

## Research Article

# Improved Vessel Trajectory Prediction Model Based on Stacked-BiGRUs

Yang Xu <sup>1,2</sup>, Jilin Zhang <sup>2,3</sup>, Yongjian Ren <sup>1,2</sup>, Yan Zeng <sup>1,2</sup>, Junfeng Yuan <sup>1,2</sup>,  
Zhen Liu <sup>1,2</sup>, Lei Wang <sup>1,2</sup> and Dongyang Ou <sup>1,2</sup>

<sup>1</sup>School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

<sup>2</sup>Key Laboratory of Complex Systems Modeling and Simulation, Hangzhou Dianzi University, Hangzhou 310018, China

<sup>3</sup>School of Cyber Security, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Jilin Zhang; [jilin.zhang@hdu.edu.cn](mailto:jilin.zhang@hdu.edu.cn)

Received 28 February 2022; Revised 3 April 2022; Accepted 6 April 2022; Published 13 May 2022

Academic Editor: Jinbo Xiong

Copyright © 2022 Yang Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An intelligent maritime navigation system is expected to play an important role in the realm of Internet of Vessels (IoV). As a key technology in navigation systems, vessel trajectory prediction technology is critical to the IoV. Automatic identification system (AIS), an automated tracking system, is used extensively for vessel trajectory prediction. However, certain characteristics in the AIS data, such as the large number of anchored trajectories in the area, anomalous sharp turns of some trajectories, and the behavioral differences of vessels in different segments, limit the prediction accuracy. In this study, we propose a novel vessel trajectory prediction model for accurate prediction with the following characteristics: (1) an anchor trajectory elimination algorithm to eliminate anchor trajectories; (2) a statistical trajectory restoration algorithm to repair sharp turning; (3) a two-stage clustering algorithm (D-KMEANS) to distinguish vessel behavior; and (4) a deep bidirectional gate recurrent unit (Stacked-BiGRUs) model to predict vessel trajectory and compare the accuracy of the model before and after improvement. The results show that the mean square error and the mean absolute error of the improved model are reduced by 27% and 46%, respectively. This research shows good potential for maritime navigation early warning and safety.

## 1. Introduction

As an extension of Internet technology, the Internet of things (IoT) takes advantage of communication sensing technology to realize the information exchange between things [1, 2]. Automatic driving technology benefits from the rapid development of the IoT [3], and its functions such as intelligent collision avoidance and collaborative control [4] are becoming increasingly mature. Internet of Vessels (IoV) provides vessel sensing and traffic information service in the whole drainage area. IoV exchanges large volumes of data among vessels and base stations, such as course, speed, and location. Therefore, IoV has the ability to provide intelligent navigation, a safer collision avoidance decision, and an efficient port area management by realizing the refinement of vessel trajectory prediction. However, in offshore ports with

high vessel density and complex traffic, it is very challenging to predict the trajectory of vessels. Unlike vehicle or pedestrian trajectory prediction, moving objects in a maritime environment are not restricted by geometric structure and their movement patterns are more complex than those of land vehicles. According to the “International Regulations for Preventing Collisions at Sea” (COLREGS), the navigation rules of vessels rely substantially on experience, which is difficult to quantitatively analyze. Furthermore, the historical Automatic Identification System (AIS) data contain the potential movement patterns of vessels, such as usual behavior in some areas or periodic entry and exit of a channel. However, the current methods rarely eliminate anchor trajectory, repair abnormal AIS data, and classify the behavior of vessels before predicting. The following features in the raw AIS data reduce the prediction accuracy:

- (1) There are irregular anchor trajectories in the raw data. The vessel in the anchored state will float with the wind and waves, producing irregular trajectories in a small range. The vessel at anchor can be regarded as a static obstacle, which will mislead the model training and reduce the prediction accuracy. Therefore, eliminating anchor trajectory is of great significance to improve the accuracy of trajectory prediction.
- (2) There are acute bends caused by abnormal points in the raw data. The cause of the abnormal point is that the vessel urgently avoids obstacles or the marine equipment sends wrong data. The purpose of model training is to learn the usual behaviors of the vessel, but the abnormal point will reduce the convergence speed of the model. Therefore, it is necessary to design an algorithm to repair abnormal points.
- (3) There are different behaviors in vessel navigation, such as setting sail, crossing the waterway, and working. Mixing these low similarity trajectories will reduce the accuracy of prediction. Therefore, classifying vessel behaviors plays an important role in improving prediction accuracy.

Recurrent neural networks (RNNs) can explore the inherent laws from AIS data and have superior generalization ability. In this study, we proposed an improved vessel trajectory prediction model based on Stacked-BiGRUs. The main contributions are as follows:

- (1) We proposed an anchor trajectory elimination algorithm to eliminate anchor trajectories. The anchor trajectory is identified by the speed characteristics of vessel berthing and setting sail.
- (2) We designed a statistical trajectory restoration algorithm to repair outliers. The outliers are repaired based on the probability distribution of the latitude and longitude changes in the trajectory.
- (3) We proposed a two-stage trajectory clustering method (D-KMEANS) to classify the vessel behaviors. The trajectories are classified by the DBSCAN and KMeans to extract behavior sets.
- (4) We built a Stacked-BiGRUs model. Compared with other recurrent neural networks, the bidirectional structure gained additional feature extraction, which effectively improved the prediction accuracy.

## 2. Related Work

*2.1. Trajectory Restoration.* Under ocean environment conditions, vessel trajectory data are prone to inaccuracies due to equipment abnormalities, wind, and waves. Therefore, data repair technology is required to eliminate these inaccuracies. The technology is mainly divided into constraint-based trajectory restoration and machine learning method-based trajectory restoration.

For the restoration method based on constraints, Song et al. [5] first proposed a data cleaning method based on

speed constraints, considering the limitation of the speed of data change. They achieved good results in a series of time series data restoration experiments. Tu et al. [6] used an improved RDP algorithm to address acute bends and self-intersections in trajectory data, which improved the accuracy of trajectory prediction. Li et al. [7] used an improved A\* shortest path algorithm to fully consider road network topology and historical matching points and proposed a new trajectory restoration algorithm. Gao et al. [8] used a dynamic programming method to set multiple intervals for sequence data, and searched for candidate repair points in an iterative manner, avoiding excessive repair of sequence data.

For the repair method based on machine learning, Kanarachos et al. [9] combined wavelet, neural network, and Hilbert transform to propose a new time series anomaly detection algorithm. Cheng et al. [10] proposed a trajectory restoration algorithm based on bidirectional LSTM, which had a good effect on trajectory restoration in curved waterways. Xue et al. [11] proposed a fractional gradient RBF neural network that drives momentum. The training error of this algorithm was lower than that of gradient descent, stochastic gradient descent, and momentum gradient descent.

*2.2. Trajectory Clustering.* As an important spatiotemporal object data type, vessel trajectories record the behavior characteristics of vessels. The trajectory behavior category can be divided using the clustering method. The method of vessel trajectory clustering, which is categorized based on distance, density, graph, and statistics, is summarized as follows:

For distance-based clustering methods, Mao et al. [12] proposed an incremental clustering algorithm, OCLUST, for the online processing of trajectory stream data, which achieved superior performance in clustering streaming trajectories. Xiong et al. [13] proposed a privacy and availability data clustering scheme (PADC) based on KMeans to enhance the selection of the initial center point.

For density-based clustering methods, Liu et al. [14] applied the extended density-based spatial clustering of applications with noise (DBSCAN) algorithm to correlate International Maritime Organization (IMO) rules with vessel trajectories for cluster analysis. Sun et al. [15] proposed a clustering method based on the minimum boundary matrix and the similarity of the buffer zone, and applied the DBSCAN algorithm twice to improve the accuracy of trajectory clustering. Han et al. [16] proposed an enhanced spatial clustering method based on density, which ensured the accuracy of the behavior recognition result by including additional geospatial information based on vessel speed and direction.

For graph clustering methods, Tian et al. [17] present a graph clustering privacy-preserving method that improves the security of private information. Budimirovic et al. [18] present a novel graph clustering method (IBC1/IBC2) to cluster human behaviors, and the method has reference significance in vessel behavior clustering.

For statistics-based clustering methods, Wen et al. [19] applied the improved algorithm PrefixSpan for sequential

pattern mining to vessel trajectory pattern mining, defined vessel principal vectors, cross sections, and boundaries, and identified vessel trajectories with similar motion patterns through pruning strategies. Peel et al. [20] described the activities of vessels as the four states of anchoring, sailing, entering/exiting ports, and trawling. Hidden Markov models were used to identify the laws of vessel activities and cluster the different states of vessels. Riveiro et al. [21] used kernel density estimation (KDE) to cluster vessel trajectories by selecting a suitable kernel function and window width and using observations to characterize the overall vessel motion pattern.

*2.3. Trajectory Prediction.* There have been several studies on vessel trajectory prediction methods. These methods are mainly based on dynamic model analysis, statistics, and machine learning.

*2.3.1. Trajectory Prediction Based on Dynamic Model Analysis.* The Kalman filter is a classic method in the field of linear system analysis. Several scholars have proposed various trajectory prediction methods based on the Kalman filter. Jaskolsk et al. [22] used the Discrete Kalman filter (KF) algorithm to improve the possibilities of vessel motion trajectory and monitoring in the TSS (Traffic Separation Scheme) and fairways area. Qiao et al. [23] proposed a dynamic trajectory prediction method based on Kalman filtering that used the estimated value at the previous moment and the observation value at the current moment to update the estimation of state variables, and subsequently predict the position of the vessel at the next moment.

*2.3.2. Trajectory Prediction Based on Statistical Models.* A Bayesian network is a probabilistic graph model that comprises a directed acyclic graph composed of nodes representing variables and directed edges connecting these nodes. By combining empirical knowledge and prior information, posterior information was obtained. Mazzarea et al. [24] proposed the Bayesian vessel position prediction algorithm KB-PF based on particle filters.

The Markov model is a statistical model that can predict the trend of data changes at equal time intervals in the future based on historical data. Tong et al. [25] used Markov chain- and gray prediction-related methods to propose a hidden Markov model based on the adaptive update parameters of environmental data captured by dynamic objects. It showed high accuracy in curve prediction. Qiao et al. [26] developed a trajectory prediction algorithm, PutMode, based on Continuous Time Bayesian Networks (CTBNs), and the experimental results showed that PutMode could predict the possible motion curves of objects in a more accurate and efficient manner.

The Gaussian process is a stochastic process mainly used to solve regression problems. Rong et al. [27] proposed a probabilistic trajectory prediction model, which decomposed vessel motion into horizontal and vertical predictions. In the horizontal direction, a Gaussian process was used to

model the uncertainty of horizontal motion, and the vertical direction was estimated through acceleration. Anderson et al. [28] regarded the trajectory as a one-dimensional Gaussian process. They calculated the posterior distribution of the predicted value by obtaining the joint prior density and covariance matrix of the observed value and the predicted value. Qiao et al. [29] proposed the Gaussian mixture model-based trajectory prediction method (GMTP), which used a Gaussian mixture model to model complex motion modes and calculated the probability distribution of different motion modes. Subsequently, the Gaussian process regression was used to predict the plausible motion trajectory of a moving object. Dalsnes et al. [30] proposed the Gaussian mixture model (GMM), which provided a measure of the uncertainty of the prediction results and addressed multiple modalities.

*2.3.3. Trajectory Prediction Based on Machine Learning.* Extreme learning machines (ELMs) are a single hidden layer feedforward neural network. Mao et al. [31] proposed an ELM-based trajectory prediction algorithm to predict the trajectory of vessels. The algorithm did not require the weights and biases of an iterative neural network; thus, its training speed was faster.

An autoencoder (AE) is an unsupervised neural network model that includes encoding and decoding. Inspired by the generative model, Murray et al. [32] proposed a bilinear autoencoder method to iteratively predict the future state and then generate the entire vessel trajectory. The model could estimate the distribution of future trajectories of vessels and quantify the uncertainty in predicting vessel positions.

A long short-term memory (LSTM) solves the long-term dependence of RNNs. Gao et al. [33] proposed a method that combines the advantages of LSTM and TPNNet. The proposed method was not only easy to implement and suitable for real-time analysis, but also presented a high prediction accuracy. Nguyen et al. [34] proposed a scalable sequence-to-sequence learning model combined with LSTM. Chen et al. [35] combined the advantages of LSTM, support vector machine (SVM), and extreme value optimization algorithms and avoided the weak generalization ability and robustness of a single deep learning method. Suo et al. [36] compared the accuracy and training efficiency of gated recurrent unit (GRU) and LSTM in vessel trajectory prediction. Xiao et al. [37] proposed a two-step LSTM, the unidirectional and bidirectional LSTM (UB-LSTM), combined with behavior recognition for vehicle trajectory prediction. Zhang et al. [38] proposed a multiscale convolutional neural network (MSCNN)-based high-frequency (HF) radar vessel trajectory prediction method to predict the trajectory hidden in the clutter. Jaseena et al. [39] combined the wavelet transform and the bidirectional LSTM and proposed the EWT-LSTM model to forecast wind. Xue et al. [40] proposed social-scene-LSTM for pedestrian trajectory prediction, which was a novel hierarchical LSTM-based network. It considered the social neighborhood and scene composition and employed three different LSTMs to capture people,

society, and scene scale information. The accuracy of pedestrian trajectory prediction was significantly improved.

### 3. Trajectory Prediction Model

This section introduces the trajectory prediction model. As shown in Figure 1, the model was divided into the four parts, namely anchor trajectory elimination, outlier repair, classification of vessel behavior, and trajectory prediction. We proposed an anchor trajectory elimination algorithm and a statistical trajectory restoration algorithm to improve trajectory quality. In the classification of vessel behavior, we designed a two-stage trajectory clustering algorithm (D-KMEANS) to extract the main navigation modes of vessels. Finally, in trajectory prediction, we trained the Stacked-BiGRUs model and use sliding window to predict vessel trajectory.

**3.1. Anchor Trajectory Elimination Algorithm.** Some anchored trajectories existed during the sailing cycle of vessels. As shown in Figure 2, these trajectories generally appeared as overlapping points at the same position or irregular clumps formed by reciprocating motion in a small area. We proposed an anchor trajectory elimination algorithm to eliminate the anchor trajectory.

Dividing the trajectories of different vessels according to the MMSI number, if the total number of vessels is  $M$ s, get the trajectory set  $\text{Traj} = \{\text{Traj}_1, \text{Traj}_2, \dots, \text{Traj}_M\}$ ;  $\text{Traj}_m$  is the set of trajectory points of the  $m$ th vessel, eliminating the anchor trajectory  $\text{Traj}_m$  for each vessel.

The very high-frequency (VHF) transceiver automatically broadcasted the vessel's kinematic information (vessel position, speed, heading, etc.) and static information (vessel name, vessel unique identifier, message serial number, vessel type, vessel size, current time, etc.) [41] in the form of AIS messages. We defined the trajectory of the vessel in article  $m$  as  $\text{Traj}_m = \{p_i(\text{MMSI}, t, \text{lon}, \text{lat}, \text{Sog}) | i = 1, 2, \dots, N\}$ , and  $p_i$  represented the locus point at time  $i$  on  $\text{Traj}_m$ , which included the marine mobile service identification (MMSI), timestamp ( $t$ ), longitude ( $\text{lon}$ ), latitude ( $\text{lat}$ ), and speed over the ground ( $\text{Sog}$ ).

Based on the above symbols, the anchor trajectory elimination process of  $\text{Traj}_m$  is shown in Algorithm 1. The specific process of the algorithm is as follows:

- (1) Every point of  $\text{Traj}_m$  was traversed. When the  $\text{Sog}$  at a certain point  $p_i$  was less than  $\text{Sog}_0$ , the next  $T_s$  points were continuously judged. When the  $\text{Sog}$  of all points was less than  $\text{Sog}_0$ , the point  $p_i$  was marked as an anchoring point and the sailing point was located. Otherwise, the detection of the anchoring point was continued until the end of the trajectory traversal.
- (2) If the anchor point of the vessel is detected, continue to detect whether there is a trajectory point  $P_k$  after the trajectory point  $P_{i+T_s}$ , of which the  $\text{Sog}$  is higher than the sailing speed threshold  $\text{Sog}_1$ . If yes, continue to check whether the consecutive  $\text{Sog}$  of  $T_s$  points following the trajectory point  $P_k$  is higher than the sailing speed threshold  $\text{Sog}_1$ . If yes, determine the

trajectory point  $P_k$  as the point where the anchor is weighed, and delete the trajectory between the anchor point and the point where the anchor is weighed; otherwise, repeat step (2) to detect the anchor point until the loop is over.

- (3) Return to step (1) and continue to detect until the end of the trajectory traversal.

**3.2. Statistical Trajectory Restoration Algorithm.** The trajectories that have undergone anchor trajectory elimination still include some abnormal points, resulting in abnormal movement patterns. To repair these outliers, a statistical trajectory restoration algorithm is used.

Each vessel trajectory is split into longitude and latitude sequences, which are marked as  $S_\beta = \{\beta_1, \beta_2, \dots, \beta_n\}$ . The longitude and latitude sequences then both receive anomaly repairs. In the  $S_\beta$ ,  $\beta = \text{lon}$  (longitude) or  $\text{lat}$  (latitude).  $S_{\text{lon}}$  and  $S_{\text{lat}}$ , respectively, represent the sequence of all longitudes and latitudes of a trajectory. The acceleration of a trajectory point  $\beta_i$  is  $a_i$ , which is calculated by the formula:

$$a_i = \frac{V\beta_{i,i+1} - V\beta_{i-1,i}}{t_{i+1} - t_{i-1}},$$

$$V\beta_{i,i+1} = \frac{\beta_{i+1} - \beta_i}{t_{i+1} - t_i}, \quad (1)$$

$$V\beta_{i-1,i} = \frac{\beta_i - \beta_{i-1}}{t_i - t_{i-1}},$$

$$i = 2, 3, \dots, n - 1.$$

The specific steps of probabilistic trajectory anomaly repair are as follows:

- (1) The acceleration sequence  $S_a = \{a_1, a_2, \dots, a_n\}$  of  $S_\beta$  is calculated from formula 1.  $S_a$  is used to establish the table of acceleration probability distribution  $P_a$ . The schematic diagram of  $P_a$  is shown in Figure 3. Consider the number of intervals is  $n - 2$  and the interval size is  $\delta = \max(S_a) - \min(S_a)/n - 2$ ; the probability value of each interval equals the ratio between the number of trajectory points whose accelerations fall within the interval and the total number of trajectory points.
- (2) Initialization  $W_i = 0$ ,  $i = 2, 3, \dots, n - 1$ . The sequence is windowed (size: 3, step: 1). The subsequence under each window is  $S_{\beta_{i-1,i,i+1}} = \{\beta_{i-1}, \beta_i, \beta_{i+1}\}$ ,  $i = 2, 3, \dots, n - 1$ .
- (3) Determine whether the current  $i$  is equal to  $n - 1$ . If yes, record the post-repair sequence as  $S'_\beta$ . If no, update  $i = i + 1$  and then proceed to step (4).
- (4) Repair trajectory points under the  $i$ -th window.
  - (a) Make  $W_{i+1} = W_i$ ;
  - (b) Build a repair value array for the trajectory points  $\{\beta_b, \beta_b - \varepsilon_{\max, \beta} + u, \dots, \beta_b + \varepsilon_{\max, \beta}\}$ , where  $b = i - 1, i, i + 1$ . The elements in the array are sorted in ascending order.  $\varepsilon_{\max, \beta}$  and  $u$  are the

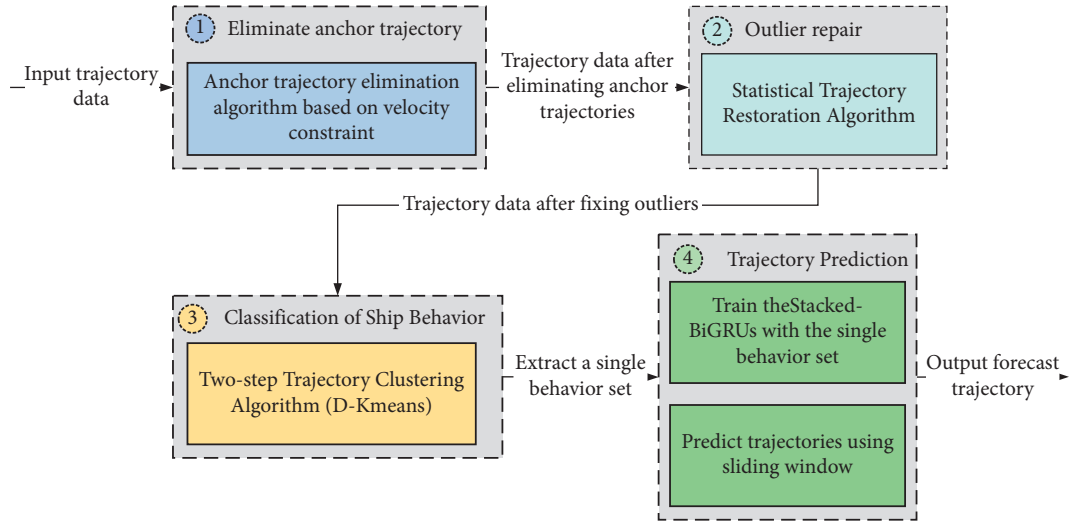


FIGURE 1: Improved vessel trajectory prediction model.

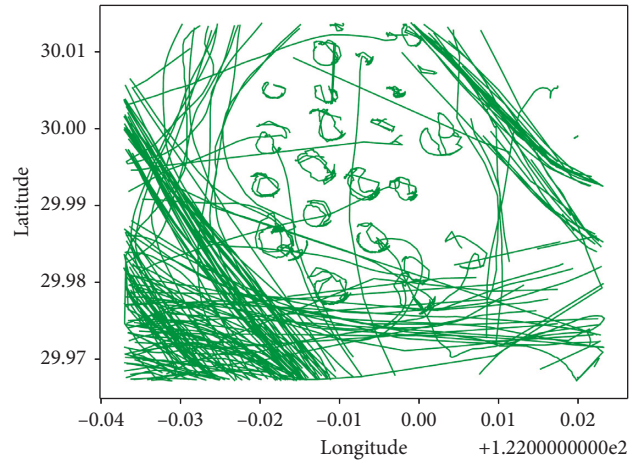


FIGURE 2: A marine chart of anchor trajectories.

- (i) **Input:** trajectory to be processed  $Traj_m$ , number of trajectory point  $n$ , anchoring speed threshold  $Sog_0$ , sailing speed threshold  $Sog_1$ , Detection length  $T_s$
- (ii) **Output:** processed trajectory  $Traj_m$
- (1)  $Traj_m = [P_1, P_2, \dots, P_n]$
  - (2) **For**  $P_i$  in  $Traj_m$  **do**//Start identifying anchoring point
  - (3) **If**  $P_i \cdot Sog < Sog_0$  and all  $P_j$  in  $[P_i.t, P_i.t + T_s]$  meet  $P_j \cdot Sog < Sog_0$
  - (4) Mark  $P_i$  as an anchor point
  - (5) **For**  $P_k$  in  $[P_i.t + T_s, P_n.t]$  **do**//Start identifying sailing point
  - (6) **If**  $P_k \cdot Sog > Sog_1$  and all  $P_l$  in  $[P_k.t + T_s + 1, P_k.t + 2T_s + 1]$  meet  $P_l \cdot Sog > Sog_1$
  - (7) Mark  $P_k$  as a sailing point
  - (8) Delete the trajectory between  $P_i$  and  $P_k$  //Eliminate anchor trajectory
  - (9) Break
  - (10) **End If**
  - (11) **End For**
  - (12) **End If**
  - (13) **End For**
  - (14) **Return**  $Traj_m$

ALGORITHM 1: Anchor Trajectory Elimination.

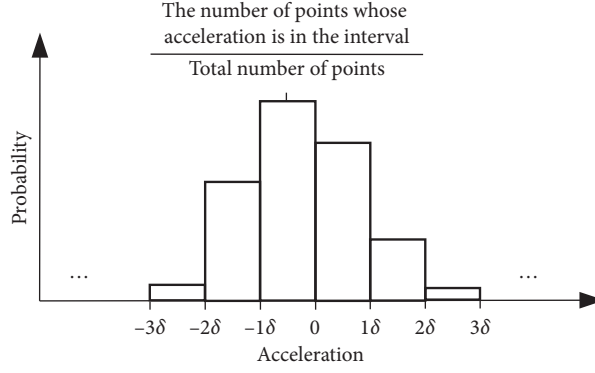


FIGURE 3: Schematic of the acceleration probability distribution of trajectory points.

- (i) **Input:** Sequence to be repaired  $S_\beta = \{\beta_1, \beta_2, \dots, \beta_n\}$ , maximum repair range  $\varepsilon_{\max, \beta}$ , step length of each repair  $u$
- (ii) **Output:** repaired sequence  $S'_\beta = \{\beta'_1, \beta'_2, \dots, \beta'_n\}$
- (1) Compute  $S_a = \{a_2, a_3, \dots, a_{n-1}\}$  according to equation (1)
  - (2) Create a Probability distribution table of acceleration  $P_a$  according to step 1
  - (3) **Initialize**  $W_i$ ,  $i = 2, 3, \dots, n - 1$
  - (4) **For**  $i = 2$  to  $n - 1$  do//Start repairing
    - (5)  $S_{\beta_{i-1,i,i+1}} = \{\beta_{i-1}, \beta_i, \beta_{i+1}\}$  //Windowing the subsequence (size: 3, step: 1)
    - (6) **For**  $\beta'_{i+1}$  in  $\{\beta_{i+1} - \varepsilon_{\max, \beta}, \beta_{i+1} - \varepsilon_{\max, \beta} + u, \dots, \beta_{i+1} + \varepsilon_{\max, \beta}\}$
    - (7) **For**  $\beta'_i$  in  $\{\beta_i - \varepsilon_{\max, \beta}, \beta_i - \varepsilon_{\max, \beta} + u, \dots, \beta_i + \varepsilon_{\max, \beta}\}$
    - (8) **For**  $\beta'_{i-1}$  in  $\{\beta_{i-1} - \varepsilon_{\max, \beta}, \beta_{i-1} - \varepsilon_{\max, \beta} + u, \dots, \beta_{i-1} + \varepsilon_{\max, \beta}\}$
    - (9) Compute  $a'_i$  according to equation (2)
    - (10) Read  $P(a'_i)$  from  $P_a$
    - (11) **If**  $W_{i+2} < W_{i+1} + P(a'_i)$  //If the probability goes up
    - (12)  $S_{\beta_{i-1,i,i+1}}' = \{\beta'_{i-1}, \beta'_i, \beta'_{i+1}\}$  //Update the subsequence
    - (13)  $W_{i+2} = W_{i+1} + P(a'_i)$  //Update the acceleration sequence
    - (14) Update  $S_a$ ,  $a_i = a'_i$  //Update the probability
    - (15) Update  $P_a$  //Update the table of acceleration probability distribution
    - (16) **End If**
    - (16) **End For**
  - (17) **End For**
  - (18) **End For**
  - (19) **End For**
  - (20) **Return**  $S'_\beta = \{\beta'_1, \beta'_2, \dots, \beta'_n\}$

ALGORITHM 2: Statistical Trajectory Restoration.

maximum repair range and step length of each repair, respectively. Traverse the candidate repair value array and then attempt to repair the trajectory point using the candidate repair values. Calculate the post-repair acceleration according to Formula (2) and then obtain the probability from the acceleration probability distribution table  $P_a$ . If  $W_{i+1} < W_i + P(a'_i)$ , replace with  $\{\beta'_{i-1}, \beta'_i, \beta'_{i+1}\}$  and update the probability value  $W_{i+1} = W_i + P(a'_i)$ .

- (c) Determine whether  $b$  is equal to  $i + 1$ . If yes, skip to step (3). If no, update  $b = b + 1$  and then return to step b).

$$a'_i = \frac{V\beta'_{i,i+1} - V\beta'_{i-1,i}}{t_{i+1} - t_{i-1}},$$

$$V\beta'_{i,i+1} = \frac{\beta'_{i+1} - \beta'_i}{t_{i+1} - t_i}, \quad (2)$$

$$V\beta'_{i-1,i} = \frac{\beta'_i - \beta'_{i-1}}{t_i - t_{i-1}},$$

$$i = 2, 3, \dots, n - 1.$$

The specific flow of the algorithm is shown in Algorithm 2:

**3.3. Ship Behavior Classification Algorithm.** Vessels have different or even conflicting navigation behaviors in the voyage cycle. For example, when the vessel starts sailing, the trajectory is in one direction, and the shape of the trajectory is short and dense. When the vessel goes to the target location at high speed, the trajectory is characterized by long distance, less turning, and smoothness. When the vessel reaches the target location, the trajectory is characterized by periodic repeated folding. Mixing different and conflicting trajectory features is not conducive to improving the accuracy of prediction. After obtaining the trajectory repaired in the previous section, this section mainly introduces the vessel behavior classification algorithm based on D-KMeans (DBSCAN-KMeans), which is used to distinguish different behavior patterns. The behavior sets are used for model training.

Ship locations are considered as spatial data; similar vessel behaviors can be given as clusters with enough spatial proximity. From the characteristics of vessel behaviors, we found that the DBSCAN meets the requirement of extracting the behavior trajectories. In the DBSCAN, it is necessary to specify two parameters,  $MinPts$  and  $\epsilon$ , which are the smallest number of vessels in a cluster and the sailing radius to a behavioral cluster. When vessels are sailing, a distance between vessels is typically calculated by the Mercator method. The distance unit of the Mercator method is sea mile. When the DBSCAN is applied to oceanographic data such as AIS data, the Mercator method is more accurate than the Euclid method to calculate the distance between two data points. Moreover, the time complexity of the Mercator method is similar to the Euclid method. Therefore, it is more reasonable to adopt the Mercator method in vessel trajectory clustering.

After clustering by DBSCAN, a large number of clusters  $C = (C_1, C_2, \dots, C_m)$  are generated. To merge these clusters into three vessel behaviors, KMeans is required. Because the points belonging to the same behavior have similar speed, KMeans is expected to cluster the average speed set  $V = \{\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(m)}\}$ ;  $\mu^{(i)}$  is the average speed of points in  $C_i$ . In KMeans, the data are divided into  $k$  clusters, setting the  $k$  value of KMeans to 3; by calculating the average speed of each cluster in the result of the previous step, and merging the first-step clusters with similar average speed, three types of vessel behaviors were obtained.

The algorithm is shown in Algorithm 3. The D-KMeans flow is described below:

- (1) The DBSCAN algorithm is used to cluster the vessel trajectory points that received outliers repair in section 3.3 to obtain the first-step clustering result.
- (2) The average speed set of each cluster  $V = \{\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(m)}\}$  is calculated from the first-step clustering result  $C = \{C_1, C_2, \dots, C_m\}$  in step (1).
- (3) With  $k = 3$ , the KMeans algorithm is used for the second-step clustering of average speed set  $V = \{\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(m)}\}$ . This is to obtain the three behaviors of the vessel, including setting sail, crossing waterway, and working.

**3.4. Stacked-BiGRUs Model.** After obtaining the vessel behavior set in the previous section, we used the behavior set to train the Stacked-BiGRUs model. As shown in Figure 4, the Stacked-BiGRUs model includes an input layer, three BiGRU units, and a dense layer.

The trajectory data are vectorized, and the trajectory points of several consecutive time steps are used as an input trajectory  $l = [p_1, p_2, \dots, p_n]$ .

To ensure dimensionless interference, the trajectory data were standardized before being used as the input trajectory of the model. The z-score standardization method was used to process the longitude and latitude in the trajectory data separately. As shown in Equation (3),  $l$  is the input trajectory,  $u$  is the mean of the series,  $\sigma$  is the standard deviation of the series, and  $l' = [p'_1, p'_2, \dots, p'_n]$  is the normalized input trajectory.

$$l' = \frac{l - u}{\sigma}. \quad (3)$$

In the trajectory prediction task, the bidirectional recurrent neural network processes the entire trajectory in the forward and reverse orders, and each output node comprises complete context information at the current time. The bidirectional GRU (BiGRU) structure is shown in Figure 5. The first GRU network processes the forward vessel trajectory, whereas the second GRU network processes the reverse vessel trajectory. The outputs of the forward and reverse networks are spliced into the final output  $h_t = (h_{t1}, h_{t2})$  after each time step. Compared with an ordinary GRU, the BiGRU has additional feature extraction.

In the forward calculation process, the trajectory  $[p'_1, p'_2, \dots, p'_n]$  is input into the forward GRU unit, and the hidden layer output of the forward unit is saved. In the backward calculation process, input the trajectory  $[p'_n, p'_{n-1}, \dots, p'_1]$  into the backward GRU unit, and save the output of the backward hidden layer. At each moment, concatenate the corresponding output results; the output of the BiGRU layer is  $[h_1, h_2, \dots, h_n]$ .

The dense layer maps the output to the target dimension, and the result  $pre$  of the Stacked-BiGRUs model is the next location of the vessel. The output  $pre$  should be mapped to the original dimension of the sample  $pre'$ .

$$pre' = \sigma pre + u. \quad (4)$$

Multi-step prediction of vessel trajectory can be realized using the sliding window method. Figure 6 is a schematic diagram of the sliding window method, in which the window size is 5 and the number of response steps is 1. For the trajectory on the left, the sliding window inputs the historical trajectory from  $t-4$  to  $t$ , and outputs the predicted point  $pre'$  at  $t+1$ . For the trajectory on the right, the historical trajectory point from  $t-3$  to  $t$  and  $pre'$  was taken as input, and the predicted trajectory point at  $t+2$  was the predicted point. In this way, the predicted trajectory of any time step can be output.

To evaluate the model, the mean square error (MSE) and the mean absolute error (MAE) were used to evaluate the effect of trajectory prediction. MSE is the squared expectation of the difference between the predicted value and the true value; MAE is the average of the absolute error.

```

(i) Input: Samples to be clustered  $L$ , sailing radius  $\epsilon$ , the smallest number of vessels in a cluster  $MinPts$ , number of behaviors  $k$ 
(ii) Output: clustering results  $Act$ 
(1) Mark all points in  $L = \{p_1, p_2, \dots, p_n\}$  as unvisited//Start DBSCAN clustering
(2) Calculate the matrix  $M$ , with each cell representing the Mercator distance between each two points
(3) Do
(4) Randomly select an unvisited point  $p_i$ 
(5) Mark  $p_i$  as visited
(6) Initialize  $C = \emptyset$ 
(7) If there are at least  $MinPts$  points in  $\epsilon$  field of  $p_i$ , then//The mercator distance between two points can be found in the M
(8) Initialize  $C_{temp} = \emptyset$ , add  $p$  to  $C_{temp}$ 
(9) Let  $N$  be the points set in the  $\epsilon$  field of  $p_i$ 
(10) For each  $p'_i$  in  $N$ 
(11) If  $p'_i$  is unvisited, then
(12) Mark  $p'_i$  as visited
(13) If there are at least  $MinPts$  points in the  $\epsilon$  field of  $p'_i$ , then
(14) Add points to  $N$ 
(15) End If
(16) If  $p'_i$  is not a member of any cluster, then
(17) Add  $p'_i$  to  $C_{temp}$ 
(18) End If
(19) End If
(20) End For
(21) Add  $C_{temp}$  to  $C$ 
(22) Else mark  $p_i$  as noise point
(23) End If
(24) Until all the points are marked,  $C = \{C_1, C_2, \dots, C_m\}$  //DBSCAN clustering is complete
(25) Compute the average speed set  $V = \{\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(m)}\}$  of each  $C_i$  in  $C = \{C_1, C_2, \dots, C_m\}$ 
(26) Select  $k$  points as the initial center point:  $\{\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(k)}\}$  //Start KMeans clustering
(27) Do
(28) Initialize  $Act_i = \emptyset (1 \leq i \leq k)$ 
(29) For  $x^{(j)}$  in  $V = \{\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(m)}\}$  do
(30) Compute the speed difference between  $x^{(j)}$  and  $\mu^{(i)}$ 
(31) The cluster label of  $x^{(j)}$  was determined according to the nearest cluster center
(32) Add  $x^{(j)}$  to the nearest cluster:  $Act_i = Act_i \cup \{x^{(j)}\}$ 
(33) For  $i = 1, 2, \dots, k$  do
(34) Compute the new cluster center:  $(\mu^{(i)})' = 1/|Act_i| \sum_{x \in Act_i} x$ 
(35) If  $(\mu^{(i)})' = \mu^{(i)}$ , then
(36) Update  $(\mu^{(i)})'$  as the cluster center
(37) End If
(38) End For
(39) End For
(40) Until the update of all clusters is complete//KMeans clustering is complete
(41) Return  $Act = \{Act_1, Act_2, \dots, Act_k\}$  //Behavior classification is complete

```

ALGORITHM 3: D-KMeans.

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2, \quad (5)$$

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|.$$

## 4. Results and Discussion

**4.1. Experimental Environment and Dataset.** The platform hardware configuration was a 2.9 GHz six-core Intel i5-9400CPU with 16 GB memory and Intel UHD Graphics 630. The following frameworks were used in the development

process: *Python 3.7*-based deep learning framework TensorFlow 2.0 and Keras, Scikit-learn for data processing, and GeoPandas and MovingPandas for trajectory analysis and visualization.

The dataset was selected from the data of vessels in the East China Sea, containing more than 100 GB of AIS point information collected from different types of vessels. The data were stored in the Analytical Massively Parallel Processing (MPP) database in real time, and the spatial connection and spatial index (PostGIS) were established simultaneously to realize the rapid extraction of trajectory data at a specific time and area. Between January 28 and February 1, 2021, 624,307 AIS data points from 522 vessels were selected as experimental data.



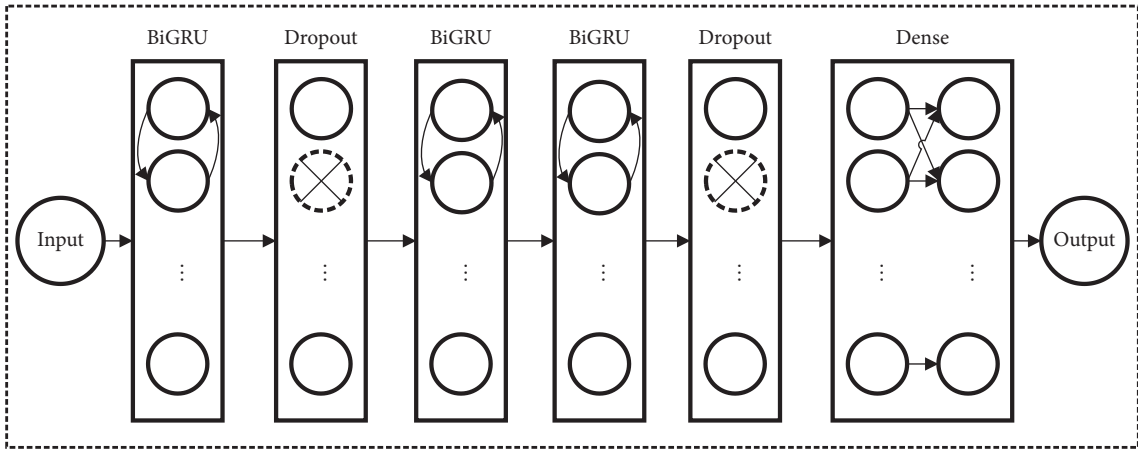


FIGURE 4: Structure of a Stacked-BiGRUs model.

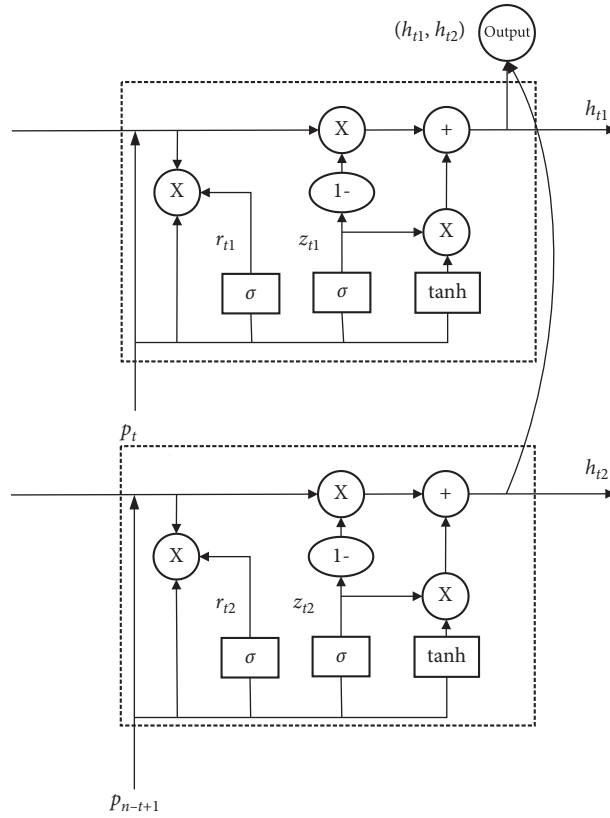


FIGURE 5: Structure of a bidirectional GRU.

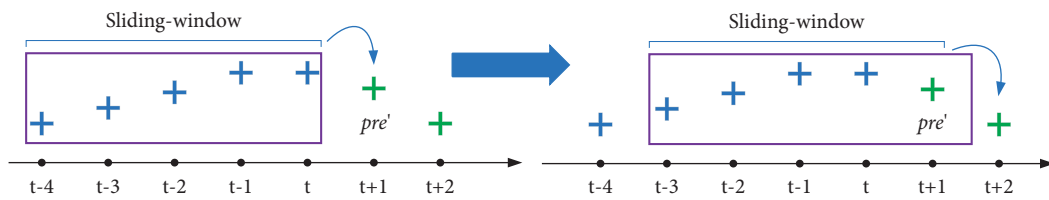


FIGURE 6: Sliding window method in trajectory prediction.

**4.2. Anchor Trajectory Elimination.** The anchor trajectory elimination algorithm is based on speed constraints; hence, it was necessary to perform statistical analysis on the speed of the Zhoushan offshore vessel. The primary research object of this study was a small vessel with a length of less than 60 m. The hull was characterized by small linear dimensions, low mass, small acceleration, and stopping inertia. Therefore, it was easily affected by external forces during movement. When the length of this type of vessel is twice the length of the berth, the speed of the vessel can be controlled below 0.3 knots.

From Figure 7, the speed of the vessels in the dataset is approximately two knots, and the remaining speeds are distributed between 0 and 0.5 knots and between 4 and 12 knots. The position of the vessel below 0.3 knots represents the anchoring state of the vessel. In the experiment, the anchor speed threshold  $V_T$  in the algorithm was set to 0.3 knots and the time step  $T_s$  was set to 5.

The experiment uses the dataset marked with anchor trajectories to test the algorithm performance. The dataset contains a total of 39,662 AIS trajectory points of 58 vessels, of which 16,411 trajectory points are vessel anchor points, and 17 vessels are completely berthed vessels. Table 1 shows the comparison of the number of stopped vessels and the number of anchored AIS points before and after processing the dataset by the algorithm. The results show that the total number of vessels processed by the algorithm has decreased by 17, all completely berthed vessels have been identified, and their anchoring trajectories have been eliminated; all points are reduced by 41%, and the total number of anchor points is reduced by 97.9%, indicating that the anchor trajectory elimination algorithm can effectively eliminate most of the anchor trajectories. The remaining anchor points that have not been cleared are mainly composed of abnormal points.

The visualization comparison of the chart before and after the algorithm processing is shown in Figure 8. The line segments of different colors represent the AIS trajectories of different vessels. The dense ring-shaped trajectories in the figure represent the trajectory data of the floating and anchored vessels. Affected by wind and ocean currents, it reciprocates in a small area. The picture on the right shows the processed chart trajectory. Compared to the left picture, the anchor trajectories are completely eliminated, which proves that the algorithm has a better processing effect.

**4.3. Trajectory Restoration.** After eliminating anchor trajectories, a section of the vessel trajectory that includes 1527 AIS points was selected to carry out experiments; the anchor trajectory was eliminated and the trajectory was split into a longitude and latitude sequence. The latitude sequence was selected as an example to show the repair result. Gaussian noise was added to the true sequence to obtain a dirty sequence, as shown in Figure 9.

The max repair range  $\varepsilon_{\max, lat}$  changed from one to four, as shown in Figure 10. As the repair cost increased, the repair effect increased accordingly. The repaired curve gradually fitted the real curve before Gaussian noise was added.

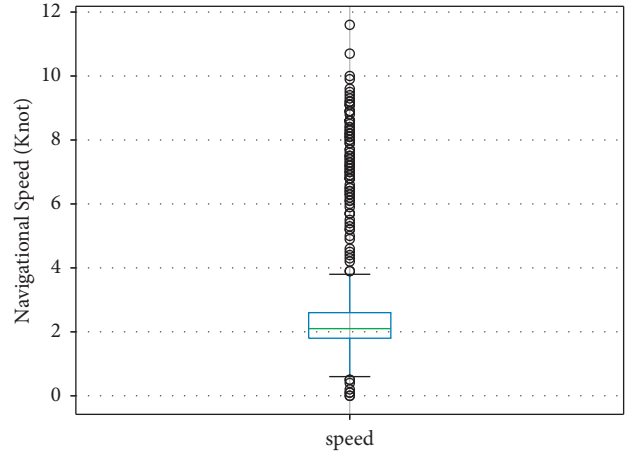


FIGURE 7: Box diagram of vessel speed.

Figure 11 is a graph of the RMSE of curve repair versus repair cost. Experimental results show that when the repair cost was four, the repair effect was strong, and the RMSE reached 0.0131, which is 58.9% lower than when the repair cost was one.

**4.4. Classification of Ship Behavior.** After repairing the outliers, the two-stage vessel trajectory flow clustering algorithm D-KMEANS was used to extract the trajectory of vessels crossing the waterway, as shown in Figure 12. We chose a vessel that has been processed in the previous section; the trajectory of the vessel 271217 contained 2459 AIS points. The vessel had experienced multiple departure and return cycles, and the behaviors of the vessel in different voyages had obvious temporal and spatial characteristics. We considered the historical trajectory data as the sample  $L$  and used the D-KMEANS algorithm for clustering.

- (1) We calculated the distance between each point of the trajectory and stored it as a Mercator distance matrix. The density of DBSCAN reached a radius of 3.6, the minimum sample value was 2, and the 2459 points in  $L$  were clustered. The spatial clustering results were clustered into 266 categories, as shown in Figure 12. Of these, 23 categories contained only one piece of data, which were outliers.
- (2) We excluded abnormal categories, leaving 243 categories to form a new sample  $L'$ . We then calculated the average speed of each type of AIS point and recorded the average speed set of all as  $V$ . The first step of clustering was complete.

Second, we performed a second-step clustering of the average speed set using KMEANS, as follows:

- (1) We clustered the average velocity set  $k$  into three categories. The average speed cluster between 0.5 and 1.5 knots was classified as low speed, the average speed cluster between 1.5 and 3 knots was classified as medium speed, and the average speed cluster between 5.5 and 11 knots was classified as high speed.

TABLE 1: Comparison of data before and after anchor trajectory eliminating.

State	Number of vessels	Number of points	Number of berthed vessels	Number of anchor points
Before	58	39662	17	16411
After	39	23578	0	327

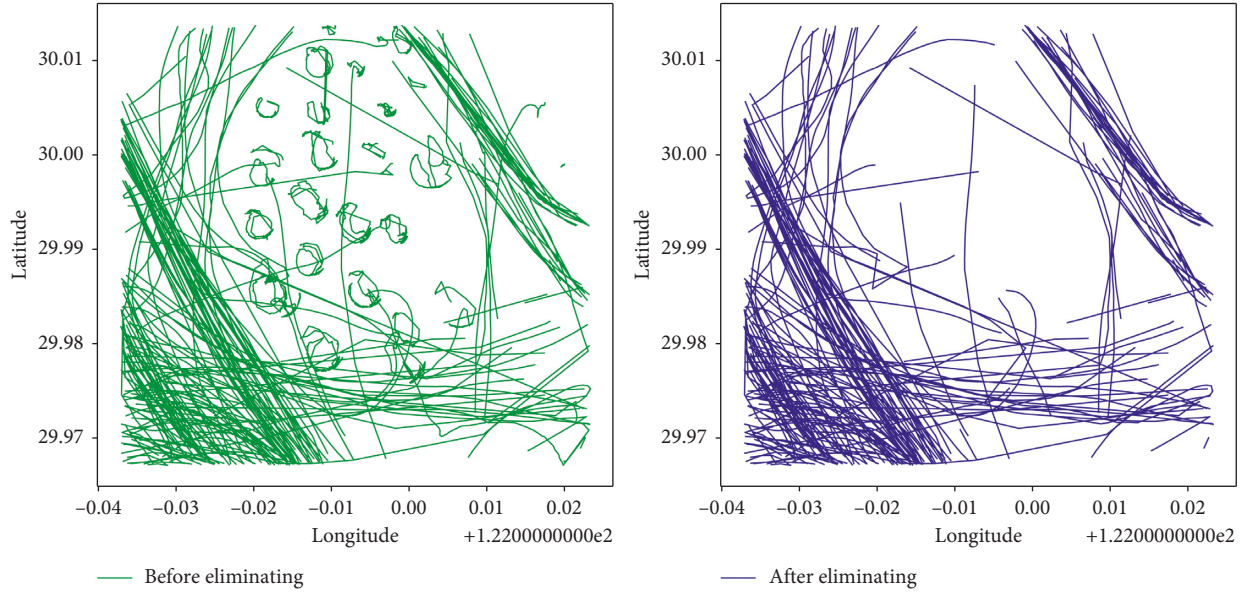


FIGURE 8: Anchor trajectories before and after eliminating.

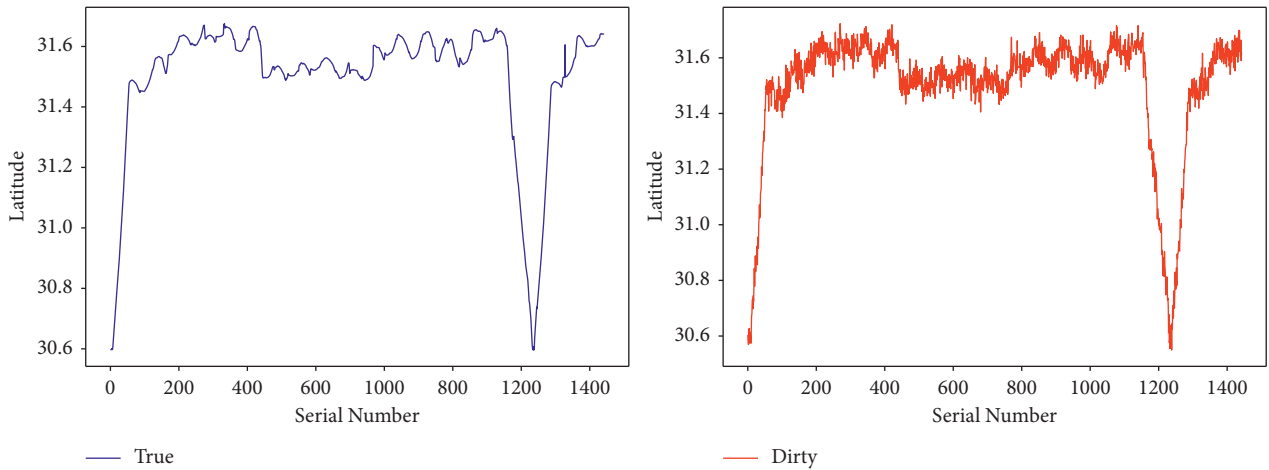


FIGURE 9: Latitude series with Gaussian noise.

Each average speed in  $V$  was labeled as low speed, medium speed, or high speed.

- (2) We used the label obtained in step (1) to divide the 266 classes in  $L'$  into three classes. Finally, the second stage of clustering was complete.

As shown in Figure 13, clusters formed by blue dots represent low-speed trajectories, red squares represent medium-speed trajectories, and green triangles represent high-speed trajectories. This distribution shows the following obvious behavioral characteristics: when the vessel was in the initial state, its speed was slow; when the

vessel entered the waterway to sail to the work area, its speed increased, and when the vessel reached its destination for operation, its speed was medium. Table 2 lists the relationship between the speed and vessel behavior. This study used the green high-speed trajectory of a vessel sailing in a waterway as an example to perform the next prediction.

**4.5. Trajectory Prediction.** The dataset was divided into the following three parts: training set, verification set, and test set, with a ratio of 6 : 2 : 2. The training set was used to train

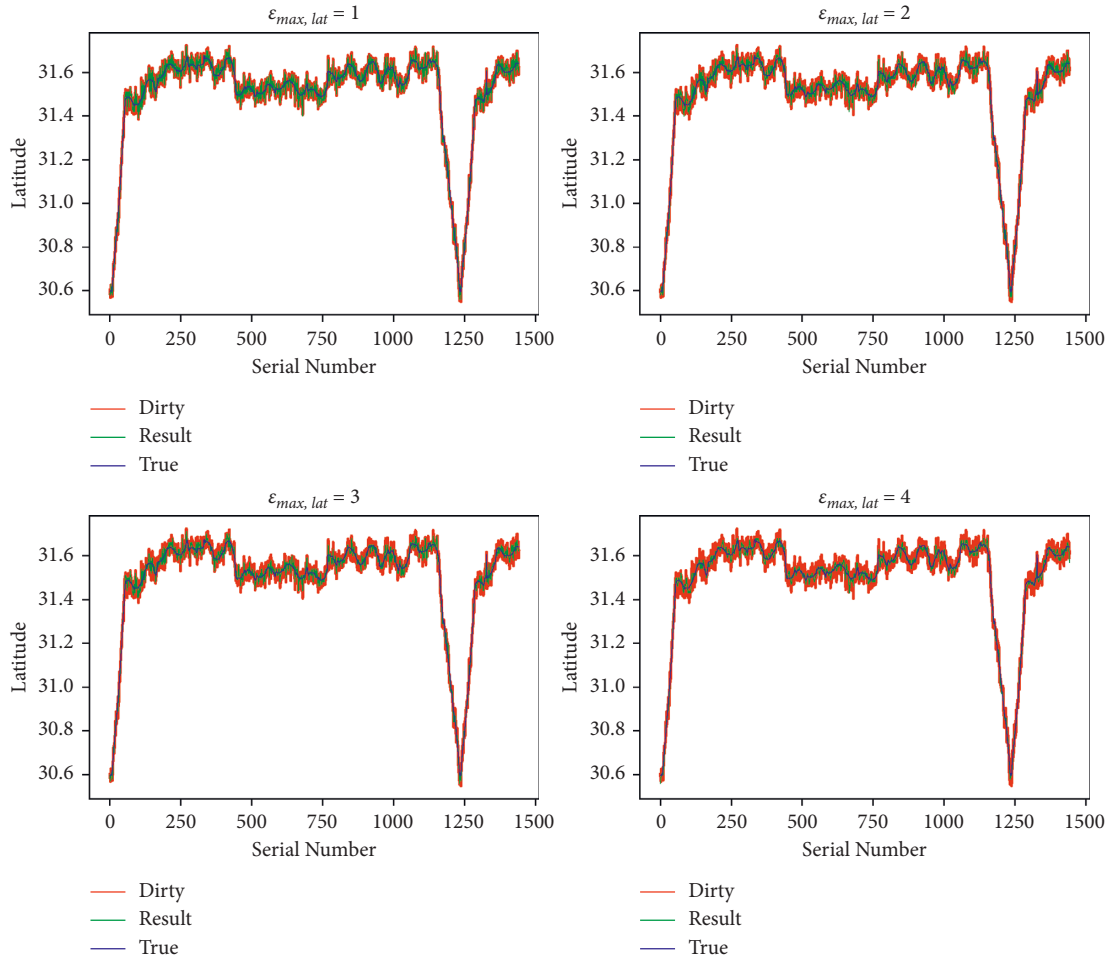
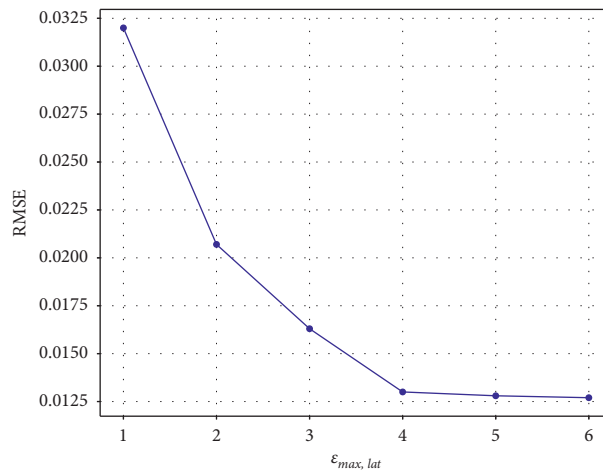
FIGURE 10: Latitude sequence repair effect changes with  $\epsilon_{max, lat}$ .

FIGURE 11: Variation of RMSE with the repair cost.

the model. In the training process, the verification set was employed to verify the performance of the model and improve its generalization ability. The test set was used to generate some prediction results. The Adam optimizer was

used as the activation function of the hidden and output layers. The selectable range of the batch size was  $\{16, 32, 64, 128, \text{ and } 256\}$ , and the experimental results of different batch size parameters are shown in Table 3.

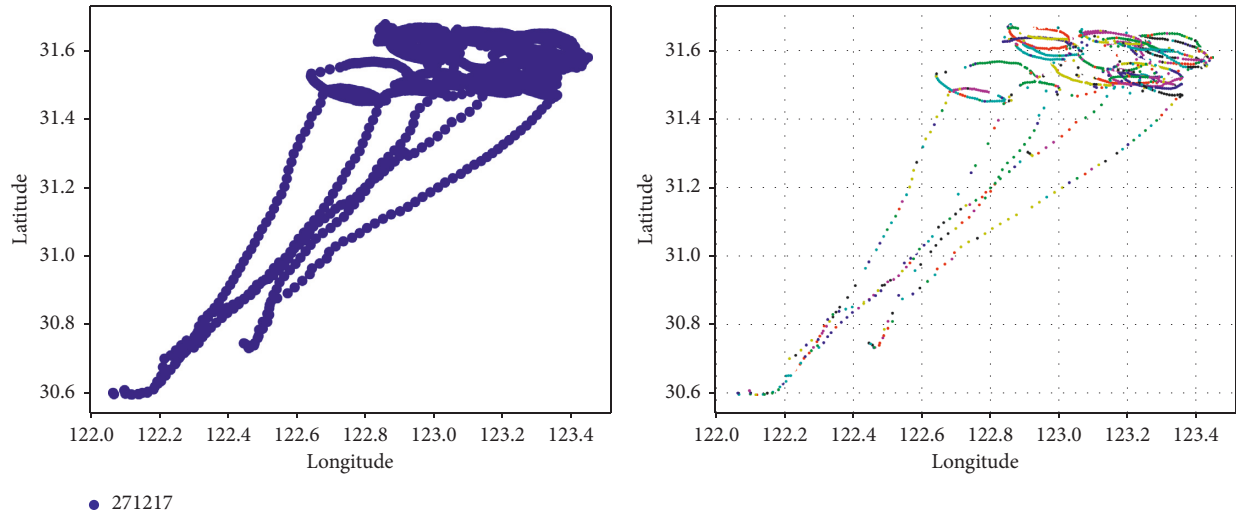


FIGURE 12: First-stage DBSCAN clustering result. First, we performed the first-step clustering of the trajectory using the DBSCAN algorithm, as follows:

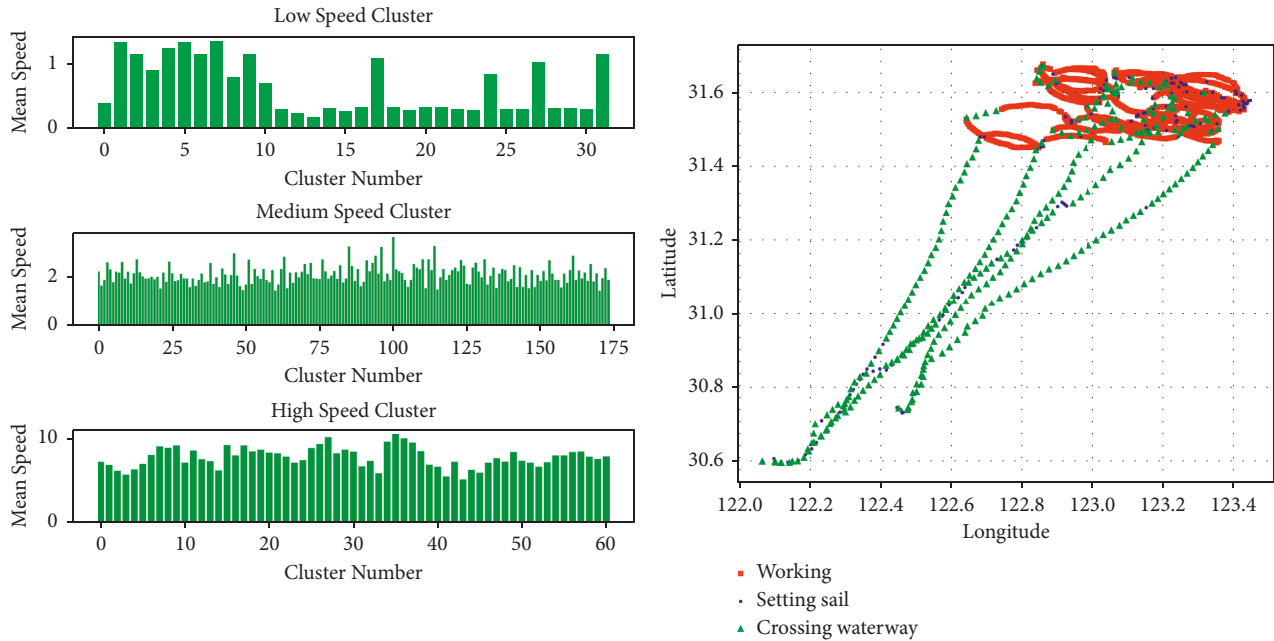


FIGURE 13: Second-stage KMeans clustering result.

TABLE 2: Comparison of data before and after anchor trajectory eliminating.

Speed status	Ground speed (knots)	Ship behavior
Low speed	$0.5 < Sog \leq 1.5$	Setting sail
Medium speed	$1.5 < Sog \leq 5.5$	Working
High speed	$5.5 < Sog \leq 12$	Crossing waterway

The training process is shown in Figure 14. The three models had fast iteration speeds in the first three rounds. When the number of training rounds was approximately 150, the models reached the extremum. The MSE of the LSTM model was 0.0037 and the MAE was 0.036; the MSE of the stacked-BiLSTM model was 0.0021 and the MAE was 0.0194; and the MSE of the stacked-BiGRU was 0.0018 and

the MAE was 0.0191. The deep bidirectional structure had additional feature extraction; therefore, the stacked-BiLSTM model and the stacked-BiGRU model presented lower errors.

The stacked-BiGRU and stacked-BiLSTM models had similar losses. When the number of training rounds was approximately 10, the MSE of the stacked-BiGRU model was

TABLE 3: Comparison of the experimental results of different batch size parameters.

Model	Metrics	Batch size				
		16	32	64	128	256
LSTM	MSE	0.00383	0.00371	0.00472	0.00611	0.0093
	MAE	0.0378	0.0363	0.0465	0.0505	0.0721
Stacked-BiLSTMs	MSE	0.00221	0.00209	0.00313	0.00422	0.00627
	MAE	0.0202	0.0194	0.0298	0.0441	0.0602
Stacked-BiGRUs	MSE	0.00184	0.00180	0.00247	0.00317	0.0544
	MAE	0.0216	0.0191	0.0207	0.0322	0.0511

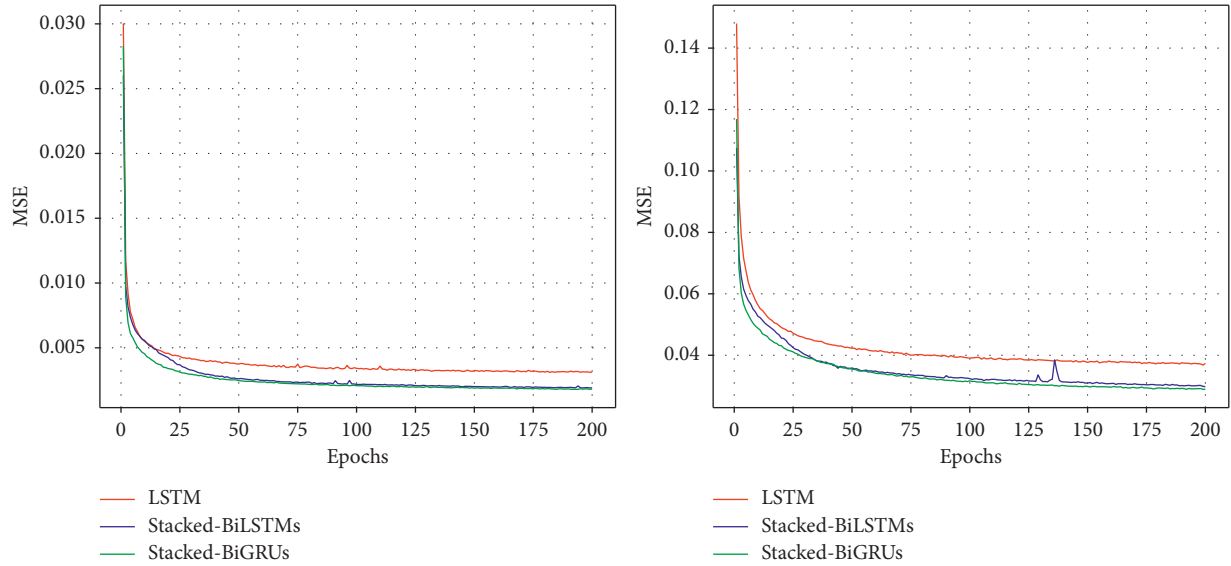


FIGURE 14: Comparison of three models.

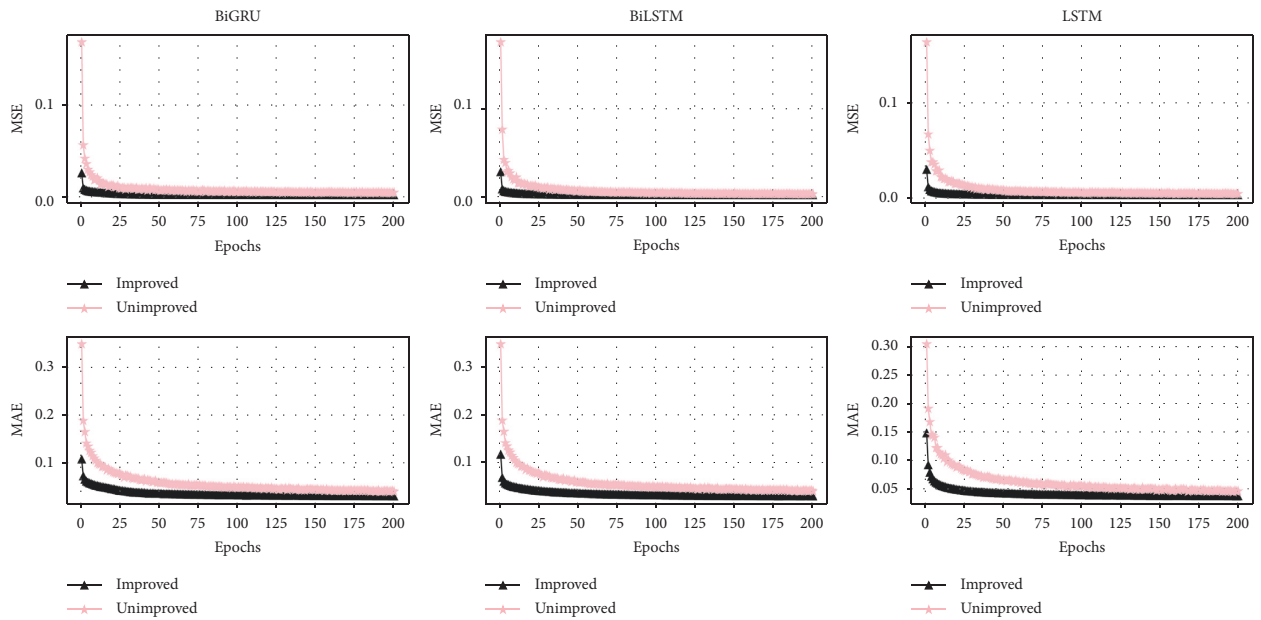


FIGURE 15: The impact of model improvement on prediction accuracy.

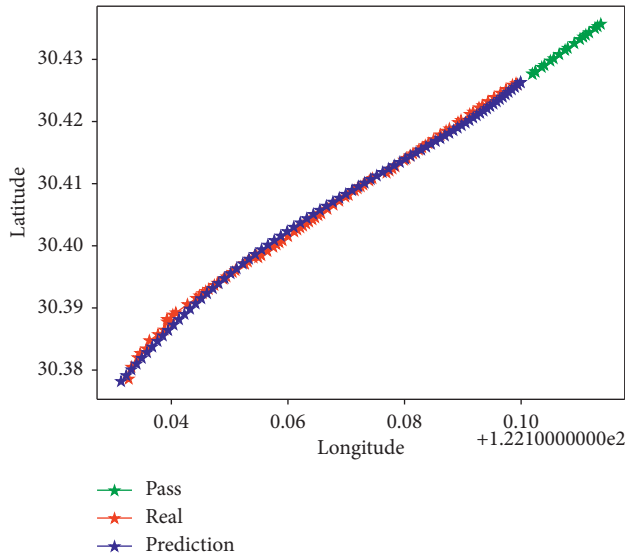


FIGURE 16: Comparison of the predicted trajectory and the real trajectory.

0.004, while the stacked-BiLSTM model reached this value at the 20th round. The stacked-BiGRU model converged faster mainly because the gate of the GRU unit was more simplified than the LSTM unit.

We also compared the impact of anchor trajectory elimination, outlier repair, and behavior classification improvement on different recurrent neural network models. As shown in Figure 15, the MSE of the improved model was 27% lower than that of the unimproved model on average, and the MAE was 46% lower than that of the unimproved model. The model converges after 55 epochs on average before improvement, and the improved model converges after 26 epochs. The results show that the improved model has quicker convergence rapidity and less error. This is because after improving, the abnormal data were eliminated, and the characteristics of the trajectory data were more concentrated, making it easier to analyze the inherent laws of the trajectory data.

The results of the simulation prediction are shown in Figure 16. From a path plan developed by the test, using a flexible window, the output of the previous model was used as the new trajectory data input, the planning model results at the corresponding time were repeatedly generated, and the predictions were 100 trajectories of flight trajectories. The online green line represents the historical trajectory, the blue line represents the predicted trajectory, and the red line represents the real trajectory. The predicted trajectory basically fitted the real trajectory, achieving a good trajectory prediction effect.

## 5. Conclusions

Trajectory prediction is a key requisite for navigation; in this research, to further improve the quality of maritime navigation in IoV, we considered the influence of anchor trajectory, trajectory abnormal points, and different vessel behavior characteristics on trajectory prediction, and

designed an improved vessel trajectory prediction model based on a recurrent neural network. For the anchor trajectory in the data, an anchor trajectory elimination algorithm was proposed to detect and eliminate abnormal data. A statistical trajectory restoration algorithm was proposed to repair the abnormal points in the trajectory. The vessel behavior classification algorithm D-KMEANS realized the extraction of different vessel behavior trajectories. Finally, a Stacked-BiGRUs model was built, and the sliding window was used to iteratively predict the position of the vessel at any step length.

The experimental results of the data processing part showed that the proposed algorithm achieved the expected results in terms of anchor trajectory elimination, trajectory repair, and vessel behavior classification. The comparative experiment of prediction models proved the performance of the Stacked-BiGRUs model in terms of prediction accuracy and convergence speed. This was mainly because the bidirectional model extracted additional features of the data, and the simplified gate structure of the GRU unit improved the training efficiency. The comparative experiments to verify the accuracy of the model showed the mean square error of the improved model is 0.0018 and the mean absolute error is 0.0191, which are reduced by 27% and 46%, depicting that the improved method can effectively improve prediction accuracy. The processing eliminated anchor trajectories and repaired abnormal data, and behavior classification resulted in a higher concentration of the characteristics of the vessel trajectory data, which made it convenient for the model to mine the inherent laws of trajectory data. Owing to this, the method proposed in this study may be well suited to proactively assist collision avoidance systems in ports and offshore areas.

## Data Availability

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

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This study was supported in part by the National Natural Science Foundation of China under grants J2024009 and 62072146 and in part by the Zhejiang Key Research and Development Program under grants 2019C05005 and 2021C03187.

## References

- [1] W. Zhang, A. M. Abdulghani, M. A. Imran, and Q. H. Abbasi, "Internet of things (IoT) enabled smart home safety barrier system," in *Proceedings of the 2020 International Conference on Computing, Networks and Internet of Things*, pp. 82–88, Sanya, China, April 2020.

- [2] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2019.
- [3] J. Xiong, R. Bi, Y. Tian, X. Liu, and D. Wu, "Towards lightweight, privacy-preserving cooperative object classification for connected autonomous vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 4, 2021.
- [4] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [5] S. Song, A. Zhang, J. Wang, and P. S. Yu, "SCREEN: Stream Data Cleaning under Speed constraints," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pp. 827–841, Victoria, Australia, May 2015.
- [6] E. Tu, G. Zhang, S. Mao, L. Rachmawati, and G.-B. Huang, "Modeling Historical AIS Data for Vessel Path Prediction: A Comprehensive treatment," 2020, <https://arxiv.org/abs/2001.01592>.
- [7] B. Li, Z. Cai, M. Kang et al., "A trajectory restoration algorithm for low-sampling-rate floating car data and complex urban road networks," *International Journal of Geographical Information Science*, vol. 35, pp. 1–24, 2020.
- [8] F. Gao, S. Song, S. Song, and J. Wang, "Time series data cleaning under multi-speed constraints," *International Journal of Software and Informatics*, vol. 11, no. 1, pp. 29–54, 2021.
- [9] S. Kanarachos, S.-R. G. Christopoulos, A. Chronos, and M. E. Fitzpatrick, "Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and Hilbert transform," *Expert Systems with Applications*, vol. 85, pp. 292–304, 2017.
- [10] Z. Cheng, Z. Jiang, X. Chu, and L. Liu, "Inland vessel trajectory restoration by recurrent neural network," *Journal of Navigation*, vol. 72, no. 6, pp. 1359–1377, 2019.
- [11] H. Xue, "Fractional-order gradient descent with momentum for RBF neural network-based AIS trajectory restoration," *Soft Computing*, vol. 25, no. 2, pp. 869–882, 2021.
- [12] J. Mao, Q. Song, C. Jin, Z. Zhang, and A. Zhou, "Online clustering of streaming trajectories," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 245–263, 2018.
- [13] J. Xiong, J. Ren, L. Chen et al., "Enhancing privacy and availability for data clustering in intelligent electrical service of IoT," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1530–1540, 2018.
- [14] B. Liu, E. N. de Souza, S. Matwin, and M. Sydow, "Knowledge-based clustering of vessel trajectories using density-based approach," in *Proceedings of the 2014 IEEE International Conference on Big Data (Big Data)*, pp. 603–608, IEEE, Washington, DC, USA, October 2014.
- [15] M. Sun and J. Wang, "An approach of ship trajectory clustering based on minimum bounding rectangle and buffer similarity," *IOP Conference Series: Earth and Environmental Science*, vol. 769, no. 3, Article ID 032017, 2021.
- [16] X. Han, C. Armenakis, and M. Jadidi, "DbSCAN optimization for improving marine trajectory clustering and anomaly detection," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, pp. 455–461, 2020.
- [17] Y. Tian, Z. Zhang, J. Xiong, L. Chen, J. Ma, and C. Peng, "Achieving graph clustering privacy preservation based on structure entropy in social IoT," *IEEE Internet of Things Journal*, vol. 9, p. 1, 2021.
- [18] N. Budimirovic and N. Bacanin, "Novel algorithms for graph clustering applied to human activities," *Mathematics*, vol. 9, no. 10, p. 1089, 2021.
- [19] Y. T. Wen, C. H. Lai, P. R. Lei, and W. C. Peng, "Routeminer: mining vessel routes from a massive maritime trajectories," vol. 1, pp. 353–356, in *Proceedings of the 2014 IEEE 15th international conference on mobile data management*, vol. 1, pp. 353–356, IEEE, Brisbane, Australia, July 2014.
- [20] D. Peel and N. M. Good, "A hidden Markov model approach for determining vessel activity from vessel monitoring system data," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 68, no. 7, pp. 1252–1264, 2011.
- [21] M. J. Riveiro, *Visual Analytics for Maritime Anomaly detection*, Örebro universitet, Örebro, 2011.
- [22] K. Jaskólski, "Automatic identification system (AIS) dynamic data estimation based on discrete kalman filter (KF) algorithm," *Scientific Journal of Polish Naval Academy*, vol. 211, no. 4, pp. 71–87, 2017.
- [23] S. Qiao, N. Han, X. Zhu, L. Chen, J. Ma, and C. Pen, "A dynamic trajectory prediction algorithm based on Kalman filter," *Acta Electronica Sinica*, vol. 46, no. 2, p. 418, 2018.
- [24] F. Mazzarella, V. F. Arguedas, and M. Vespe, "Knowledge-based vessel position prediction using historical AIS data," in *Proceedings of the 2015 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–6, IEEE, Bonn, Germany, October 2015.
- [25] T. Xiaopeng, C. Xu, S. Lingzhi, M. Zhe, and W. Qing, "Vessel trajectory prediction in curving channel of Inland river," in *Proceedings of the 2015 International Conference on Transportation Information and Safety (ICTIS)*, pp. 706–714, IEEE, Wuhan, China, June 2015.
- [26] S. Qiao, C. Tang, H. Jin et al., "PutMode: prediction of uncertain trajectories in moving objects databases," *Applied Intelligence*, vol. 33, no. 3, pp. 370–386, 2010.
- [27] H. Rong, A. P. Teixeira, and C. Guedes Soares, "Ship trajectory uncertainty prediction based on a Gaussian Process model," *Ocean Engineering*, vol. 182, pp. 499–511, 2019.
- [28] S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression," *Autonomous Robots*, vol. 39, no. 3, pp. 221–238, 2015.
- [29] S. J. Qiao, K. Jin, N. Han, C. J. Tang, and G. L. Gesangduoji, "Trajectory prediction algorithm based on Gaussian mixture model," *Journal of Software*, vol. 26, no. 5, pp. 1048–1063, 2015.
- [30] B. R. Dalsnes, S. Hexeberg, A. L. Flåten, B. -O. H. Eriksen, and E. F. Brekke, "The neighbor course distribution method with Gaussian mixture models for AIS-based vessel trajectory prediction," in *Proceedings of the 2018 21st International Conference on Information Fusion (FUSION)*, pp. 580–587, IEEE, Cambridge, UK, July 2018.
- [31] S. Mao, E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G.-B. Huang, "An automatic identification system (AIS) database for maritime trajectory prediction and data mining," in *Proceedings of the ELM-2016*, pp. 241–257, Springer, Cham, 2018.
- [32] B. Murray and L. P. Perera, "A dual linear autoencoder approach for vessel trajectory prediction using historical AIS data," *Ocean Engineering*, vol. 209, Article ID 107478, 2020.
- [33] D.-W. Gao, Y.-S. Zhu, J.-F. Zhang, Y.-K. He, K. Yan, and B.-R. Yan, "A novel MP-LSTM method for ship trajectory prediction based on AIS data," *Ocean Engineering*, vol. 228, Article ID 108956, 2021.



- [34] D. D. Nguyen, C. Le Van, and M. I. Ali, "Vessel trajectory prediction using sequence-to-sequence models over spatial grid," in *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems*, pp. 258–261, Hamilton, New Zealand, June 2018.
- [35] J. Chen, G.-Q. Zeng, W. Zhou, W. Du, and K.-D. Lu, "Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization," *Energy Conversion and Management*, vol. 165, pp. 681–695, 2018.
- [36] Y. Suo, W. Chen, C. Claramunt, and S. Yang, "A ship trajectory prediction framework based on a recurrent neural network," *Sensors*, vol. 20, no. 18, p. 5133, 2020.
- [37] H. Xiao, C. Wang, Z. Li et al., "UB-LSTM: a trajectory prediction method combined with vehicle behavior recognition," *Journal of Advanced Transportation*, vol. 2020, Article ID 8859689, 12 pages, 2020.
- [38] L. Zhang, J. Zhang, J. Niu, Q. M. J. Wu, and G. Li, "Track prediction for HF radar vessels submerged in strong clutter based on MSCNN fusion with GRU-AM and AR model," *Remote Sensing*, vol. 13, no. 11, p. 2164, 2021.
- [39] K. U. Jaseena and B. C. Kooor, "Decomposition-based hybrid wind speed forecasting model using deep bidirectional LSTM networks," *Energy Conversion and Management*, vol. 234, Article ID 113944, 2021.
- [40] H. Xue, D. Q. Huynh, and M. Reynolds, "Ss-lstm: A Hierarchical Lstm Model for Pedestrian Trajectory prediction," in *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1186–1194, IEEE, Lake Tahoe, NV, USA, March 2018.
- [41] M. Robards, G. Silber, J. Adams et al., "Conservation science and policy applications of the marine vessel Automatic Identification System (AIS)-a review," *Bulletin of Marine Science*, vol. 92, no. 1, pp. 75–103, 2016.