(U (t+1)\), followed by \(a_j (t)\) and \(b_j (t)\) after block increments. Therefore, time-consuming matrix inversion is avoided, time is saved, and operation efficiency is improved.

### III. TRAJECTORY PREDICTION OF THE ONLINE MULTIPLE OUTPUTS LEAST-SQUARES SUPPORT VECTOR REGRESSION MODEL BASED ON AUTONOMOUS INFORMATION SYSTEM DATA AND SELECTION MECHANISM

This chapter selects some features in the pre-processed AIS data to describe the sailing conditions of the ship, and uses it as the input data of the model for model training. The selection mechanism of whether to use online learning is set according to the error threshold, and the sliding window method is used to delete some samples, and an OMLSSVR ship trajectory prediction model based on AIS data and SM is established.

#### A. AIS DATA PREPROCESSING

1) DATA CLEANING

During signal reception or analog-to-digital conversion, AIS generates a large number of illogical abnormal points. These abnormal points mainly show that the longitude, latitude, course, or speed of the ship exceeds the reasonable range. They also show the abnormal position of a certain position point, length of Maritime Mobile Service Identity (MMSI) that is not 9 bits, and other navigational information. Moreover, AIS data have numerous missing values. Such an abnormality or loss negatively affects subsequent track prediction. Fig. 1 shows the original AIS trajectory of Tianjin port in March 2015. The figure shows numerous messy trajectory segments that partly cross the land. To ensure the quality and accuracy of trajectory prediction, AIS data must be cleaned, abnormal points must be identified and removed, and missing values must be reconstructed.

![FIGURE 1. Schematic of the original AIS trajectory of Tianjin port in March 2015. The arrows and red boxes indicate that the area has a trajectory across the land.](image)

Abnormal and missing values of AIS, which reduce the accuracy of the data, are attributed to (a) AIS equipment failure, (b) loss of stability of the receiver when receiving global positioning system signals, (c) signal transmission interference, and (d) network connection error [25]. To ensure that the accuracy of the subsequent trajectory prediction was not affected, correct data were used as input and cubic spline interpolation [26] was used for interpolating abnormal data to complete the reconstruction of the missing values. Furthermore, there are numerous extremely short trajectories in the original trajectory set of AIS. They are often concentrated in a small area or the sampling time interval is extremely long to reflect the true trajectory of continuous ship movement. Therefore, herein, the threshold for the number of position points in the trajectory was set to 50, i.e., the ship trajectory with less than 50 trajectory points is considered to incompletely reflect the ship’s motion characteristics and thus was deleted [27].

2) DATA COMPRESSION

Based on the different states of the ships, AIS data are sent at intervals of two seconds to several minutes. In sea areas with heavy traffic, hundreds of ships may pass every day, resulting in a huge amount of AIS data. In this study, AIS data of Tianjin port water area were considered as the sample AIS dataset. Every ship has a unique MMSI. Thus, different ship trajectories are divided based on their MMSI identifiers, after which the OMLSSVR algorithm based on SM can be used for simulation modeling of track prediction. Technically, the huge amount of AIS data creates enormous difficulties in terms of data storage, partition, and modeling, which necessitates an appropriate compression algorithm to compress the AIS trajectory data.

Huang et al. [28] realized the compression and visualization of ship trajectory based on Douglas Peucker (DP) and Kernel density estimation (KDE) algorithms, respectively. Further, they significantly accelerated the AIS trajectory compression process using the large-scale parallel computing function of GPU architecture. Li et al. [29] proposed a time-series clustering model based on the improved adaptive constrained dynamic time warping (ACDTW) algorithm and applied it to marine traffic ship modeling. The simulation results indicated that the ACDTW algorithm outperforms the DP and KDE algorithms. Gao et al. [30] proposed an online compression model of ship AIS trajectory data based on the improved sliding window algorithm, in which they recommended using one-angle and three-distance thresholds. With this model, users can choose the appropriate compression threshold based on their needs for real-time and online compression. Compared with the simulation experiment of the offline DP algorithm, Gao et al.’s model achieved a higher compression efficiency, along with a ship trajectory that was not easily distorted after processing. Thus, this model was used herein for AIS trajectory compression.

#### B. DESCRIPTION OF SHIP NAVIGATION CHARACTERISTICS

The trajectory points of a ship are the set of its position points during voyage. Further, the ship trajectory is the set of curves formed by successively connecting the ship trajectory points in a very small time interval. The ship course, speed, longitude, latitude, and time interval are all navigational information most closely related to the ship’s position.
When a ship sails on the sea according to the planned route and needs to alter its course to avoid obstacles, the OOW will steer a rudder angle according to the current situation and the International Regulations for Preventing Collisions at Sea (COLREGS). Thereafter, the ship’s course changes according to the steering direction and rudder angle, and the ship deviates from the preset route. When required to slow down or stop to avoid obstacles, the ship is unable to reach a certain position at the original time according to the planned route. The extent of ship navigation under the current course and speed is determined by the time interval, which also affects the ship’s position. These factors closely related to the ship’s position can influence the ship’s trajectory prediction. Thus, the ship’s future position can be predicted based on the ship’s historical position (longitude and latitude), course and speed, and time interval. The calculation formula for the first type of dead reckoning (DR) can verify the influence of these parameters on the ship’s trajectory: the estimated destination \((\phi_2, \lambda_2)\) can be calculated by determining the starting position \((\phi_1, \lambda_1)\), constant course \(c\) (°), and voyage \(s\) (n miles). The two most commonly used DR methods are middle-latitude and Mercator’s sailing methods. The calculation formula for the middle-latitude sailing method is

\[
\begin{align*}
\phi_2 &= \phi_1 + s \cos c / 60 \\
\lambda_2 &= \lambda_1 + s \sin c \cdot \sec \left( \frac{\phi_1 + \phi_2}{2} \right) / 60,
\end{align*}
\]

and that for the Mercator’s sailing method is

\[
\begin{align*}
\phi_2 &= \phi_1 + s \cos c / 60 \\
\lambda_2 &= \lambda_1 + (MP_2 - MP_1) \cdot \tan c,
\end{align*}
\]

where \(MP_1\) and \(MP_2\) are the difference in meridional parts of the starting position and estimated destination, respectively.

The AIS data include statics, dynamics, voyage-related, and navigation-safety information. The MMSI in the AIS static data is used to identify different ships to describe the trajectory characteristics of different ships at a certain time. Moreover, the longitude \(lon_t\), latitude \(lat_t\), course \(c_t\), speed \(speed_t\), and time stamp \(time_t\) in dynamic information are used to describe the navigation characteristics of the ship. The ship’s behavioral characteristics at time \(t\) can be expressed as

\[
B_t = \{mmsi, lon_t, lat_t, course_t, speed_t, time_t\}. \quad (26)
\]

In an actual voyage, the target ship’s characteristic values of current and past time trajectories are used as historical sample data to predict its trajectories in the future. Moreover, the ship’s \(B_t\) at the current time \(t\) and past \(n\) times, as well as the time stamp at the next time, are used as input data for the model. Further, the longitude and latitude at the next time are used as the output data. The input and output datasets can be expressed as

\[
\begin{align*}
I_{input} &= \{B_t, B_{t-1}, \ldots, B_{t-n-1}, time_{t+1}\}; \quad O_{output} \\
&= \{lon_{t+1}, lat_{t+1}\}. \quad (27)
\end{align*}
\]

C. ERROR THRESHOLD OF THE ONLINE MULTIPLE OUTPUTS LEAST-SQUARES SUPPORT VECTOR REGRESSION MODEL BASED ON SELECTION MECHANISM

Certain differences between the training and new samples of ship trajectories exist during actual sea navigation. Some training samples have strong representativeness, and the trained model exhibits high generalization, which can accurately predict the new samples. Conversely, some training samples have weak trajectory representations. These samples can only be improved through online models by adding new samples to the training dataset, updating the model in real time, and then performing trajectory prediction. Therefore, a screening mechanism for selective online learning must be set up to reduce online learning and improve the learning efficiency by meeting the prediction accuracy requirements. Before their online learning, the samples are trained to construct an offline multiple outputs LSSVR model that cannot be changed after the completion of the training. The model is used to predict the tracks of new samples and calculate the error index, which becomes the threshold for determining whether online learning must be used. If the error is less than the threshold, the new trajectory samples are considered to minimally impact the learning effect of the multiple outputs LSSVR model and no new information has been generated; thus, online learning is not required. If the error exceeds the threshold, the online learning algorithm is used to update the model.

In this study, the error of the training samples was set as the screening criteria regarding whether to use online learning. The training sample set was divided into 10 subsets. The data of each subset were used as the test set, whereas the remaining \(k-1\) subsets were used as the training set to obtain the \(k\) models. The average prediction error of the test set of these \(k\) models was regarded as the error of the total training samples and similarly considered as the error threshold of trajectory prediction. Based on the definition of geographical coordinates, the loss function was defined as the error threshold

\[
e = \frac{1}{n} \sum_{i=1}^{n} \sqrt{(y_{lon} - y_{lon})^2 + (y_{lat} - y_{lat})^2}, \quad (28)
\]

where \(y_{lon}\) and \(y_{lat}\) are the predicted values of the longitude and latitude, respectively, and \(Y_{lon}\) and \(Y_{lat}\) are their actual counterparts.

D. SETTINGS FOR THE SLIDING WINDOW

According to (21), the online multiple outputs LSSVR can well solve the inverse operation of high-dimensional matrices. However, continuous addition of new samples enlarges the sample pool, widens the matrix dimension, and complicates the matrix operation, which increases the prediction time and reduces the prediction efficiency. Thus, some samples may not be required at the same time as online learning. Thus, when new samples are added, some samples with low importance can be deleted to maintain the prediction accuracy and retain the total sample length in a certain range.
This technique significantly reduces the calculation complexity and improves the efficiency and real-time performance of prediction.

The sample deletion process removes the samples based on their importance. For instance, the ship’s navigation track at the next moment has the greatest importance affected by navigation behavior at the latest moment. Thus, the earliest added samples are less important for the current trajectory prediction. This operation describes the mechanism of the sliding moving window algorithm, as illustrated in Fig. 2. Here, the training sample of the model slides like a window. The total length of the window is constant, which slides from one position to another with time \( t \). Assuming that the original sample is \( S_{\text{origin}} = \{S_1, S_2, S_4, S_5, \ldots, S_l-1, S_l\} \), adding the new sample \( S_{l+1} \) slides the sample window from position 1 to position 2, adding \( S_{l+2} \) slides the window from position 2 to position 3, and so on, until no new sample is added.

\[
\text{FIGURE 2. Schematic of the sliding window.}
\]

Assuming that the sample size is \( l \), the sliding window length is equal to 1. Based on this rationale, the training sample set can be expressed as \( \{x(t), y(t)\} \), where \( x(t) = [x_t, x_{t+1}, \ldots, x_{t+l-1}] \), \( y(t) = [y_t, y_{t+1}, \ldots, y_{t+l-1}] \), \( x_t \in \mathbb{R}^m \), \( y_t \in \mathbb{R}^n \). At this time, the iteration matrix can be expressed as

\[
H(t) = \begin{bmatrix} f(t) & F(t) \\ F(t)^T & W(t) \end{bmatrix},
\]

where \( f(t) = K(x_t, x_t) + \frac{1}{C} \),

\[
F(t) = [K(x_{t+1}, x_t), K(x_{t+2}, x_t), \ldots, K(x_{t+l-1}, x_t)],
\]

and

\[
W(t) = \begin{bmatrix} K(x_{t+1}, x_{t+1}) + \frac{1}{C} & \cdots & K(x_{t+l-1}, x_{t+1}) \\ \vdots & \ddots & \vdots \\ K(x_{t+1}, x_{t+l-1}) & \cdots & K(x_{t+l-1}, x_{t+l-1}) + \frac{1}{C} \end{bmatrix}.
\]

Note that \( W(t) \) is a square matrix with \((l-1) \times (l-1)\) dimension. At time \( t+1 \), new samples \( \{x_{t+1}, y_{t+1}\} \) are added and the earliest samples \( \{x_t, y_t\} \) are deleted. At this time, the iteration matrix can be expressed as

\[
H(t+1) = \begin{bmatrix} K(x_{t+1}, x_{t+1}) + \frac{1}{C} & \cdots & K(x_{t+l-1}, x_{t+1}) \\ \vdots & \ddots & \vdots \\ K(x_{t+1}, x_{t+l-1}) & \cdots & K(x_{t+l-1}, x_{t+l-1}) + \frac{1}{C} \end{bmatrix}
\]

\[
= \begin{bmatrix} W(t) & V(t+1) \\ V(t+1)^T & V(t+1) \end{bmatrix},
\]

where \( V(t+1) = [K(x_{t+1}, x_{t+1}), K(x_{t+1}, x_{t+2}), \ldots, K(x_{t+l-1}, x_{t+1})]^T \) and \( v(t+1) = K(x_{t+1}, x_{t+1}) + \frac{1}{C} \).

According to two lemmas in [21], \( H(t+1)^{-1} \) can be iteratively solved using \( H(t)^{-1} \). The procedure is similar to online learning; thus, it has not been discussed here. Therefore, when new samples are added to the sample set, the earliest samples are deleted from the sample set and current samples are quickly updated and iterated.

E. ONLINE MULTIPLE OUTPUTS LEAST-SQUARES SUPPORT VECTOR REGRESSION TRAJECTORY PREDICTION MODEL BASED ON AUTOMATIC INFORMATION SYSTEM DATA AND SELECTION MECHANISM

In this study, an OMLSSVR trajectory prediction model based on AIS data and SM was proposed to predict the trajectory of the target ship. The AIS data were first received, cleaned, and compressed to obtain the key points of the trajectory. These key points were then considered as training samples to establish the offline multiple outputs LSSVR track prediction model. When new samples were added, the corresponding error was assessed. If the error exceeded the threshold, online learning performed. Pruning algorithm was used to delete the earliest samples to ensure consistent length of sample window. Otherwise, the offline model was used to predict and determine the next new samples.

Fig. 3 shows a flow chart of the OMLSSVR track prediction model based on AIS data and SM. The specific steps of the flow chart are provided below:

Step 1: The AIS sample data are cleaned and their trajectory compressed; the initial sliding window length \( l \) is set. \( l \) samples are used to train the offline multiple outputs LSSVR model and construct the AIS trajectory prediction model. \( f^{(i)}(x) = \sum_{j=1}^{l} \alpha_j K(x, x_j) + b_j \), \( j = 1, 2, \ldots, k \). The error of all samples is calculated using (28), which is then regarded as the error threshold, \( e_{\text{threshold}} \).

Step 2: Once \( m \) new samples \( \{x_i, y_i\} \) have arrived, the latest trained model is used to predict these samples and obtain the predicted value. The prediction error \( e_i (i = 1, 2, \ldots, m) \) of the new samples in this batch is calculated using (28).

Step 3: The prediction error \( e_i (i = 1, 2, \ldots, m) \) of each sample is compared with \( e_{\text{threshold}} \). If \( |e_i| < e_{\text{threshold}} \), online learning is not needed and we can immediately proceed to Step 4. If \( |e_i| \geq e_{\text{threshold}} \), the new sample is added to the current sample set, the original sample is deleted for online learning, and the model is updated.

Step 4: The value of \( i \) is determined against \( m \). If \( i = m \), then Step 5 can immediately follow. Otherwise, let \( i = i+1 \) and Step 3 is repeated.

Step 5: If \( i = t+1 \), Step 2 is repeated until a collision avoidance decision is established, which implies that the target ship is clear to pass.

As the sampling continues, the sample data are continuously updated. To maximize the information contained,
each online training is considered as a combination of new and old samples with a fixed number of windows. Thus, compared with offline prediction models, the credibility of the online model’s prediction is enhanced.

The advantages of this method are summarized as follows:

(a) This method is suitable for small-sample application scenarios. Being a variant of the SVR algorithm, LSSVR possesses the advantages of SVR algorithm and exhibits good nonlinear prediction performance. Moreover, it can maintain high prediction efficiency in the case of limited samples, which is suitable for application scenarios involving the characteristics of small samples. LSSVR changes the inequality constraint in the SVR algorithm to equality constraint, and the original solution of QP problem is redefined as the solution of linear equations, which improves the operation efficiency.

(b) The solving efficiency of the model is high. The traditional single-output model of LSSVR is improved to a multiple outputs model to adapt to the characteristics of the multiple features outputs of trajectory prediction. Moreover, the complex matrix-solving problem in the model is transformed into an iterative solution problem using computational skills, further improving the efficiency of the matrix solution.

(c) Real-time prediction can be achieved using the proposed method. The online learning algorithm is combined with the multiple outputs LSSVR model, and the SM determines whether to use the online learning algorithm, thus ensuring the prediction accuracy in the case of the AIS data flow. Furthermore, the total sample size is determined using the pruning algorithm, which prevents the adverse impact of the growing sample on the real-time calculation.

IV. SIMULATION

This chapter describes the simulation experiment from three aspects: the description of the experimental data, the model parameter setting and the experimental process.

A. DESCRIPTION OF EXPERIMENTAL DATA

The training samples comprised 2346643 AIS data obtained from 8207 ships in Tianjin port in March 2015. Fig. 1 shows the original AIS trajectory in March 2015, and Table 1 lists the sample data structure. After data cleaning and track compression, the AIS trajectory transforms as shown in Fig. 4. During sea voyage when a ship and the target ship are at collision risk, the AIS data of the target ship were acquired in real time for model training. In the simulation, the ship trajectory is randomly extracted as a sample training model.

### TABLE 1. Sample data structure.

| MMSI   | Longitude (°) | Latitude (°) | Course (°) | Speed (kn) | Timestamp (s) |
|--------|----------------|---------------|------------|------------|---------------|
| 412287 | 117.73948      | 38.50701      | 274.4      | 2.5        | 14250391      |
| 412287 | 117.73895      | 38.51507      | 315.5      | 2.6        | 14260400      |
| 412287 | 117.72696      | 38.5207       | 296        | 2.1        | 14260410      |
| 413563 | 118.45722      | 38.94535      | 241.8      | 0          | 14271787      |
| 006    |                |               |            |            | 38            |
| 538004 | 120.01953      | 38.71058      | 279.1      | 11.9       | 14262849      |
| 758    |                |               |            |            | 21            |
| 538004 | 119.93421      | 38.72084      | 278.2      | 11.8       | 14262861      |
| 758    |                |               |            |            | 60            |
| 538004 | 119.88676      | 38.72634      | 278.5      | 11.8       | 14262868      |
| 758    |                |               |            |            | 73            |

The ship trajectory with MMSI 412326162 was randomly extracted, as shown by the track in Fig. 5, where the direction of the black arrow indicates the ship movement direction. Overall, 59 track points were recorded. Of these, 60% (35 points) were considered to represent the original sample set and 40% (24 points) the new dataset. In Fig. 5,
FIGURE 4. AIS trajectory of ship entering and leaving the water near Tianjin port after preprocessing in March 2015.

FIGURE 5. Ship trajectory with MMSI 412326162. The direction of the arrow indicates the ship’s navigation direction.

the fifth point in the online dataset showed a large range of altering course; thus, accurate prediction was difficult after this point.

The target ship’s track characteristic data at the present time and past nine times, as well as the time stamp at the next time, were used as input data to predict the ship’s position at the next time. The specific input and output data were as follows:

\[
I_{input} = \{B_{t}, B_{t-1}, \ldots, B_{t-9}, \text{time}_{t+1}\};
\]

\[
O_{output} = \{\text{lon}_{t+1}, \text{lat}_{t+1}\}. \quad (31)
\]

B. PARAMETERS SET UP

The original sample set was considered as the training sample of the offline model, whereas the total number of the original sample set, 35, was considered as the length of the sliding window. The online dataset provided new samples, and a batch of samples was added at each iteration. In addition to the online learning algorithm, three other parameters in multiple outputs LSSVR algorithm with significant influence on the model prediction results were considered, i.e., penalty factor \( C \), insensitivity factor \( \varepsilon \), and kernel function parameter \( \sigma \). To improve the prediction accuracy of the LSSVR model, ACDE algorithm was used to optimize the parameters of the offline model [8]. The ACDE algorithm is a variant of the heuristic DE algorithm. It uses the adaptive scaling factor instead of the fixed scaling factor, which compensates for DE algorithm’s lack of the search ability attributed to the fixed scaling factor. Moreover, the introduction of the chaos theory reduces the search time of the differential evolution algorithm and solves the problem wherein the population is caught in premature owing to the weak local-search ability in the later period of the traditional algorithm. The ACDE algorithm parameters were set according to the setting basis proposed in [8]: the population size was set to 50, maximum number of iterations was 150, scaling factors were 1.2 and 1.3, and crossover probability was 0.7. Moreover, (28) was used as the fitness function. After the parameters of the ACDE algorithm were set, the parameters of the model proposed in this study were optimized using the results listed in Table 2.

| Parameter name          | Penalty factor \( C \) | Insensitivity factor \( \varepsilon \) | Kernel function parameter \( \sigma \) |
|-------------------------|------------------------|--------------------------------------|--------------------------------------|
| Optimization results    | 46.3142                | 0.0271                               | 10.4520                              |

C. EXPERIMENTATION

K-fold cross validation was used to calculate the error, as well as the error threshold, of the sample dataset. The sample was divided into 10 subsets, each tested once, and the remaining \( k-1 \) subsets were used to train the models. A total of 10 models were obtained. The average value of the prediction errors of the test sets of these \( k \) models was regarded as the error of the total sample set and trajectory prediction error threshold, \( \varepsilon_{\text{threshold}} = 3.4417 \times 10^{-5} \). The new dataset samples were considered as new arrival samples and divided into batches with a 5-min interval. The new samples were the test set to predict the trajectory and calculate the error \( e_i \), which was then compared with \( \varepsilon_{\text{threshold}} \). If \( e_i > \varepsilon_{\text{threshold}} \), this sample point was added to the original data set and the first sample point was deleted. The sample set was then used for online learning and to update the model. Otherwise, online learning was not required; thus, the model was not updated. Subsequently, the model was used to predict the next trajectory point and calculate the prediction error. The iterative calculation was performed as shown in Fig. 2 until the track prediction was completed.

V. RESULTS AND DISCUSSION

In order to verify the accuracy and efficiency of the proposed algorithm, the prediction error and running time of the proposed algorithm are compared with the offline multiple outputs LSSVR model. At the same time, in order to verify the generalization of the proposed algorithm, this chapter also selects several sample tracks with different characteristics to train and test the model. In order to further verify the superiority of the proposed model, the prediction results of this algorithm are compared with other common prediction algorithms (RNN-LSTM, BP neural network, etc.).
TABLE 3. New sample errors of ships with MMSI of 412326162 using the offline model and the proposed online model.

| Serial number of new sample batch | Serial number of new samples | Prediction error of the proposed model $e(^\circ)$ | Error threshold $e_{threshold}(^\circ)$ | Whether online learning is required | Prediction error of the offline model $e(^\circ)$ |
|----------------------------------|-------------------------------|--------------------------------------------|-------------------------------------|-----------------------------------|---------------------------------------------|
| 1                                | 1                             | $2.2865 \times 10^{-2}$                    | $3.4417 \times 10^{-5}$           | No                                | $2.2865 \times 10^{-2}$                    |
| 2                                | 2                             | $2.4177 \times 10^{-2}$                    | $3.4417 \times 10^{-5}$           | No                                | $2.4177 \times 10^{-2}$                    |
| 3                                | 3                             | $2.1423 \times 10^{-2}$                    | $3.4417 \times 10^{-5}$           | No                                | $2.1423 \times 10^{-2}$                    |
| 4                                | 4                             | $3.1452 \times 10^{-2}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.1452 \times 10^{-2}$                    |
| 5                                | 5                             | $3.3541 \times 10^{-2}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.3541 \times 10^{-2}$                    |
| 6                                | 6                             | $4.2145 \times 10^{-3}$                    | $3.4417 \times 10^{-5}$           | Yes                               | $4.2145 \times 10^{-4}$                    |
| 7                                | 7                             | $1.4134 \times 10^{-4}$                    | $3.4417 \times 10^{-5}$           | Yes                               | $6.0431 \times 10^{-4}$                    |
| 8                                | 8                             | $7.9470 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | Yes                               | $8.1274 \times 10^{-4}$                    |
| 9                                | 9                             | $4.4134 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | Yes                               | $8.9452 \times 10^{-4}$                    |
| 10                               | 10                            | $3.0124 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $7.2546 \times 10^{-4}$                    |
| 11                               | 11                            | $3.3860 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $5.1425 \times 10^{-4}$                    |
| 12                               | 12                            | $2.3416 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $1.0253 \times 10^{-4}$                    |
| 13                               | 13                            | $2.9412 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $6.4521 \times 10^{-3}$                    |
| 14                               | 14                            | $3.1108 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.9654 \times 10^{-5}$                    |
| 15                               | 15                            | $3.4314 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.4215 \times 10^{-5}$                    |
| 16                               | 16                            | $2.9643 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.9152 \times 10^{-5}$                    |
| 17                               | 17                            | $3.1419 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.4412 \times 10^{-5}$                    |
| 18                               | 18                            | $3.1129 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.1450 \times 10^{-5}$                    |
| 19                               | 19                            | $2.6742 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $2.9452 \times 10^{-5}$                    |
| 20                               | 20                            | $3.0019 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.4502 \times 10^{-5}$                    |
| 21                               | 21                            | $3.2417 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.4577 \times 10^{-5}$                    |
| 22                               | 22                            | $2.6920 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.0470 \times 10^{-5}$                    |
| 23                               | 23                            | $3.3142 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $3.8452 \times 10^{-5}$                    |
| 24                               | 24                            | $3.2415 \times 10^{-5}$                    | $3.4417 \times 10^{-5}$           | No                                | $4.1452 \times 10^{-5}$                    |
| Average value                    | /                             | $5.5846 \times 10^{-4}$                    | /                                  | /                                 | $1.9421 \times 10^{-4}$                    |

A. ERROR COMPARISON BETWEEN THE ONLINE MULTIPLE OUTPUTS LEAST-SQUARES SUPPORT VECTOR REGRESSION MODEL BASED ON SELECTION MECHANISM AND THE OFFLINE MULTIPLE OUTPUTS LEAST-SQUARES SUPPORT VECTOR REGRESSION MODEL

To compare the prediction accuracy of the online and offline models, the same sample data used in the SM–OMLSSVR model were used to train its offline model counterpart. The original sample set was used as the training set of the offline model, whereas the new sample set was the test set. The same model parameters were used to predict the ship trajectory.

Table 3 highlights the prediction error obtained using the proposed online model and the offline model. For the offline model, the first five points yielded a prediction error that was less than the error threshold. Thus, online learning was not required to update the model. At the fifth trajectory point, the ship sharply altered its course. Thus, when the original offline model was used for the sixth point, the prediction error was large and exceeded the error threshold. The online algorithm should be used here to add the sample to the original sample set, delete the first sample in the original sample set, and conduct online learning to update the LSSVR model, followed by the prediction of the subsequent location data points. Moreover, the offline model prediction error exceeded the error threshold until the ninth position; thus, the online learning algorithm was needed to continuously update the model until the tenth point, after which the prediction errors were less than the threshold. At this point, it was unnecessary to update the model; the model updated by the ninth point would be continuously used for the subsequent trajectory prediction. Therefore, online learning was not required in the first five points for the proposed model and the offline model, indicating that both models have equal prediction errors. However, to predict the ship’s subsequent positions, the proposed model used an online learning algorithm, added new sample points, removed the same number of original samples from the initial position of the sample set, and constantly updated the offline multiple outputs LSSVR model. This resulted in visible variation in the prediction errors between both models. The average error of the proposed model was $5.5846 \times 10^{-5}$, while that of the offline model was $1.9421 \times 10^{-4}$, which shows that online learning significantly improved the accuracy of the model.

Figs. 6 and 7 show a comparison of the predicted and actual trajectories obtained using the online multiple outputs LSSVR model based on SM and the offline multiple outputs LSSVR model, respectively. Areas with large errors were locally enlarged. The online model exhibited significantly improved the prediction accuracy of the trajectory points with large heading changes.

The main reason for the significantly improved accuracy of the proposed SM–OMLSSVR model relative to the traditional LSSVR model is the introduction of the online learning mechanism. Aiming at the influx of the AIS data stream, the model can be simultaneously predicted and learned. To verify the reliability of the model and ensure further credibility of the simulation results, a goodness-of-fit test and a model significance test were conducted. The former was used to evaluate the fitting degree of the regression equation to the sample data. The latter was used to determine whether
the relation between the explained and explanatory variables in the model was significantly and fully established through hypothesis testing, including the linear relation test of the regression equation and the test of the significance of the regression coefficient.

Establishing the coefficient of determination $R^2$ was required to examine the goodness of fit of the proposed model. The closer the statistical value is to 1, the higher is the goodness of fit of the model. The $R^2$ obtained in this model is 0.9876, which is close to 1. Hence, the input and output features of the model are highly correlated.

The significance test mainly included the F- and t-tests. If the variable is significant, the regression coefficient should differ significantly from 0. The statistical value $t$ in the t-test follows the $t$ distribution with a degree of freedom $(n-k-1)$, where $k$ is the number of explanatory variables in the model and $n$ is the sample size. Thus, the parameters in model (11) were tested to determine whether they differed significantly from 0. The original hypothesis $H_0$ was proposed as $\alpha_{i,j} = b_j = 0$, $(i, j = 1, 2, \ldots, n)$, and the calculated F statistic $f > F_\alpha$ value was 637.1612. The significance level $\alpha$ was selected as 0.05. Based on the F-distribution table, the critical value $F_{\alpha}(2, 32)$ of 3.302 was obtained. Because $f > F_{\alpha}$, the original hypothesis was rejected under the confidence probability of 0.95, i.e., the influence of the explanatory variables on the explained variables was significant.

Next, the t-test was used to test the significance of the regression coefficient. If the variable is significant, the regression coefficient should differ significantly from 0. The statistical value $t$ in the t-test follows the $t$ distribution with a degree of freedom $(n-k-1)$. The t statistic value in this model was 36.4150, and the significance level $\alpha$ was selected as 0.05. Based on the t-distribution table, the critical value $T_{\alpha}(0.025, 32)$ of 2.0369 was obtained. Because $t > T_{\alpha}$, the original hypothesis was rejected under the confidence probability of 0.95, i.e., the influence of the explanatory variables on the explained variables was significant.

B. RUNNING TIME COMPARISON BETWEEN THE ONLINE MULTIPLE OUTPUTS LEAST-SQUARES SUPPORT VECTOR REGRESSION MODEL BASED ON SELECTION MECHANISM MODEL AND THE OFFLINE MULTIPLE OUTPUTS LEAST-SQUARES SUPPORT VECTOR REGRESSION MODEL

To further verify that the SM–OMLSSVR significantly reduces the computational complexity of the traditional LSSVR model, ship trajectories with different numbers of trajectory points were randomly extracted. The SM–OMLSSVR algorithm and the LSSVR algorithm were used to predict the extracted tracks, and the running time and average prediction error were also recorded. It should be explained that the run time here refers to the time required to train the model. The results are shown in Table 4, where it is seen that the running time of the SM–OMLSSVR model was significantly reduced compared with that of the traditional LSSVR model with standard inversion. Moreover, the running time reduction rate increased with increasing number of samples. This suggests that unlike the operation efficiency of the standard LSSVR that becomes very poor with increasing data samples up to a certain scale, the operation efficiency for the online LSSVR algorithm considerably improved with more samples. Based on the table, the prediction error of SM–OMLSSVR was high but that of LSSVR was obviously higher, with higher number samples. In particular, for two ships with MMSI of 413018020 and 33842000, the prediction error of the traditional LSSVR algorithm increased to the order of $10^{-4}$.

| MMSI       | Number of samples | Run time (s) | Predicted error (%) |
|------------|------------------|--------------|---------------------|
| 412332     | 65               | 3.1457       | 3.1456 x 10^{-5}    |
| 165        |                  | 13.527       | 6.5921 x 10^{-5}    |
| 413018     | 124              | 5.4210       | 7.1452 x 10^{-5}    |
| 020        |                  | 23.545       | 4.5241 x 10^{-5}    |
| 240385     | 205              | 6.9543       | 4.0171 x 10^{-4}    |
| 000        |                  | 30.242       | 8.2419 x 10^{-4}    |
| 338420     | 112              | 7.6324       | 6.5421 x 10^{-4}    |
| 000        |                  | 41.254       | 2.0415 x 10^{-4}    |

The table shows the comparison of the run-time and predicted error between SM–OMLSSVR and standard inversion LSSVR models for prediction of four trajectories with different sample sizes. Among them, run time refers to the model training time.
Compared with the traditional LSSVR model, the improvement in the SM–OMLSSVR model’s prediction accuracy was primarily attributed to the introduction of the online learning mechanism. Further, the improvement in computing efficiency was attributed to the following reasons. First, the traditional LSSVR algorithm does not have sparsity and all samples are support vectors, implying that the participation of all training samples is required for every prediction. Second, the LSSVR model requires a complicated matrix inverse solution process, which requires $O(m^3)$ computational complexity, where $m$ is the number of training samples. The proposed algorithm utilized the pruning algorithm to alleviate the problem wherein the solution of LSSVR does not have sparsity. Moreover, it employed the matrix technique to transform the complex matrix inversion operation into an iterative solution, which greatly reduced the time complexity.

C. ERROR COMPARISON OF ONLINE MULTIPLE OUTPUTS LEAST-SQUARES SUPPORT VECTOR REGRESSION MODEL BASED ON SELECTION MECHANISM FOR PREDICTION OF SAMPLES WITH DIFFERENT TRAJECTORY CHARACTERISTICS

To further verify the prediction accuracy of the proposed model, six sample trajectories with different trajectory characteristics in different regions are selected in the sample trajectory set for model verification, as shown in Fig. 8. The sample characteristics of the six trajectories are shown in Table 5, in which the training and test set samples account for 60% and 40% of the total sample capacity, respectively. The prediction results and error graphs of the six trajectories are shown in Figs 9–20. Table 6 lists the prediction error for the six sample trajectories. From the error graphs and tables, the six predicted trajectories are noted to have a high degree of fit with the real trajectory, with a prediction error magnitude of $10^{-4}$ or even $10^{-5}$. The prediction error of trajectory ③ is $2.10807 \times 10^{-4}$ (23.42 m), which is the trajectory with the largest prediction error among the six trajectories. The graph observation shows that the test set trajectories are the most tortuous and complicated among the six trajectories;

**TABLE 5.** Trajectory characteristics of six sample trajectories.

| Serial number | MMSI   | Number of training set samples | Number of test set samples | Timestamp range (s) |
|---------------|--------|--------------------------------|----------------------------|---------------------|
| ①            | 41232400 | 131                            | 87                         | 1425233554–1425258058 |
| ②            | 41300900 | 93                             | 62                         | 1427678644–1427701483 |
| ③            | 150203465 | 104                            | 69                         | 1425618196–1425657444 |
| ④            | 477752300 | 72                             | 47                         | 1426333918–1426346917 |
| ⑤            | 353932000 | 99                             | 65                         | 1427191664–1427195265 |
| ⑥            | 371240000 | 71                             | 47                         | 1426741648–1426754388 |

![FIGURE 8. Six sample trajectories with different trajectory characteristics in different areas.](image1)

![FIGURE 9. Prediction result of trajectory ③.](image2)

![FIGURE 10. Prediction error of trajectory ③.](image3)

![FIGURE 11. Prediction result of trajectory ③.](image4)
however, the prediction accuracy of the proposed prediction model can still reach 23.42 m, which meets the requirements of trajectory prediction to avoid collision at sea.

D. ERROR COMPARISON BETWEEN THE ONLINE MULTIPLE OUTPUTS LEAST-SQUARES SUPPORT VECTOR REGRESSION MODEL BASED ON SELECTION MECHANISM AND OTHER COMMON PREDICTION MODELS

To further verify the prediction accuracy of the proposed model, it was compared with the RNN–LSTM model, traditional LSSVR model, and BP neural network. ANN is based on the research circling modern neurology, biology, psychology, and other disciplines. It reflects the basic process of the biological nervous system to address external things. ANN is a computing system that was developed based on simulating human brain nerve tissues. It is a network system comprising a large number of processing units extensively interconnected with each other. The advantages of ANN include large-scale parallelism, distributed processing, self-organization, and self-learning. Moreover, it is widely used in speech analysis, image recognition, digital watermarking, computer vision, and numerous other fields, delivering outstanding results. Among many forms of the neural network,
the multilayers forward BP network is the most widely used. The RNN–LSTM algorithm is a variant of the RNN. It is a new time-series processing model that can fully mine time-series-related information, and it is gradually being used in various fields of time-series prediction [31]–[33]. This method caters to several applications as it does not require many properties of time-series data and advanced calculation of the time-series parameters. The LSSVR model, a variant of the SVR model, is a novel small-sample learning method with a solid theoretical foundation. Unlike existing statistical methods, it does not involve probability measurement and the law of large numbers. In other words, it avoids the traditional process from induction to deduction and realizes efficient “transductive inference” from training the samples to forecasting the samples, which significantly simplifies the regression prediction problem. The main differences between the proposed method and the methods described in the section are listed:

(a) BP neural network is a local-search optimization method, where it is easy to fall into the local optimal value, resulting in the failure of network training. Moreover, BP neural network is sensitive to the setting of the initial weight. When the network is initialized with different weights, it converges to different local minima. The algorithm proposed in this study is a variant of the LSSVR model, which has strict theoretical and mathematical basis. Based on the principle of structural risk minimization, the proposed algorithm has better generalization ability than the former. Further, the proposed algorithm has global optimality.

(b) The network construction process of the RNN–LSTM model is complicated, and many hyperparameters must be set. Thus, improper data processing or unreasonable parameter settings hamper the realization of good results. Moreover, a large amount of data is required to maintain high accuracy; however, training such a data volume requires serious hardware resources and a long training time. Conversely, the proposed algorithm can maintain high prediction accuracy under the condition of limited samples. Furthermore, only three hyperparameters must be set in the proposed algorithm. Optimal parameters can be obtained using the parameter optimization algorithm to improve the prediction effect.

(c) The traditional LSSVR model is a multiple inputs and single output (MISO) structure, which can only output a single variable. When faced with complex multiple variables prediction problems, only multiple models can be established and separately trained, which is inefficient. Moreover, the traditional LSSVR model is an offline model. Once the new samples significantly differ from the original training set samples and the model cannot be updated, the prediction accuracy gradually worsens. The proposed algorithm alleviates the above problem. The internal structure of the MISO model is transformed into multiple inputs and multiple outputs. Further, the online learning algorithm is combined with the multiple outputs LSSVR model. The difference between the newly added samples and the original training set samples becomes the basis for establishing the SM of online learning to ensure the prediction accuracy under different data streams.

To ensure equity of the simulation comparison experiment, the same dataset and input and output characteristics were used.
used for the four prediction models. The parameter setting of the machine learning prediction model is very sensitive to the prediction accuracy. To eliminate the influence of improper parameter setting on the prediction effect and ensure equity of the comparison experiment of different prediction models, the input parameters of each algorithm are set according to the application background and experience. Further, the optimal hyperparameters are determined using the optimization algorithm. Three input parameters were provided to the RNN–LSTM algorithm: time step, batch size, and input size. According to this application scenario and subsection A in Section IV (i.e., using the ship behavior at the present time and past nine times as input data), the time step is set as 10. The batch size refers to the number of lines in a feed. Here, it is set to 35, which is the same as the window length of the proposed algorithm. Moreover, the input size is set to 61, i.e., the number of input parameters per line is 61. For hyperparameters, the parameter range is manually determined, after which the ACDE algorithm is used to automatically determine the optimal solution and train the model. The optimal combination of the parameters involved 2 hidden layers, with 56 and 26 neurons in the hidden layers, respectively, and 50 training cycles. For the BP neural network, the parameter range is manually set by analyzing the application scenarios, after which the optimization algorithm is used to determine the optimal solution. The following combinations of parameters are obtained: 61 input dimensions, 2 output dimensions, and 56 units in each hidden layer, 100 training cycles, and 0.001 learning rate. Meanwhile, for the traditional LSSVR model, the same parameters used in the proposed model are used. The prediction errors of the four models are shown in Fig. 21. All four models demonstrated good prediction accuracy at test samples of less than 5 and more than 15, where the magnitude of error was in the order of $10^{-5}$. At the fifth test sample, the ship sharply altered its course, resulting in reduced accuracy of the prediction models owing to the errors in different ranges. It is evident from Fig. 21 that the prediction accuracy of the proposed model and traditional LSSVR model in the first six sample points is the same. This is because at this time, the prediction error has not yet reached the starting conditions for the online learning algorithm to update the model. In this case, the proposed model only changes the internal structure of the multiple outputs and the prediction accuracy of both models is consistent. The online multiple outputs LSSVR model based on SM displayed the lowest error growth rate, which subsequently dropped to a stable point. The prediction errors of the other three algorithms roughly exhibited the same trend. In particular, the traditional LSSVR model reached its maximum value at the ninth sample point and then started to decline before finally converging at the 14th sample point. The maximum error of the model exceeded $8 \times 10^{-4}$. The prediction errors of RNN–LSTM and BP neural network models reached the maximum at the seventh sample point and then started to decline before finally converging at the 14th sample point. The maximum errors of both these models were nearly $10^{-4}$. Note that both the traditional LSSVR model and BP neural network model are offline prediction models. Thus, when the sixth sample point started turning sharply, the two models were not updated online in time, resulting in gradually increasing error. The training error will not be increased until the subsequent sample points are merged and retrained, indicating that these algorithms cannot meet the accuracy requirements of real-time prediction at sea. Regrettably, as a popular time-series prediction algorithm at this stage, the performance of RNN–LSTM is not ideal. Investigating the reasons, the algorithm is a variant of the RNN in deep learning. Its structure is extremely complex and requires a large number of training samples to achieve good model generalization. However, the number of samples in the application scenario of this model is limited because it must use real-time AIS data of the target ship sailing at sea. Therefore, under the application background of this article, the prediction performance of RNN–LSTM is not good.

The prediction accuracy of the four models was quantitatively analyzed by determining the mean value of the prediction error of all test samples, as shown in Table 7. As depicted from Table 7, the accuracy of the four models follows the order SM–OMLSSVR > LSSVR > RNN–LSTM > BP neural network. Note that only the SM–OMLSSVR maintained a prediction error at an order of magnitude of $10^{-5}$, whereas the traditional LSSVR model displayed a relatively low prediction error among the other three models. Compared with the traditional LSSVR model, the SM–OMLSSVR model combines the online learning algorithm based on selection mechanism to solve the problem of low accuracy in the case of data flow. Different from SM–OMLSSVR and LSSVR,

![FIGURE 21. Comparison of the prediction errors obtained using the four models.](image-url)
although RNN–LSTM and BP neural network models are also effective prediction methods, they are more suitable for large sample sizes. The running time follows the following order from low to high: SM–OMLSSVR > BP neural network > RNN–LSTM > LSSVR. The traditional LSSVR exhibits high time complexity because it does not have the sparsity of the solution and complex matrix inversion calculation. Conversely, the proposed SM–OMLSSVR model adopts the pruning algorithm and uses matrix operation skills to transform matrix inversion into an iterative calculation, which significantly reduces the training time of the model.

Thus, the online multiple outputs LSSVR model based on SM proposed in this study satisfies the requirements of online prediction and significantly improves the efficiency and accuracy of the offline LSSVR model.

VI. CONCLUSION

The following conclusions can be generalized based on the above simulation and comparison experiments:

1. The SM–OMLSSVR and traditional LSSVR algorithms are used to simulate and predict ship trajectories with different sample sizes (simulation experiments in Section A and B). Compared with the traditional LSSVR track prediction model, the proposed SM–OMLSSVR algorithm improves the prediction accuracy and operation efficiency of the model.

2. Using the proposed algorithm (simulation experiment in Section C), six trajectories with different trajectory characteristics in Tianjin port water are predicted. The prediction accuracy met the requirements of marine prediction.

3. Compared with other machine learning methods (simulation experiment in Section D), the proposed SM–OMLSSVR algorithm can maintain higher operation efficiency and prediction accuracy under the premise of a small sample size.

In summary, the SM–OMLSSVR is highly accurate in the online and real-time prediction of a target ship’s trajectory when sailing at sea. The prediction accuracy can be sustained at small samples. When new samples arrive with time, the proposed model provides online learning mechanism to update the model when its prediction error is less than the error threshold. Moreover, unlike the traditional LSSVR model that requires numerous matrix inversion calculations that are more complex and time-consuming, the proposed model employs an iterative, more convenient solution. In the future, the prediction model can be used in the collision avoidance decision generation system to predict the future trajectory of the target ship to assist and establish advanced collision avoidance decisions.

However, some aspects discussed in this study required indepth study and improvement. For instance, during online learning, the lack of sparsity of the traditional LSSVR model was alleviated using the pruning algorithm by deleting the sample data. The first sample data in the sample set was assumed to provide the least contribution to the ship trajectory prediction model; thus, these samples were regularly deleted to keep the sample set unchanged. However, some samples that have greatly contributed to the model in the earliest sample set were selected for deletion. Thus, to some extent, the accuracy of the model may have been damaged owing to the hasty subtraction. Therefore, selecting a more suitable pruning algorithm should further improve the prediction performance of the proposed model.

Furthermore, this system can be applied to collision avoidance decision-making system of MASS for real-time prediction of the trajectory of a target ship posing a collision risk with its own ship. The model can be used in the motion prediction module of MASS. When the ship encounters potential danger, the prediction trajectory of the target ship is used to determine the collision risk and detect the collision. At the same time, combining the module with the conflict resolution module, when the ship needs to determine the collision avoidance scheme, the predicted trajectory is used for real-time collision risk inspection. The generation of collision avoidance decision depends not only on the current navigation information of the target ship, but also on its navigation behavior in a certain time in the future. The future behavior can be predicted by trajectory prediction algorithm. The model proposed in this article will facilitate better understanding of the target ship’s movements and provide optimal collision avoidance decision.

REFERENCES

[1] Annual Overview of Marine Casualties and Incidents 2019. Accessed: Jul. 11, 2019. [Online]. Available: http://www.emsa.europa.eu/emsa homepage/2-news-a-press-centre/news/3734-annual-overview-of-marine-casualties-and-incidents-2019.html

[2] Y. Huang, L. Chen, P. Chen, R. R. Negenborn, and P. H. A. J. M. van Gelder, “Ship collision avoidance methods: State-of-the-art,” Saf. Sci., vol. 121, pp. 451–473, Jan. 2020, doi: 10.1016/j.ssci.2019.09.018.

[3] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, “Human motion trajectory prediction: A survey,” Int. J. Robot. Res., vol. 39, no. 8, pp. 895–935, Jul. 2020, doi: 10.1177/0278364920917446.

[4] X. P. Tong, X. Chen, L. Z. Sang, and Z. Mao, “Vessel trajectory prediction in curving channel of inland river,” in Proc. Int. Conf. Transp. Inf. Saf., Wuhan, China, 2015, pp. 706–714, doi: 10.1007/978-3-319-23215-6.

[5] Y. H. Cheng, Y. Y. Qiao, and J. Yang, “An improved Markov method for prediction of user mobility,” in Proc. Int. Conf. Netw. Service Manage., 2016, pp. 394–399, doi: 10.1109/CNSM.2016.7818454.

[6] M. Schreier, W. Willert, and J. Adamy, “An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 10, pp. 2751–2766, Oct. 2016, doi: 10.1109/TITS.2016.2522507.

[7] S. J. Qiao, K. Jin, N. Han, C. J. Tang, Gesangduoji, and L. A. Gutierrez, “Trajectory prediction algorithm based on Gaussian mixture model,” J. Softw., vol. 26, no. 5, pp. 1048–1063, Oct. 2015, doi: 10.13328/j.cnki.jos.004796.

[8] J. Liu, G. Shi, and K. Zha, “Vessel trajectory prediction model based on AIS sensor data and adaptive chaos differential evolution support vector regression (ACDE-SVR),” Appl. Sci., vol. 9, no. 15, p. 2983, Jul. 2019, doi: 10.3390/app9152983.

[9] L. Wang, L. Zhang, and Z. Yi, “Trajectory predictor by using recurrent neural networks in visual tracking,” IEEE Trans. Cybern., vol. 47, no. 10, pp. 3172–3183, Oct. 2017, doi: 10.1109/TCYB.2017.2705345.

[10] M. Gao, G. Shi, and S. Li, “Online prediction of ship behavior with automatic identification system sensor data using bidirectional long short-term memory recurrent neural network,” Sensors, vol. 18, no. 12, p. 4211, Nov. 2018, doi: 10.3390/s18124211.

[11] P. Payeur, H. Le-Huy, and C. M. Gosselin, “Trajectory prediction for moving objects using artificial neural networks,” IEEE Trans. Ind. Electron., vol. 42, no. 2, pp. 147–158, Apr. 1995, doi: 10.1109/41.370380.
H. Li, J. Liu, Z. Yang, R. W. Liu, K. Wu, and Y. Wan, “Adaptive constrained dynamic time warping for time series classification and clustering,” *Inf. Sci.*, vol. 534, pp. 97–116, Sep. 2020, doi: 10.1016/j.ins.2020.04.009.

M. Gao, G. Y. Shi, and W. F. Li, “Online compression algorithm of AIS trajectory data based on improved sliding window,” *J. Traff. Transp. Eng.*, vol. 18, no. 3, pp. 218–227, 2018, doi: CNKI:SUN:JYGJ.0.2018-03-025.

N. Eswar, S. Ashique, A. Panchbhai, S. Chakraborty, H. P. Sethuraman, K. Kuchi, A. Kumar, and S. S. Channappayya, “Streaming video QoE modeling and prediction: A long short-term memory approach,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 661–673, Mar. 2020, doi: 10.1109/TCSVT.2019.2895223.

T. Gao and Y. Chai, “Improving stock closing price prediction using recurrent neural network and technical indicators,” *Neural Comput.*, vol. 30, no. 10, pp. 2833–2854, Oct. 2018, doi: 10.1162/neco_a_01124.

H. Kim, M. Park, C. W. Kim, and D. Shin, “Source localization for hazardous material release in an outdoor chemical plant via a combination of LSTM-RNN and CFD simulation,” *Comput. Chem. Eng.*, vol. 125, pp. 476–489, Jun. 2019, doi: 10.1016/j.compchemeng.2019.03.012.

J. Liu et al.: Online Multiple Outputs LSSVR Model of Ship Trajectory Prediction

**JIAO LIU** received the B.Eng. degree in nautical science from Ningbo University (NBU), Ningbo, China, in 2017. She is currently pursuing the Ph.D. degree in maritime science and technology with Dalian Maritime University (DMU), Dalian, China. Her research interests include the fundamental study of traffic flow on the sea for channel design and ship collision avoidance.

**GUOYOU SHI** received the B.Eng., M.Eng., and Ph.D. degrees from Dalian Maritime University (DMU), Dalian, China, in 1993, 1996, and 2006, respectively. He was an Assistant Professor, a Lecturer, and an Associate Professor with the Navigation College, DMU, where he is currently working as a Professor. His research interests include ship maneuvering, intelligent ship collision avoidance, and technologies for maritime autonomous surface ship.

**KAIGE ZHU** received the B.Eng. degree in nautical science from Guangdong Ocean University (GDOU), Zhanjiang, China, in 2016. He is currently pursuing the Ph.D. degree in maritime science and technology with Dalian Maritime University (DMU), Dalian, China. His research interests include the fundamental study of traffic flow on the sea for channel design and ship collision avoidance.