

## Research Article

# TPR-DTVN: A Routing Algorithm in Delay Tolerant Vessel Network Based on Long-Term Trajectory Prediction

Chao Liu <sup>1</sup>, Yingbin Li <sup>1</sup>, Ruobing Jiang <sup>1</sup>, Yong Du <sup>1</sup>, Qian Lu <sup>2</sup>,  
and Zhongwen Guo <sup>1</sup>

<sup>1</sup>Department of Computer Science and Technology, Ocean University of China, China

<sup>2</sup>Department of Computer Science and Technology, Qingdao University, China

Correspondence should be addressed to Ruobing Jiang; [jrb@ouc.edu.cn](mailto:jrb@ouc.edu.cn) and Yong Du; [csyongdu@ouc.edu.cn](mailto:csyongdu@ouc.edu.cn)

Received 19 October 2020; Revised 4 December 2020; Accepted 15 January 2021; Published 29 January 2021

Academic Editor: Xiao Zhang

Copyright © 2021 Chao Liu 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 efficient and low-cost communication system has great significance in maritime communication, but it faces enormous challenges because of high communication costs, incomplete communication infrastructure, and inefficient routing algorithms. Delay Tolerant Vessel Networks (DTVNs), which can create low-cost communication opportunities among vessels, have recently attracted considerable attention in the academic community. Most existing maritime ad hoc routing algorithms focus on predicting vessels' future contacts by mining coarse-grained social relations or spatial distribution, which has led to poor performance. In this paper, we analyze 3-year trajectory data of 5123 fishery vessels in the China East Sea. Using entropy theory, we observe that the trajectory of the vessel has strongly spatial-temporal distribution regularity, especially when previous states were given. To predict accurate future trajectories, we develop a long-term accurate trajectory prediction model by improving the Bidirectional Long-Short Term Memory (Bi-LSTM) model. Based on predicted trajectories and the confident degree of each prediction step, we propose a series of routing algorithms called TPR-DTVN to achieve efficient communication performance. Finally, we carry out simulation experiments with extensive real data. Compared with existing algorithms, the simulation results show that TPR-DTVN can achieve a higher delivery ratio with lower cost and transmission delay.

## 1. Introduction

As the critical technology for smart ocean, a low-cost and efficient maritime communication system plays a key role in daily communication [1, 2]. Most maritime activities, such as environmental monitoring, fishery operations, and knowledge exchange, can be successfully implemented under the above communication system [3, 4]. Existing maritime communication systems, including satellite-based, sea-based, and shore-based, have severe shortages in terms of high cost volume, limited network coverage, and absence of facilities [5], which cannot meet the requirements of low cost, wider data transmission in ocean regions [6].

With growing marine operations, the mobile vessel is becoming extremely important in the ocean due to its storage and communication capabilities. In the academic commu-

nity, considerable attention has been paid in the maritime delay tolerant vessel networks (DTVN) [7–11]. The routing algorithm of DTVN should make a correct decision on each relay based on known or expected information to ensure an effective data transmission. Thus, it is pivotal to get vessel mobility trends and accurately predict future contacts according to historical data [12, 13].

Recent works mainly focus on modeling mobile regularity by setting up the default movement model and mining social relationship [7, 8, 12, 14]. Most of the above work achieves coarse-grained, which is not enough to describe precise mobility patterns. The future trajectory is one of the most vital factors to build an efficient delay tolerant network system [13]. Once given the future trajectory, it is feasible to calculate expected contacts and get an optimal routing plan. Thus, the gap between actual and predicted trajectory has

significant effects on data delivery in DTVN, which means that the accurate trajectory prediction model plays a vital role in the routing process.

Vessels' movement is free space moving due to lack of map topology. Many factors such as fish distribution, weather conditions, and ocean current have significant effects on their movement [15–17], which makes traditional prediction models on land not infeasible for vessel trajectory prediction. For modeling vessels' mobile regularity, some researchers design models based on high-order Markov chain, Bayes model, and deep learning methods [15, 16, 18]. By splitting area to nets, some machine learning models can also be applied by transferring regression to classification problem [19]. The methods based on long-term high-order Markov and Bayes models can predict long-term trajectory. However, the model has low cross-regional extendibility based on our experimental study, which means that the model cannot work when facing a new ocean area. Existing deep learning-based methods all focus on short-term (less than 20mins) trajectory prediction or vessel state estimation. Besides, the regression results do not have proper evaluation metrics on each prediction step, making routing algorithms not adopt it.

To address the above problems, we design a trajectory prediction algorithm to model the vessel's mobile regularity and get the long-term accurate predicted trajectory. Besides, the routing algorithm takes the evaluation result of predicted trajectory as a condition. Even the future trajectories are available, the routing algorithm is essentially a global optimization problem, which is also an NP-hard problem. Therefore, we need to properly design the routing algorithm using predicted future trajectory and evaluation results, to improve the routing performance.

In this paper, we propose a set of low-cost, large-scale DTVN algorithms based on long-term predicted trajectories. Firstly, we utilize entropy to verify the predictability of vessels with massive actual traces data. It can be observed that the regularity in trajectory has lower entropy with more previous states and sailing features. Then, we design an improved Bi-LSTM trajectory predict model with great cross-regional extendibility to model the mobile regularity. To address the predicted trajectory evaluation problem, we utilize the prediction model's performance on the test set to design the confident degree of each predicted step calculation method. Finally, a series of low-cost, large-scale DTVN routing algorithms based on the predicted trajectory is designed.

We have made the following intellectual contributions in this paper:

- (1) It is the first work, to the best of our knowledge, to exploit the precise mobility pattern of vessel networks and use them for routing algorithms design in the vessel maritime communication system
- (2) We design an accurate long-term trajectory prediction model based on improved Bi-LSTM to model vessels' mobility patterns, which achieves more than 9-hour accurate predictions. The trajectory prediction model shows an excellent prediction perfor-

mance even the input sequences have never appeared in training sets. It indicates that our model can achieve a cross-regional trajectory prediction

- (3) An advanced evaluation method on trajectory regression results is designed to solve the regression model evaluation problem, making the predicted trajectory fully used by our DTVN routing algorithms
- (4) We propose a series of routing algorithms based on predicted trajectory to solve the optimal routing problem, which proves to be NP-Hard. Simulation in extensive actual trajectory data set, including 5123 vessels in 3 years, shows that our routing algorithm can achieve a higher delivery ratio than other routing strategies with lower cost and delay

The rest of the paper is organized as follows. The system model and problem formulation are given in Section 2. Section 3 introduces our basic idea and challenges. The structure of our trajectory prediction model is proposed in Section 4. The routing algorithm based on trajectory prediction is presented in Section 5. Section 6 gives the introduction of distributed algorithms. Section 7 shows the performance of our algorithm. Section 8 reviews related works. This paper is concluded in Section 9.

## 2. System Model and Problem Formulation

In this section, we introduce the system model of DTVN. Then, we describe the optimal target of system, expected delivery probability, and system evaluation metrics including system cost, delivery delay, delivery ratio, and system efficiency. Finally, we prove that the system optimization problem is NP-Hard.

*2.1. System Model.* All vessels in DTVN are considered to be a set of nodes represented by  $S$ . Node trajectory  $s$  in  $S$  is specified as a sequence of states over a period of time  $(t1, tn)$

$$T_s = \{c_s, t_1, c_s, t_2, \dots, c_s, t_n\}, s \in S. \quad (1)$$

In  $T_s$ , each state  $c_{s,t}$  has its information, including latitude, longitude, speed, and direction. For nodes  $s_i$  and  $s_j$  in  $S$ , we use  $Dis(s_i, s_j)$  to denote the distance between them and utilize  $r_{s_i}$ ,  $r_{s_j}$  to reflect their communication radius. Therefore,  $s_i$  and  $s_j$  can form a communication link when  $Dis(s_i, s_j)$  is smaller than  $\min(r_{s_i}, r_{s_j})$ . The whole set  $S$  can form a link set at the time  $t$ , which can be specified

$$L_t = \{(s_i, s_j) | s_i, s_j \in S\}, \quad (2)$$

$$Dis(p_{s_i,t}, p_{s_j,t}) < \min(r_{s_i}, r_{s_j}).$$

For a packet  $p$ , we use  $\delta(p)$  and  $\zeta(p)$  to denote its source and destination. The  $p$  could be copied for transfer to the node when they have links between them. Then, the packet

set  $P_n$  of size  $n$  is defined as

$$P_n = \{p_i(\delta(p_i), \zeta(p_i)) | \delta(p), \zeta(p) \in S, \\ i = 1, 2, \dots, n\}, \quad (3)$$

We also set TTL (time-to-live) for the packet's remaining living time,  $TTL_{\max}$  for the maximum living time,  $H$  for remaining hops, and  $H_{\max}$  for maximum hops. Packet delivery will fail if  $p_i$  does not get to its destination under the situation that TTL or  $H$  is equal to 0.

We next give the formal definitions of four system evaluation metrics system cost, delivery delay, delivery ratio, and system efficiency.

*Definition 1* (system cost). System cost represents the network occupation in the packet delivery process. In our system,  $\phi(p)$  is defined as the number of copies generated in the delivery process of  $p$ . Therefore, the system cost of  $P_n$  can be given by

$$\phi(P_n) = \sum_{i=1}^n \phi(p_i), p_i \in P_n. \quad (4)$$

*Definition 2* (delivery delay). To measure the delivery time of  $p$  from source  $\delta(p)$  to destination  $\zeta(p)$ ,  $\chi(p)$  is denoted as the delivery delay of  $p$ .  $t(\delta(p))$  is defined as the timestamp of  $p$  at the source node. So  $t(\zeta(p))$  is at the destination node. Delivery delay of  $p$  can be defined as  $\chi(p) = t(\zeta(p)) - t(\delta(p))$ . The system delivery delay is the average delay of  $P_n$ , which can be given by

$$x(P_n) = \frac{1}{n} \sum_{p_i \in P_n} x(p_i) \quad (5)$$

*Definition 3* (delivery ratio). Delivery ratio is defined as the proportion of the packets successfully delivered to total packets number  $n$ . The delivery ratio can be given as  $(|p_{\text{success}}|/n)$ .

*Definition 4* (system efficiency). A high-performance communication system should have a high delivery ratio with low cost and delay. Hence, we define system efficiency as the proportion of the delivery ratio to system cost multiply delivery delay. The efficiency of  $P_n$  is defined as

$$\text{Efficiency}(P_n) = \frac{|P_{\text{success}}|}{n\mathcal{X}(P_n) \times \phi(P_n)} \quad (6)$$

*2.2. Problem Formulation.* The target of the system is to deliver all packets from sources to destinations, respectively. Our system's objective is to maximize system expected delivery probability. Several definitions are given as follows.

*Definition 5* (packet expected delivery probability). Given a packet  $p$  with its source  $\delta(p)$  and destination  $\zeta(p)$ , the packet expected delivery probability  $Q_p$  is the expected delivery probability of  $p$ , which can be calculated by a function  $f(\cdot)$ .

$Q_p$  can be given by

$$Q_p = f(\delta(p), \zeta(p)), \delta(p), \zeta(p) \in S. \quad (7)$$

$Q_p$  is a fixed value if all the future trajectories  $T$  can be obtained. Additionally, it can be changed with nodes' different predicted future trajectories over time.

*Definition 6* (system expected delivery probability). Each packet  $p$  in  $P_n$  can get  $Q_p$  at the current time, the system expected delivery probability would be defined as  $\sum_{p \in P_n} Q_p$ .

The goal of our system is to maximize the system expected delivery probability. In conclusion, the whole optimal target of the system can be given by

$$\max \sum_{p \in P_n} Q_p. \quad (8)$$

*2.3. Complexity Analysis.* To address the expected delivery probability maximization problem, the algorithm needs to choose proper communication paths and a reasonable spectrum resource allocation strategy. The whole optimal problem is NP-hard due to its simplification problem. One can reduce the optimal resource allocation problem to the weighted maximum independent set problem (i.e., NP-hard problem).

Note that the links within interference distance cannot allocate the same spectrum resources, which might generate mutual-interference. A portion of links within the network  $L_t$  can allocate the same channel resources, constituting the reuse set. We use  $\{\omega_n\}$  to represent all possible reuse sets in  $L_t$ . Assuming that all communication paths are already known, and all reuse sets are allocated the same amount of subchannel resources.

For reuse set  $\omega_n$ , we select the links with a minimum data rate in one path and use  $\kappa_n$  to represent the sum of these links. Then, we let the vertex weight equal to the sum of the reuse sets' data rate. Edges between vertexes represent different reuse sets that have partially repeated links (i.e., no edge between two vertexes represents that two reuse sets are totally different). Even knowing  $\{\kappa_n\}$  and completed trajectory  $T_i \in S$ , this reduction can also be established. So, the weighted maximum independent set problem means to find reuse sets, which has a maximum sum data rate without repeated links.

For any instance  $I$  of the weighted maximum independent set problem, we can construct the instance  $I^*$  of the optimal resource allocation problem by using the partitioning  $\omega \in \{\omega_n\}$  to vertex set  $v \in \{v_n\}$  in  $I$  and using  $\kappa$  as weight for vertex  $v_n$ . So, if different  $\omega$  has partially repeated links in  $I^*$ , there are edges between them. The reduction can be accomplished in polynomial time, and it can verify that a feasible solution in  $I$  is a feasible solution.

### 3. Basic Idea and Challenges

In this section, we first verify the basic idea and algorithm feasibility. Also, challenges will be listed during the process of feasibility validation.

**3.1. Basic Idea.** In Section 2, we formulate the target of our algorithm. From problem formulation, we observe that the essential problem is how to predict future contacts among vessels. Obviously, if future trajectories can be reached or accurately predicted, we may measure the future contacts of the vessels in these trajectories. As a consequence, the DTVN routing problem can be converted into a trajectory forecast and predicted trajectory evaluation. The basic idea for our routing algorithm can be divided into four steps. The basic idea of our routing algorithm can be divided into four steps:

*Step 1:* long-term precise future trajectories should be predicted by applying a proper algorithm to massive historical data of vessels

*Step 2:* to make trajectories that can be used by the routing algorithm, predicted trajectories should be evaluated by specially designed measurement methods

*Step 3:* based on predicted trajectories and their evaluation results, our system calculates the future contacts' probabilities

*Step 4:* the routing algorithm utilizes contacts' probabilities to make a proper decision on each hop. A simple channel arrangement method should be applied during the data transmission process to solve the global optimal routing problem, which has been proved to be NP-Hard

**3.2. Feasibility Validation.** The effectiveness of the algorithm depends on whether the future trajectory of vessels can be predicted. To validate the predictability of the vessel's trajectory, we use conditional entropy, which has been commonly used to measure the disorder degree of time series data. Conditional entropy describes the disorder degree in the situation that some previous states have been given. The data is more predictable when it has lower conditional entropy than others.

From Section 2, we know that a vessel has  $T_s = \{c_{s,t1}, c_{s,t2}, \dots, c_{s,tn}\}$ ,  $s \in S$ . Firstly, we need to discretize the data in  $T_s$ . Meanwhile, we set the location into  $500 \text{ m} \times 500 \text{ m}$  grids among the ocean and use the center coordinate to indicate each sample's location. Speed and direction are, respectively, discretized by  $1 \text{ km}$  and  $20^\circ$ . For all the  $c_j$  of  $T_s$ , we count how many times  $c_j$  occurs and denote it as  $o_{c_j}$ , where  $1 \leq j \leq (\text{LOC} \times \text{SPE} \times \text{DIR})$ :

- (i) LOC represents the grid number
- (ii) SPE represents the discretized speed number
- (iii) DIR represents the discretized direction number

In terms of frequency ( $o_{c_j}/n$ ), we calculate the marginal entropy  $H(T_s)$  of  $T_s$  by equation (9)

$$H(T_s) = \sum_{j=1}^{\text{LOC} \times \text{SPE} \times \text{DIR}} \frac{o_{c_j}}{n} \times \log_2 \frac{1}{o_{c_j}/n}. \quad (9)$$

Extend  $T_s$  to sequence of two-tuples  $T_s^1 = \{(c_1, c_2), (c_2, c_3), \dots, (c_{n-1}, c_n)\}$ , count how many times  $(c_\alpha, c_\beta)$  occurs in  $T_s$  and denote it as  $o_{\alpha, \beta}$ , finally get joint  $H(T_s^1, T_s)$  entropy as equation (10)

$$H(T_s^1, T_s) = \sum_{\forall 1 \leq \alpha, \beta \leq (\text{LOC} \times \text{SPE} \times \text{DIR})} \frac{o_{\alpha, \beta}}{n-1} \times \log_2 \frac{1}{o_{\alpha, \beta}/(n-1)}. \quad (10)$$

Then, conditional entropy  $H(T_s^1 | T_s)$  of  $T_s^1$  can be calculated by equation (11)

$$H(T_s^1 | T_s) = H(T_s^1, T_s) - H(T_s). \quad (11)$$

Based on previous equations, we can keep calculating the conditional entropy of  $T_i^k$ , and eventually, the conditional entropy of  $T_i^k$  is derived as equation (12).

$$H\left(T_s^k | T_s^1 \dots T_s^{k-1}\right) = H\left(T_s^k | T_s^1 \dots T_s^{k-1}\right) - H\left(T_s^k | T_s^1 \dots T_s^{k-2}\right) - \dots - H(T_s). \quad (12)$$

We use entropy theory to analyze our data set. The data set from the Vessel Monitor System in China East Sea, which collects fishery vessel states from May 2015 to May 2018. We select 5123 vessels with good data integrity. System records vessels' sailing features including ship ID, latitude, longitude, speed, and direction on every 3 minutes. The whole data set has a range of  $120^\circ\text{E}$  to  $130^\circ\text{E}$  and  $25^\circ\text{N}$  to  $35^\circ\text{N}$ . Before analyzing, we sort all samples in time, remove duplicate and abnormal samples, to avoid any negative effects. Moreover, the local average interpolation method is applied to complete missing samples.

Figure 1(a) shows that the vessel's trajectory uncertainty will decrease when more previous states are given. It illustrates that more historical data will help the mining mobility pattern of vessels. We also analyze the mutual effect of conditional entropy between features. We use different combinations of features, including location, speed, and direction, to measure whether the trajectory is more predictable with different feature combinations. Figure 1(b) illustrates that these additional features are of great benefit to regularity mining. To sum up, it is highly possible that vessel's future trajectory can be predicted by specifically mining historical data.

**3.3. Challenges.** For time-series data prediction, some traditional models like Bayes or Markov chain have been applied [17, 18, 20]. Based on entropy results, a trajectory prediction model based on the high-order Markov chain has been realized. We use data from May 2015 to May 2016 to train the state transition matrix and the rest for testing. The test results of average continuous prediction time (ACPT) are shown in Table 1. It is obvious that 3-order Markov model with location, speed, and direction could only continuously predict the trajectory for 15.4 minutes on average. The continuous



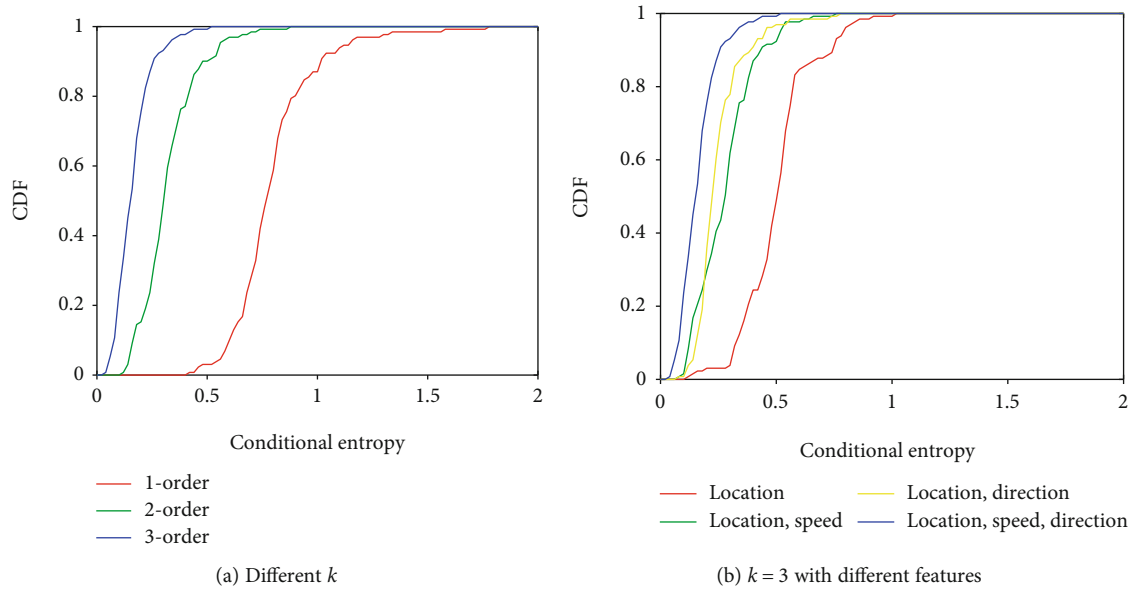
FIGURE 1: Conditional entropy CDF with different  $k$  and features.

TABLE 1: Average continuous prediction time (ACPT) of 3-order Markov model.

Features	ACPT (minutes)
Location	46.3
Location, speed	27.7
Location, direction	32.2
Location, speed, direction	15.4

prediction time can improve to 46.3 minutes on average, with 49.7% more distance bias when only applying the location feature. There are 32.2% data in the test set that cannot apply this model, because these models can only work when current states are in the training set. A trajectory cannot be predicted when vessels change their working area, type, or time. Moreover, the Markov model's continuous prediction time and availability will continue decreasing with higher-order, limiting the usage of historical data. Therefore, traditional models like Bayes or Markov chain have shortages on both long-term prediction and cross-regional extendibility. The first challenge is listed as follows:

**Challenge 1:** on account of traditional models' shortages, a novel trajectory prediction model needs to be designed to solve the historical data usage problems, long-term continuous prediction, and cross-regional extendibility. Section 4 addresses this challenge.

Since vessels' location, speed, and direction are all continuous variables, it is more suitable to use the regression model and sliding window to predict trajectory continuously. However, the regression results do not have a proper metric to evaluate each predicted step. The routing algorithm cannot make relay decisions without assessment criteria. So, challenge 2 and 3 are listed as follows:

**Challenge 2:** a metric that can approximately represent the probability that the vessel appeared at a predicted loca-

tion, which should be calculated using a novel evaluation method. The evaluation result needs to be directly adopted by the routing algorithm. This challenge is addressed in Section 5.1.

**Challenge 3:** a series of distributed routing algorithms with channel arrangements need to be proposed utilizing a novel evaluation method and predicted trajectories. The global optimal routing problem is NP-Hard, according to Section 2.3. This challenge is addressed in Section 5.2, Section 5.3, and Section 6.

## 4. Trajectory Prediction Model

In this section, we present our trajectory prediction model based on Bi-LSTM in detail. Firstly, the advantages of Bi-LSTM for time-series data are briefly introduced. Then, the detail of our model and its performance are given. Finally, we verify the cross-region extendibility of our model.

**4.1. Bi-LSTM for Time-Series Prediction.** Neural networks have a powerful ability to fit most functions. According to the universal approximation theorem, even a single hidden layer with no linear activation function neural network can approach any continuous finite dimension function with high precision.

For series data, many studies show that Recurrent Neural Network, like LSTM and GRU, has good learning performance to sequence learning problem [21, 22]. To solve the vanishing gradient and exploding gradient problems that might happen in RNN, the LSTM adds a cell, an input gate, a forget gate, and an output gate. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell. Some researchers propose that the Bidirectional LSTM shows a more robust learning ability in sequence data learning [20]. Different from traditional LSTM, one Bi-LSTM unit includes two LSTM units. They can control the forward and backward

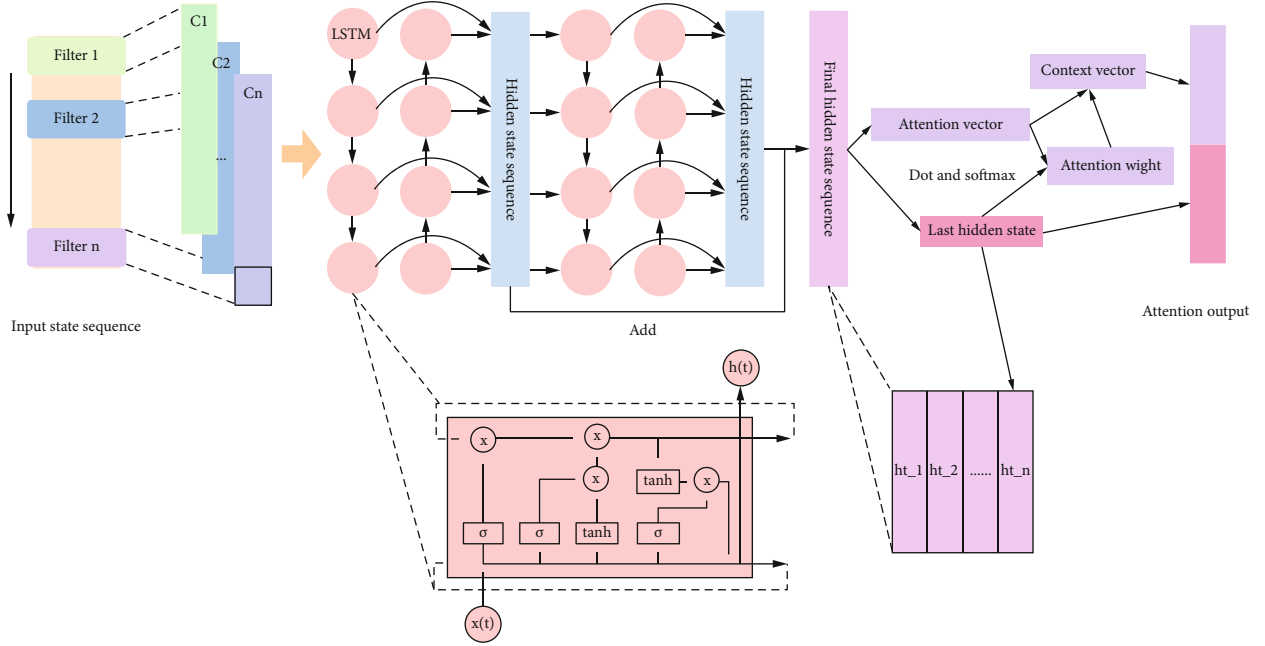


FIGURE 2: The structure of trajectory predict model and Bi-LSTM.

information flow in their cell state, respectively. In Figure 2, we could notice that the input state in the Bi-LSTM unit can flow in two directions. The unit in the next layer has connected with forward and backward units, which make it have a broader receptive field. Compared with LSTM, the Bidirectional LSTM is more suitable to predict the long-term trajectory in the ocean field.

**4.2. Model Design and Performance.** Our dataset's raw data is the vessel's state every 3 minutes in 3 years, including time, ship ID, latitude, longitude, speed, and direction. To train the neural network, we first need to get the training sample by transferring the time series to the supervised learning problem. We need to get the input state sequence and output sequence from the original series of states. Assuming the length proportion between the input and output sequence is  $m$ , and the length of the predicted sequence is  $\mathfrak{S}$ . For given state sequence  $\{c_1, c_2, \dots, c_n\}$ , the sample $_i$  in sample set is the following:

$$\text{Sample}_i = \{x_i : \langle c_i, \dots, c_i + m\mathfrak{S} - 1 \rangle, y_i : \langle c_i + m\mathfrak{S}, \dots, c_i + (m+1)\mathfrak{S} - 1 \rangle\}, \quad (13)$$

where  $1 \leq i \leq n - (m+1)\mathfrak{S} + 1$ . In this paper, we set  $m=3$  and  $\mathfrak{S}=1$ h in our model, which means that the model utilizes 3-hour historical information to predict the future 1-hour information. In addition to the location, our model also gives the prediction results of speed and direction. Therefore, if TTL is more than 1 hour, we can utilize the sliding window for long-term trajectory predictions. We shuffle the sample set and split the training, validation, and test sets according to the ratio of 7:2:1. Our model applies normalization to

avoid the negative effect of feature range in the learning process.

Figure 2 illustrates our model structure. Our model has four important parts, including convolution, Bi-LSTM, attention, and dense layers. We apply the stack of 1D convolutional layers at the beginning. The convolutional layer can process the grid structure data, including time series as 1D grids or graph as 2D grids. Because of the translation equivariant of convolution kernel, this layer helps our model learn spatiotemporal invariant patterns and reduce its capacity. The following layers also can get a bigger receptive field by the stack of the convolutional layers. As we know, one convolution kernel can only learn one pattern. Different convolution parameters like stride and activation functions have a significant impact on the model's performance. Therefore, we stack 3 layers of convolution with 64, 32, and 16 kernels with *ReLU* and set the stride = 3 in our improved model.

We use Bi-LSTM layers after convolution parts to avoid gradient explosion or vanish, which may lead to unstable state problems during the training process. Benefit from gate units and cell state, the typical LSTM has the memory of historical steps and stable gradient. Compared with typical LSTM, the Bi-LSTM has a more powerful fitting ability in sequence prediction. In each time step, the output of Bi-LSTM considers both forward and backward information. We choose four layers of Bi-LSTM with 64 cells and use *softsign* as an activation function. The *softsign* is better than *tanh* in this task, because it has a more gentle gradient performance than *tanh*. Additionally, we also choose proper weight initialization values based on the activation function. In this part, some residual connections between Bi-LSTM layers are added to avoid gradient vanishment.

The attention mechanism is added after the Bi-LSTM part. The attention mechanism can be used in time series prediction problems and show great performance in many

datasets. The attention, along with time steps, can make the model focus on some important sequences. In this scenario, the *latitude* and *longitude* are more important and have more variability than *speed* and *direction*. The model will treat all features equally if we do not use attention.

As shown in Figure 2, the attention layer gets the attention score by comparing the similarity of the last-hidden state with all history hidden states by the scaled-dot. Considering the *Last hidden state*, it has information learned from the whole input sequence, but the information from the beginning of the input sequence might disappear through long time steps. Therefore, one of the meanings that we design the attention layer is to integrate the information from the whole sequence. After that, *softmax* will be utilized to normalize the score. So, *the context vector* will be the weighted sum of attention score and hidden historical states. Then, we concat *the context vector* and *Last hidden state* as the output of attention layer.

$$\text{score}_i = \frac{\text{hidden}_{\text{last}} \cdot \text{hidden}_i}{\sqrt{\text{dim}_{\text{hidden}}}}, i = 1, \dots, m\mathfrak{S} - 1. \quad (14)$$

At the end of the model, we add dense layers after the attention layer to obtain the output. In parameter setting, we stack two dense layers with 8J and 4J cells with *ReLU* and *Linear* activation function, respectively. *Huber()*, which has excellent robustness to outlier point, is adopted as the network's loss function. We mainly set the callback, including early stopping and learning rate decay. The model will stop training if *MAE* do not reduce in 30 epochs. We use 0.001 as the learning rate at the first ten epochs and use exponential learning rate decay. Finally, we set a total of 500 epochs to train the model with previous callbacks.

Figure 3(a) indicates the average error on the test set. We can notice that our model can predict future trajectory in 1 hour with an average error of less than 300 meters. To further check the efficiency of our model, we use the sliding window to estimate the 9-hour trajectory. The model could give a longer predicted state sequence using slide window but might reduce the performance because using inaccuracy predicted states as input. And it also requires the model to predict all the features in state, which increases the complexity of trajectory prediction. The result shows that our model can achieve a 9-hour trajectory prediction with an average error of less than 2.73 kilometers.

**4.3. Cross-Region Extendability Validation.** As mentioned in Challenge 1, traditional models do not have cross-region extendability. To verify our model's extendability, we artificially divide the data set based on each vessel's activity range. We use samples within 60% distance from port to farthest point for model training and the rest for testing. Figure 3(b) shows that the test set's average error is 30% larger than that in Figure 3(a). Compared with long communication distance (average 30 km), this error can be tolerated. The possible reason for this extendability is that our model learns some patterns from other state information, like *speed* and *direction*. Also, the stack of convolutional layers makes our model learn the spatiotemporal invariant patterns. In conclusion,

cross-region extendability improves robustness during the applying process.

## 5. Global Routing Algorithm

In this section, we design the DTVN routing algorithm with global knowledge [?]. Like conventional delay-tolerant networks, vessels with larger expected encounter probability are more likely to meet in the future. When two vessels are within their range of contact, the packet should be copied and forwarded to the vessel with higher expected delivery probability in  $H$  hops. Therefore, the expected delivery probability calculation is the crucial issue of this algorithm. The process of expected delivery probability calculation is introduced in Sections 5.1 and 5.2. Section 5.3 will give a comprehensive overview of the routing algorithm.

### 5.1. Confident Degree of Predicted Trajectory

*Definition 7* (confident degree of predicted trajectory). Confident degree of predicted trajectory approximately represents the probability that the vessel is on its expected position.

Since the regression results of the improved Bi-LSTM model do not include such metrics, to evaluate the results of predicted trajectories, we propose the confident degree calculation method based on the model's performance on the test set.

The mean error  $\mu_n$  and standard deviation  $\sigma_n$  for each prediction step  $n$  on test set can be obtained after the model's training process.  $\mu_n$  and  $\sigma_n$  can show the model's performance on historical data, which helps to determine the expected effects. For all prior assumed distributions, if we have a mean and standard deviation, the normal distribution has the highest entropy, which means that it shall have the highest adaptability. So we choose a normal distribution to fit the error of each predicted step.

The error $_n$ , which is the errors of step  $n$ , is denoted as the random variable. We could assume that  $\text{error}_n \sim N(\mu_n, \sigma_n^2)$ . The Probability Density Function (PDF) with  $\mu_n$  and  $\sigma_n$  can be given by

$$f_n(\text{error}_n) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-(\text{error}_n - \mu_n)^2 / 2\sigma_n^2} dx. \quad (15)$$

Then,  $\mathcal{N}(\mu'_n, \sigma'^2_n)$  can be fitted according to the results of equation (14). Therefore, the probability of  $\text{error}_n$  less than  $(\mu'_n + 3\sigma'_n)$  is 99.87%. However, the predicted error can be tolerated in the routing scenario when the communication radius  $R$  is large enough. So the confident degree of the predicted trajectory should not only relate with  $\mu'_n$  and  $\sigma'_n$  but also relate with  $R$ . Considering that vessels are moving in 2D space, in this paper, the confident degree of predicted trajectory on step  $n$  can be given by

$$E_{v,n} = 99.87\% \frac{\left(R - (\mu'_n + 3\sigma'_n)\right)^2}{R_2}, \left(\mu'_n + 3\sigma'_n\right) \leq R, \quad (16)$$

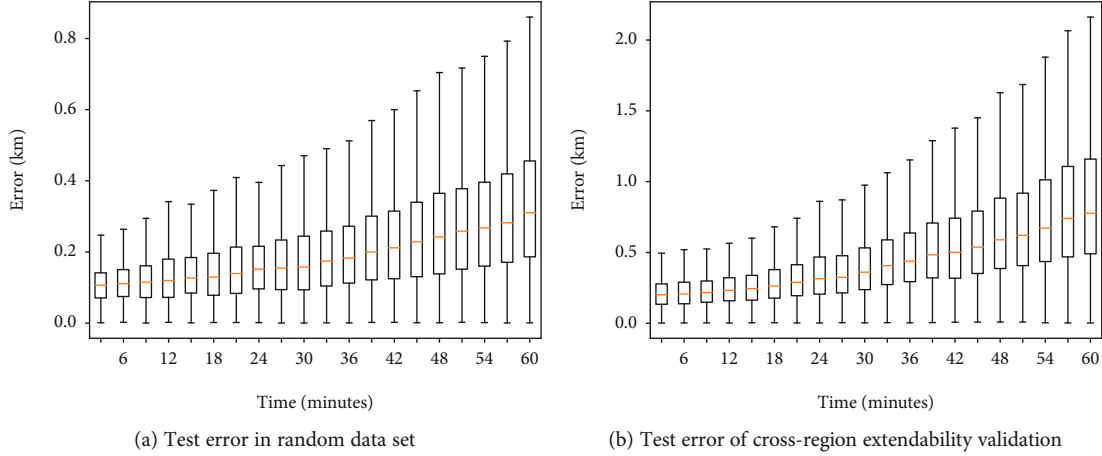


FIGURE 3: Test error in different partitions of data set.

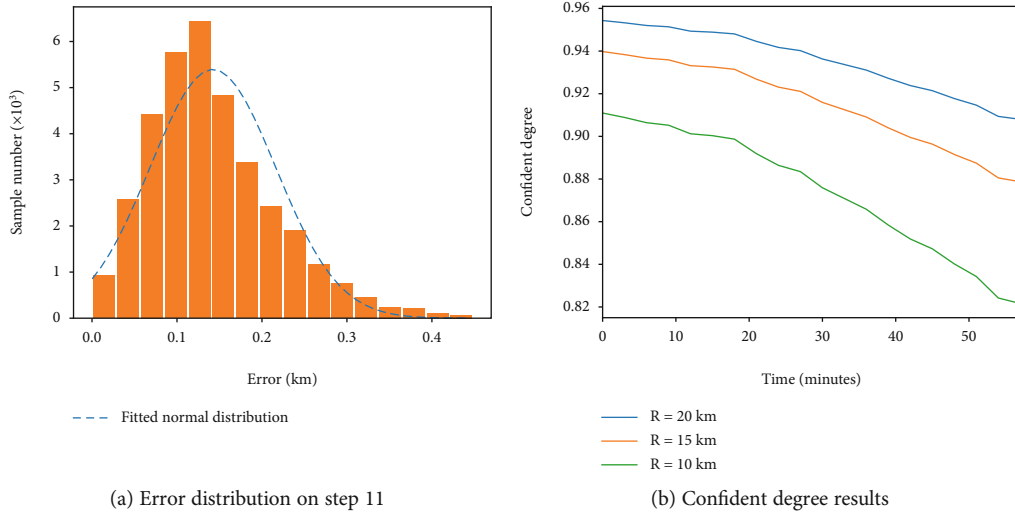


FIGURE 4: Confident degree of vessel “Zhe31828”.

where  $v$  is the identifier of vessel  $v$ . When  $(\mu'_n + 3\sigma'_n) \geq R$ , we define  $E_{v,n} = 0$ , because the error is too large to guarantee communication. The  $E_{v,n}$  could also be denoted as  $E_{v,t}$  where  $t$  is the timestamp of step  $n$ .  $E_{v,t}$  can be approximately treated as the probability of vessel appeared on predicted location based on empirical error and communication radius.

Figure 4(a) shows the distribution of the error in the 11th step of the vessel “Zhe31828” and its fitted distribution. We can see that the distribution of the error is fitted to the normal distribution. Confident degree results are shown in Figure 4(b). We can observe that the confident degree changes with different communication radius. The confident degree declines in long-term prediction. The vessel with a smaller communication radius has a lower confident degree due to its tolerance for error.

**5.2. Calculation of Packet Delivery Probability within  $n$  Hops.** For each vessel  $v$  at time  $t$ , the confident degree of predicted trajectory  $E_{v,t}$  can be obtained. We use  $v$  and  $\tau$  to denote two

nodes in the ocean.  $\mathcal{Q}^n v, \tau(t_{\text{start}}, t_{\text{end}})$  stands for expected delivery probability in time period  $[t_{\text{start}}, t_{\text{end}}]$  from  $v$  to  $\tau$  at just  $n$  hops, which means  $\mathcal{Q}^n v, \tau(t_{\text{start}}, t_{\text{end}})$  does not include the situation that packet is delivered in less than  $n$  hops.

When  $v$  and  $\tau$  are expected to meet (within their communication range) at time  $t$ , the one-hop delivery probability is  $E_{v,t} \times E_{\tau,t}$ . So, the one-hop delivery probability can be given by

$$\mathcal{Q}_{v,\tau}^1(t_{\text{start}}, t_{\text{end}}) = 1 - \prod_{t \in [t_{\text{start}}, t_{\text{end}}]} (1 - E_{v,t} \times E_{\tau,t}), \quad (17)$$

Where  $t$  is the start time of expected contacts. The accompanying time is not considered in this equation to avoid repeated calculation. The meaning of  $t$  is the same in the following derivation procedures in this subsection.

The condition of two-hop delivery can be treated as two one-hop delivery procedures with the time constraint. The



delivery probability of two-hops can be calculated as follows:

$$\mathcal{Q}_{v,r}^2(t_{\text{start}}, t_{\text{end}}) = 1 - \prod_{t_{\text{start}} \leq t \leq t_{\text{end}}} (1 - \mathcal{Q}_{v,m}^1(t_{\text{start}}, t) \times \mathcal{Q}_{m,r}^1(t, t_{\text{end}})). \quad (18)$$

Then, three-hop delivery probability can be divided into one one-hop delivery procedure and one two-hop delivery.

The three-hop delivery probability can be derived by

$$\mathcal{Q}_{v,r}^3(t_{\text{start}}, t_{\text{end}}) = 1 - \prod_{t_{\text{start}} \leq t \leq t_{\text{end}}} (1 - \mathcal{Q}_{v,m}^1(t_{\text{start}}, t) \times \mathcal{Q}_{m,r}^2(t, t_{\text{end}})). \quad (19)$$

Therefore, the delivery probability of  $n$  hops is given by

$$\mathcal{Q}_{v,r}^n(t_{\text{start}}, t_{\text{end}}) = 1 - \prod_{t_{\text{start}} \leq t \leq t_{\text{end}}} (1 - \mathcal{Q}_{v,m}^1(t_{\text{start}}, t) \times \mathcal{Q}_{m,r}^{n-1}(t, t_{\text{end}})). \quad (20)$$

Considering the routing in DTVN, a vessel has a packet  $p$  that needs to be delivered to  $\zeta(p)$  with its remaining hops  $H$  and time-to-live TTL. When the vessel meets its neighbor  $v$  at time  $t_c$ , neighbor's delivery probability within  $(H-1)$  hops needs to be calculated. The equation can be given by

$$\psi_{v,\zeta(p)}^{H-1}(t_c, t_c + \text{TTL}) = 1 - \prod_{i=1}^{H-1} (1 - \mathcal{Q}_{v,\zeta(p)}^i(t_c, t_c + \text{TTL})). \quad (21)$$

**5.3. Algorithm Description.** The routing algorithm based on long-term trajectory prediction is described as follows. Assuming that each vessel can obtain recent historical knowledge and trajectory prediction model of other vessels, and vessels sharing the same channel resource set need to compete for limited useful links.

For packet list  $P$  in vessel  $v$ , the source vessel first calculates  $\psi_{v,\zeta(p)}^{H-1}(t_c, t_c + \text{TTL})$  of each  $p$  in  $P$  on  $v$ 's neighbors.

The algorithm selects the relay node with the largest delivery probability. If links in the area are over link limitation, the algorithm will transfer the packet by probability rank to avoid channel collision.

When the packet  $p$  is copied and transferred, the system will update the copied packet's information. The algorithm would renew  $H_{\text{copy}}$ . The packet will be abandoned if its TTL becomes 0. The packet  $p$  will stop forwarding when  $H$  is 0. The algorithm will repeatedly find the best relays for packets until packets in  $P$  are all delivered.

As an example, we show a small area of DTVN including vessel  $a, b, c, d, e$  in Figure 5. Vessel  $a$  met its neighbors  $b, c$  at time  $t_0$  and wants to deliver its packet to  $e$ . The purple lines with arrow show the predicted trajectories of each vessel. Dotted line circles are vessels' expected location with confident degree, where  $t_0 < t_1 < t_2$  and  $t_2 - t_0 \geq \text{TTL}$ . Therefore, the one-hop expected delivery probability between  $c$  and  $e$  can be calculated by  $\mathcal{Q}_{c,e}^1 = E_{c,t_2} \times E_{e,t_2}$ . Because  $c$  and  $e$  does

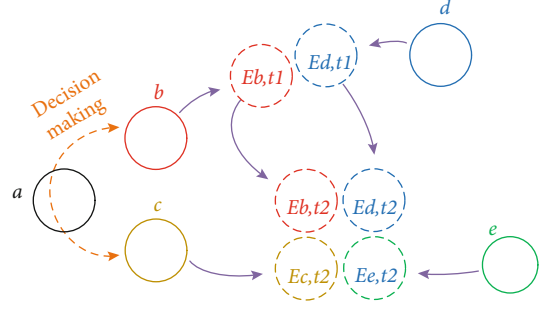


FIGURE 5: Illustration example of routing algorithm in DTVN.

not have other expected encounters. So  $\psi_{c,e}^{H-1}(t_0, t_0 + \text{TTL})$  is  $\mathcal{Q}_{c,e}^1$ . Vessels  $b$  and  $e$  have direct expected encounter at time  $t_2$ . So, the one-hop expected delivery probability between  $b$  and  $e$  can be calculated by  $\mathcal{Q}_{b,e}^1 = E_{b,t_2} \times E_{e,t_2}$ . Vessel  $b$  and  $e$  have a two-hop expected encounter situation where  $b$  meets  $d$  at  $t_1$  and  $d$  meets  $e$  at  $t_2$ . Hence,  $\mathcal{Q}_{2b,e} = (E_{b,t_1} \times E_{d,t_1})(E_{d,t_2} \times E_{e,t_2})$ . According to equation (20),  $\psi_{b,e}^{H-1} = 1 - (1 - \mathcal{Q}_{b,e}^1)(1 - \mathcal{Q}_{2b,e}^2)$ . In this case,  $\alpha$  should compare its own expected delivery probability with  $\psi_{b,e}^{H-1}$  and  $\psi_{c,e}^{H-1}$  to decide next relay.

## 6. Distributed Routing Algorithm

The previous section introduces the algorithm with global knowledge, including vessels' current location and mobility pattern. However, global knowledge is not available in distributed situations. Hence, we design the practical distributed algorithms with limited knowledge. In this section, we propose two distributed algorithms under fully distributed and cellular distributed situations.

**6.1. TPR-Distributed Design.** Compared with the global routing algorithm, the distributed algorithm further considers the actual application scenarios. So, the algorithm needs to make routing decisions utilizing incomplete knowledge of vessels and packets. In this situation, each vessel needs to have the ability to predict future contacts based on the received information and share its local information with meeting vessels. Each packet records its source node  $\zeta(p)$ , destination node  $Q_p$ ,  $H$ , and TTL.

Therefore, each vessel can construct a local meeting graph in a period by exchanged information, and they can give prediction by sliding window and form the local link set  $L_t$  in time  $t$ . When node<sub>cur</sub> meeting node  $v$  on time  $t_c$ , the optimal target of packet  $p$  in one node is shown in Equation (8).

Information exchanging between vessels is vital in the fully distributed scenario. Thus, we design the data structure of exchanged information  $inf$  considering the efficiency and cost. It includes two parts: the current state sequence and prediction model, because other information, such as the evaluation table, contact graph, and future trajectory, could be calculated by these two parts.

As shown in Algorithm 1, the node  $n$  and its meeting node  $v$  both receive and update their local information. After

**Input:**  $n, v, inf_n, inf_v, t_{cur}$   
**Variables:**  $n$ : the current node;  $v$ : the meeting node;  $inf_n, inf_v$ : information of node  $n, v$ ;  $v_n$ : stored information of node  $n$ ;  $t_{cur}$ : current time.

- 1:  $v_n = v_n \cup v$ .
- 2: Merge and Update:  $inf_n, inf_v \leftarrow inf_n \cup inf_v$ .
- 3: For each vessel in  $v_n$ , predict trajectory and evaluation table of all nodes between  $t_{cur}$  to  $t_{cur} + TTL$  by their model and sliding window.
- 4: Construct contact graph based on  $inf_n$ .
- 5: For each  $p \in P_n$ ,  $n$  calculate  $\psi_{v, \varsigma(p)}^{H-1}(t_c, t_c + TTL)$ .
- 6: Transmit each  $p$  to  $v$  when  $v$  has greater  $\psi_{v, \varsigma(p)}^{H-1}(t_c, t_c + TTL)$ .

ALGORITHM 1: TPR-distributed Algorithm.

that, both sides can construct a new local contact graph and form a local link set based on updated information. For each packet  $p$  in node  $n$ ,  $p$  will be delivered to  $v$ , when  $v$  has greater  $\psi_{v, \varsigma(p)}^{H-1}(t_c, t_c + TTL)$ . If links in the area are over link limitation, the algorithm will transfer the packet by probability rank to avoid channel collision.

**6.2. TPR-Cellular Design.** Considering the cellular coverage range along the coast, we also design a cellular-based distributed algorithm called TPRCellular. Vessel within the cellular coverage area can provide communication with base stations in high efficiency. Hence, we assume the vessel in this area has the global knowledge of all the vessels. After moving out of the cellular coverage, each vessel's knowledge will stop updating. Nevertheless, they could make the routing decision based on predicted trajectory by prediction model and sliding window. Theoretically, by utilizing a sliding window, our model can endlessly predict future trajectory without considering the accuracy. So, this algorithm's routing performance will test our model's accuracy on long-term trajectory prediction. We assume that vessels do not exchange their knowledge when meeting with each other. Except for the predicted trajectory length, the TPR-Cellular algorithm is nearly the same as the global routing algorithm.

## 7. Performance Evaluation

We evaluate the performance of TPR-DTVN with metrics defined in Section 2.1. We first present the methodology for performance evaluation, introduce compared algorithms, and finally show simulation results.

**7.1. Methodology and Experimental Setup.** We perform trajectory-driven simulations to evaluate the performance of TPR-DTVN compared with Epidemic, Random Walk (RW), and Community-Based Routing (CBR).

We train the trajectory model and perform a simulation experiment on a Linux server, which is based on ubuntu 18.04 with 64 core CPUs and 4 pieces of 1080ti GPUs. We use a total of 5123 vessels that had appeared from May 2015 to May 2018. The first 24 months' data train trajectory prediction models in default. For each packet, we randomly pick its source and destination pairs which have at least one contact before, because vessels that have not encountered each other barely have communication requirements. The number of packets varies from 200 to 1800. 5 simulations

TABLE 2: Default system parameters.

Parameter	Default value
Number of vessels	5123
Communication radius	20 km
$Hmax$	20
$TTLmax$	6 hours
Number of packets	200, 600, 1000, 1400, 1800

are carried out over 5 separate fishing months (Oct 2017, Nov 2017, Dec 2017, Jan 2018, and Feb 2018) to achieve average results. The default system parameters are shown in Table 2.

2.

**7.2. Compared Algorithms.** Three compared algorithms are briefly introduced as follows:

- (1) *Epidemic*. Packets are flooded throughout the network, which provides upper bound on delivery ratio and system cost with lower bound of delay
- (2) *Random Walk (RW)*. This algorithm randomly decides whether to forward a packet and randomly selects a relay if there is more than one neighbor vessel. In simulation experiments, the probability of forwarding a packet is set to 40%. This algorithm represents the algorithms without knowledge
- (3) *Community-Based Routing (CBR)* [12]. CBR utilizes vessels' historical contacts to form communities. Intercommunity betweenness centrality and familiarity are used to build a probability network for routing. This algorithm represents the algorithms with social knowledge

**7.3. Comparative Results.** We compare different routing algorithms' performance in delivery ratio, delay, cost, and efficiency. We vary the packet number from 200 to 1800 to simulate different network overhead. Figures 6(a) and 6(b) show six algorithms in terms of delivery ratio and average delay. The epidemic has the best performance in both metrics as expected. We can observe that TPR-Global, TPR-Distributed, TPRCellular, and CBR perform better than RW due to historical information. TPR-Global achieves a 21% higher

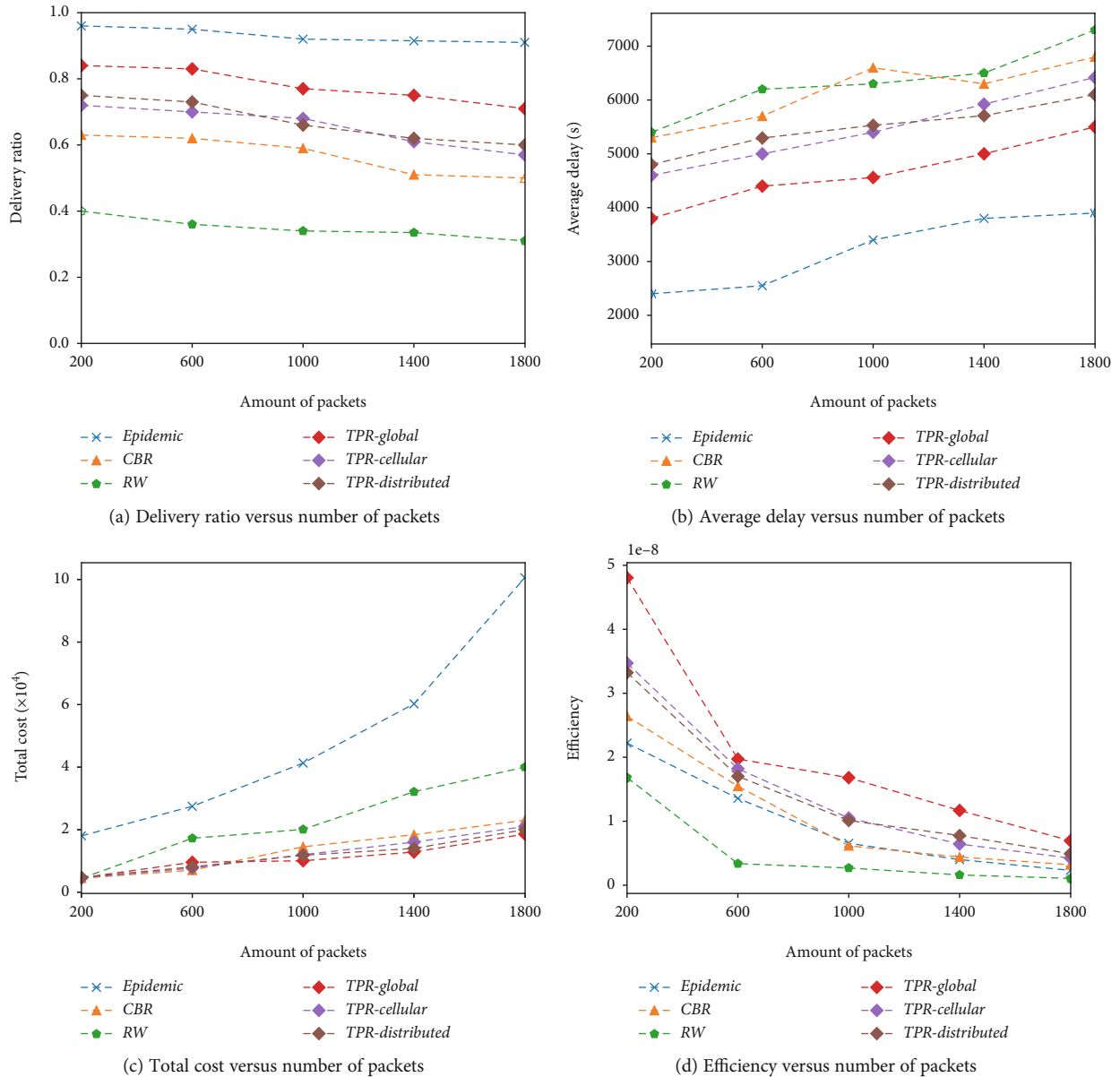


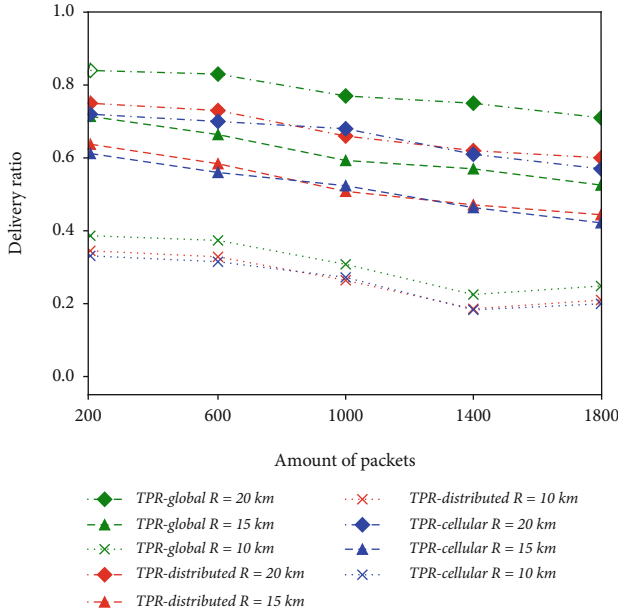
FIGURE 6: Comparative results of six algorithms.

delivery ratio than CBR with a nearly 70% delay of it. The reason is that CBR only uses contact information without considering the time factor. Although CBR can find who is familiar with the destination node, it cannot predict future contacts' approximate time. On the contrary, TPR-DTVN's probability calculation process is with the limitation of TTL, leading to more accurate results. Figure 6(c) shows the costs of six algorithms. TPR-DTVN and CBR have a lower cost than others, because wasted data transmission is avoided with historical knowledge guidance.

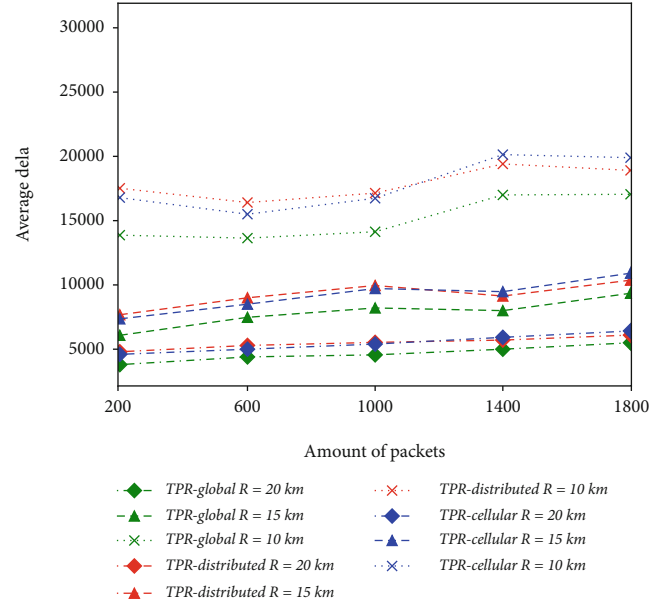
Figure 6(d) demonstrates the efficiency of all the algorithms. TPR-Global achieves the highest overall efficiency than the comparative algorithms. Because of the high cost of packet flooding, Epidemic's efficiency is uncompetitive. RW has the lowest efficiency due to its aimlessness. Also, TPR-Distributed has a 16.25% higher efficiency than TPR-Cellular.

Although historical knowledge is different, TPR-Distributed and TPR-Cellular have similar performance. Even if TPR-Cellular has the knowledge of all vessels, the vessels need to use longer prediction to get future trajectory based on its knowledge when the vessel moves out of cellular coverage. So the prediction accuracy is lower than that of TPR-Distributed. TPR-Distributed has limited knowledge, but its knowledge is up-to-date. These reasons make these two distributed algorithms have similar performance.

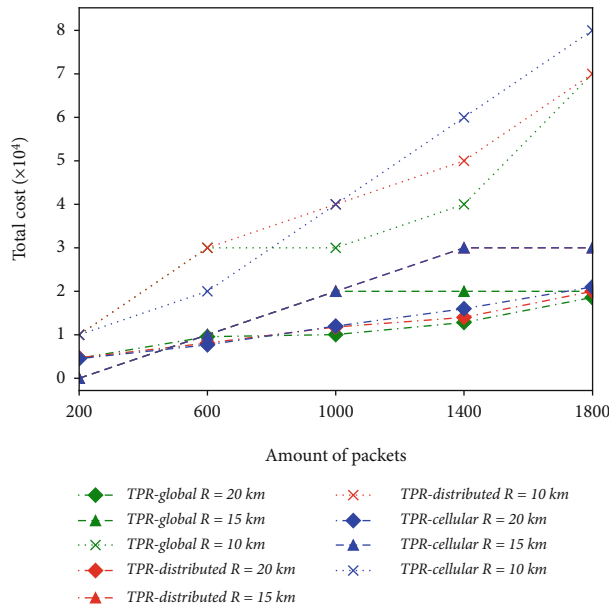
**7.4. Impact of Communication Radius.** We compare the performance of TPR-Global, TPR-Distributed, and TPR-Cellular with different communication radius, including 10 km, 15 km, and 20 km. With the decrease of communication radius, we notice that the performance in all metrics decreases in Figure 7. Delivery ratio with  $R = 10$  km has decrease 65% than that of  $R = 20$  km. Meanwhile, the average



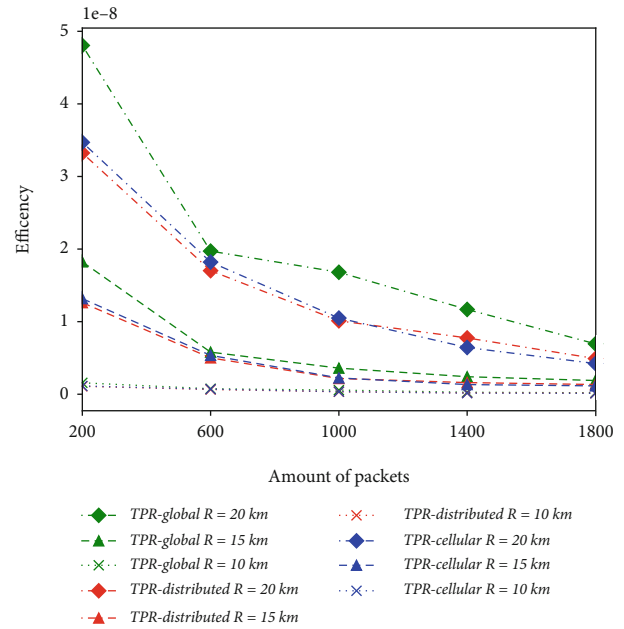
(a) Delivery ratio



(b) Average delay



(c) Total cost



(d) Efficiency

FIGURE 7: Comparative results of TPR-DTVN with different communication radius.

delay is twice of  $R = 20$  km. Compared with 20 km communication radius, these algorithms' efficiency in 15 km and 10 km average decrease 72.79% and 97.11%, respectively. The reason is that the communication opportunities of  $R = 10$  km are not enough to achieve data delivery in this vessel density. WiMAX technology can achieve maritime communication with more than 40 km communication distance, which leads to fierce channel competition. Although the shorter communication radius will have less competition, it cannot guarantee routing in our scenario. Considering the performance and channel resources, we set the communication radius to 20 km in our experiment.

**7.5. Impact of Data Set Usage.** The size of data set usage has a direct effect on TPR-Global and two distributed algorithms due to its dependency on the accuracy of the predicted trajectory. We compare the performance of different sizes of data sets, including the recent 24, 16, and 8 consecutive months. In Figure 8, with the decrease of data set, the performance of the three algorithms has notably declined. Compared train by 24 months data set, two algorithms' efficiencies in 16 months and 8 months average decrease 26.74% and 76.19%, respectively. We thought it is because the trajectory predicts model with 24 and 16 months data can learn complete mobility pattern of four seasons. 8 consecutive months cannot

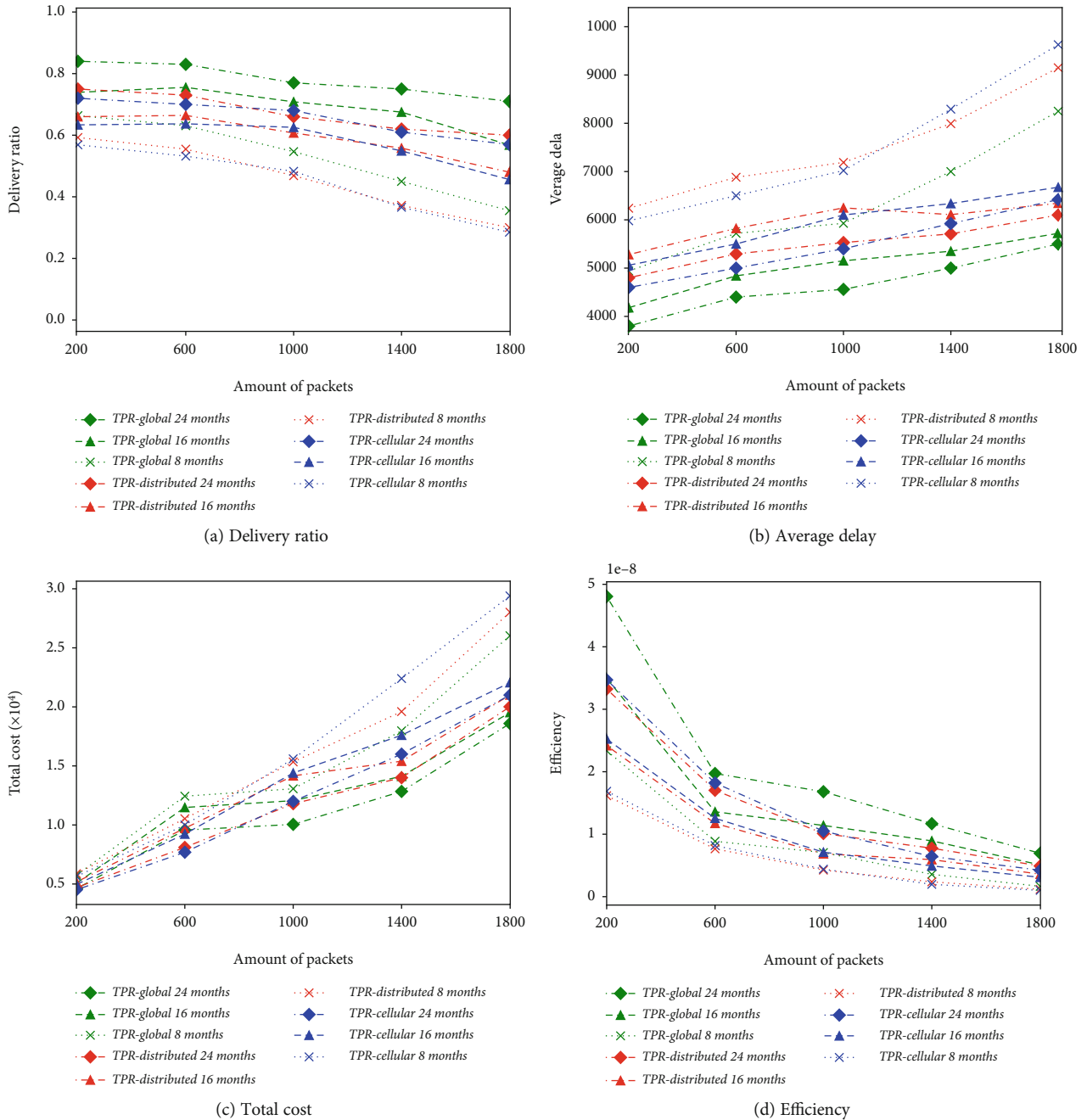


FIGURE 8: Compare the performance of TPR-DTVN in different training datasets.

cover the mobility pattern of a season, which leads to poor performance. Therefore, we should choose more than 12 consecutive months of data to train our model to obtain excellent algorithm performance.

### 8. Relate Works

DTVN, which utilizes WiMAX to achieve vessel-to-vessel or vessel-to-coast communication, is complementary to maritime wireless mesh networks [6, 23]. There have been several researches applying traditional routing algorithms in MAN-ETS. Mohsin et al. evaluate traditional algorithms' delivery ratio under different node densities and mobility behaviors

to validate the feasibility of DTVN [8]. Lambrinos et al. use real small-scale traces to simulate different protocols' performance and identify where they fall short from the perspective of our particular domain [24].

Some studies apply fixed mobility models or encounter models for the design of DTVN routing algorithms. These algorithms are often suitable for vessels with stable mobility behavior, such as container ships, cruise ships, and waste-dumping vessels. Raj et al. presume that all vessels obey the Gaussian-Markov mobility model [7]. The forwarding metrics are determined for routing on the basis of this assumption. Lanepost [25] and Mar-DTN [26] are standard graphic-based optimal routing algorithms. These algorithms



build an opportunistic routing graph based on each vessel's fixed expected routes and use the graph for forwarding decisions. Liu et al. [27] and Qin et al. [28] study delay-tolerant routing problems in the seaway model that all vessels are moving in fixed seaways. Vessels predict the speed and arrival time based on proper models and utilize them for decisions.

Mining personalized characteristics of each node in the network are more beneficial for routing algorithm performance than fixed model [13, 29, 30]. Some vessels' moving patterns, like fishery vessels, are easier to be affected by objective factors, which cannot directly apply fixed models. Nodes in the network use neighbors' personalized models for routing decisions. These models need to be dynamically updated based on historical data. FBR and CBR [12] dynamically construct communities based on vessels' recent contacts. Based on unique social relationships, vessels build a probability network for efficient packet delivery.

## 9. Conclusion

This paper proposed a set of routing algorithms called TPRDTVN for efficient data transmission in vessel networks. TPR-DTVN uses a forwarding metric that characterizes the expected delivery probability of relay nodes. To address the challenge of long-term trajectory prediction, we design a Bi-LSTM-based trajectory prediction model and an evaluation method to get the practicable predicted trajectory. Then, we evaluate expected delivery probability with predicted information. Extensive trace-driven simulations show that our algorithms can achieve a higher delivery ratio with lower cost and transmission delay.

## Data Availability

The data is not available.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

C. Liu and Y. Li contributed equally to this work and should be regarded as co-first authors. R. Jiang and Y. Du are corresponding authors.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China Grant No. 61902367, 62072287, and 41976185; the China Postdoctoral Science Foundation 2020M682240; the Natural Science Foundation of Shandong Grant No. ZR2018BF006; and the National Key R&D Program 2016YFC1401900.

## References

[1] K. Metcalfe, N. Breheret, E. Chauvet et al., "Using satellite ais to improve our understanding of shipping and fill gaps in

ocean observation data to support marine spatial planning," *Journal of Applied Ecology*, vol. 55, no. 4, pp. 1834–1845, 2018.

- [2] M. C. Domingo, "An overview of the internet of underwater things," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1879–1890, 2012.
- [3] P. Carnie, S. Kenny, E. Browell, F. Cheng, H.-C. Fang, and A. Incecik, "Global marine trends 2030: implications for naval ship technology," in *Pacific 2013 International Maritime Conference: The commercial maritime and naval defence showcase for the Asia Pacific*, p. 241, Australia, 2013.
- [4] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G. B. Huang, "Exploiting ais data for intelligent maritime navigation: a comprehensive survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–24, 2016.
- [5] F. Li, D. Yu, H. Yang, J. Yu, and X. Cheng, "Multi-armed-banditbased spectrum scheduling algorithms in wireless networks: a survey," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 24–30, 2020.
- [6] C. Liu, Y. Li, R. Jiang, F. Hong, and Z. Guo, "Oceannet: a low-cost large-scale maritime communication architecture based on d2d communication technology," in *Proceedings of the ACM Turing Celebration Conference-China*, pp. 1–6, New York, NY, USA, 2019.
- [7] D. Raj, M. V. Ramesh, and S. Duttgupta, "Delay tolerant routing protocol for heterogeneous marine vehicular mobile ad-hoc network," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 461–466, Kona, HI, USA, 2017.
- [8] R. J. Mohsin, J. Woods, and M. Q. Shawkat, "Density and mobility impact on manet routing protocols in a maritime environment," in *2015 Science and Information Conference (SAI)*, pp. 1046–1051, London, UK, 2015.
- [9] J. Dhivvyaa, S. N. Rao, and S. Simi, "Towards maximizing throughput and coverage of a novel heterogeneous maritime communication network," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 1-2, New York, NY, USA, 2017.
- [10] S. Qin, G. Feng, W. Qin, Y. Ge, and J. S. Pathmasuntharam, "Performance modeling of data transmission in maritime delay-tolerant networks," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1109–1114, Shanghai, China, 2013.
- [11] T. Yang, H. Liang, N. Cheng, R. Deng, and X. Shen, "Efficient scheduling for video transmissions in maritime wireless communication networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4215–4229, 2014.
- [12] Q. Bing, R. Jiang, and F. Hong, "Exploiting social network characteristics for efficient routing in ocean vessel ad hoc networks," in *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, London, United Kingdom, 2019.
- [13] Y. Zhu, Y. Wu, and B. Li, "Trajectory improves data delivery in urban vehicular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 1089–1100, 2013.
- [14] C. Liu, Z. Sun, J. Liu, H. Huang, Z. Guo, and Y. Feng, "Vstp: vessel spatio-temporal contact pattern detection based on mapreduce," *Eurasip Journal on Wireless Communications and Networking*, vol. 2017, no. 1, 2017.
- [15] C. Liu, Y. Liu, Z. Guo, X. Wang, and S. Guo, "Mr-similarity: parallel algorithm of vessel mobility pattern detection," in *Big*

- Data Computing and Communications*, pp. 225–235, Springer, 2016.
- [16] C. Liu, S. Guo, Y. Feng, F. Hong, H. Huang, and Z. Guo, “L-vtp: long-term vessel trajectory prediction based on multi-source data analysis,” *Sensors*, vol. 19, no. 20, article 4365, 2019.
- [17] P. Borkowski, “The ship movement trajectory prediction algorithm using navigational data fusion,” *Sensors*, vol. 17, no. 6, article 1432, 2017.
- [18] S. Guo, C. Liu, Z. Guo, Y. Feng, F. Hong, and H. Huang, “Trajectory prediction for ocean vessels base on k-order multivariate markov chain,” in *Wireless Algorithms, Systems, and Applications*, pp. 140–150, Springer, 2018.
- [19] Z. Sun, K. Hu, T. Hu, J. Liu, and K. Zhu, “Fast multi-label low-rank linearized svm classification algorithm based on approximate extreme points,” *IEEE Access*, vol. 6, pp. 42319–42326, 2018.
- [20] Y. Zhou and L. Shao, “Vehicle re-identification by adversarial bi-directional lstm network,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 653–662, Lake Tahoe, NV, USA, 2018.
- [21] F. Altche and A. de La Fortelle, “An lstm network for highway trajectory prediction,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 353–359, Yokohama, Japan, 2017.
- [22] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] M.-T. Zhou, V. D. Hoang, H. Harada et al., “Triton: highspeed maritime wireless mesh network,” *IEEE Wireless Communications*, vol. 20, no. 5, pp. 134–142, 2013.
- [24] L. Lambrinos, C. Djouvas, and C. Chrysostomou, “Applying delay tolerant networking routing algorithms in maritime communications,” in *2013 IEEE 14th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–6, Madrid, Spain, 2013.
- [25] X. Geng, Y. Wang, H. Feng, and L. Zhang, “Lanepost: lane-based optimal routing protocol for delay-tolerant maritime networks,” *China Communications*, vol. 14, no. 2, pp. 65–78, 2017.
- [26] P. Kolios and L. Lambrinos, “Optimising file delivery in a maritime environment through inter-vessel connectivity predictions,” in *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 777–783, Barcelona, Spain, 2012.
- [27] C. Liu, Z. Guo, F. Hong, and K. Wu, “Dcep: data collection strategy with the estimated paths in ocean delay tolerant network,” *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, Article ID 518439, 2014.
- [28] S. Qin, G. Feng, W. Qin, Y. Ge, and J. S. Pathmasuntharam, “Performance evaluation of data transmission in maritime delaytolerant-networks,” *IEICE Transactions on Communications*, vol. 96, no. 6, pp. 1435–1443, 2013.
- [29] Y. Yuan, F. Li, D. Yu et al., “Fast fault tolerant sampling via random walk in dynamic networks,” in *2019 IEEE 39th International Conference on Distributed Computing Systems*, pp. 536–544, Dallas, TX, USA, 2020.
- [30] H. Zhu, M. Dong, S. Chang, Y. Zhu, M. Li, and X. S. Shen, “Zoom: scaling the mobility for fast opportunistic forwarding in vehicular networks,” in *2013 Proceedings IEEE INFOCOM*, pp. 2832–2840, Turin, Italy, 2013.