Journal of Database Management
Volume 34 • Issue 3

CNN-Based Vessel Meeting Knowledge Discovery From AIS Vessel Trajectories

Peng Chen, Dalian Neusoft University of Information, China
Shuang Liu, Dalian Minzu University, China*
Niko Lukač, University of Maribor, Slovenia

ABSTRACT

How to extract a collection of trajectories for different vessels from the raw AIS data to discover vessel meeting knowledge is a heavily studied focus. Here, the AIS database is created based on the raw AIS data after parsing, noise reduction and dynamic Ramer-Douglas-Peucker compression. Potential encountering trajectory pairs will be recorded based on the candidate meeting vessel searching algorithm. To ensure consistent features extracted from the trajectories in the same time period, time alignment is also adopted. With statistical analysis of vessel trajectories, sailing segment labels will be added to the input feature. All motion features and sailing segment labels are combined as input to one trajectory similarity matching method based on convolutional neural network to recognize crossing, overtaking or head-on situations for each potential encountering vessel pair, which may lead to collision if false actions are adopted. Experiments on AIS data show that our method is effective in classifying vessel encounter situations to provide decision support for collision avoidance.

KEYWORDS

AIS Data, CNN, Dynamic Rammer-Douglas-Peucker, Knowledge Discovery, Maneuvering Pattern, Traffic Pattern, Trajectory

INTRODUCTION

Automatic identification system (AIS) data, as a fundamental source of information, plays a very important role in monitoring vessel activity in a maritime surveillance system. However, the behavior of vessels is difficult to recognize. How to take full advantage of available AIS data to discover vessel maneuvering patterns to provide decision support for collision avoidance or complex event dealing has attracted much attention in recent years. Trajectory analysis from AIS sources is an essential branch of the study. There are four levels concerning AIS research: data level, method level, knowledge level and decision level.

Regarding the data level, there are data cleaning, data compression, interpolation or time alignment, data quality, data de-noising and so on. For example, Wei et al. (2020) designed a novel

DOI: 10.4018/JDM.321636 *Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

algorithm considering the spatial and motion features of trajectories to compress AIS trajectories based on ship behavior characteristics. Tang et al. (2021) proposed an ADP (Adaptive-threshold Douglas-Peucker) algorithm based on DP (Douglas-Peucker) algorithm to determine the key points of each trajectory through the threshold change rate for ship trajectory compression. Li et al. (2020) used U-Net convolutional networks to construct AIS-based vessel trajectories to obtain the rich skip connections in the network and make great use of historical trajectories. Guo et al. (2021) presented an improved kinematic interpolation for AIS trajectory reconstruction, which integrated data preprocessing and interpolation that considered the ships' kinematic information. With automatic ship reporting systems, Greidanus et al. (2016) discussed how to complete a wide-area maritime situational picture. To combine different data processing methods, Zhang et al. (2016) proposed a new scheme for implementing the Douglas-Peucker (DP) algorithm to complete simplification of AIS trajectories and presented a new AIS-based minimum ship domain evaluation method for simplification threshold determination.

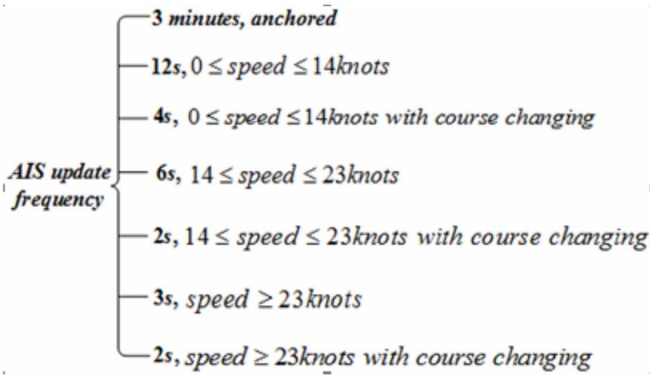
At the method level, there are clustering methods, machine learning methods, statistical methods and so on. For example, Wang et al. (2021) proposed a ship AIS trajectory clustering method based on Hausdorff distance and Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) to provide insightful knowledge for traffic management and operation optimization. Burger et al. (2020) compared the performance of the Discrete Kalman filter (DKF) and the Linear Regression Model (LRM) to conclude that LRM is a computationally simpler method for trajectory prediction. Jadidi (2021) presented an enhanced density-based spatial clustering of applications with noise (DBSCAN) method to model vessel behaviours based on trajectory point data.

In terms of knowledge level, there are trajectory analysis, maritime anomaly detection, trajectory similarity measuring, collision avoidance analysis and so on. For example, Gao & Shi (2020) put forward a sequence conditional generative adversarial network (Seq-CGAN) to learn how to generate appropriate anthropomorphic collision avoidance decisions and bypass the process of ship collision. Zhang & Furusho (2020) combined rule-based method and neural based method for data preprocessing for guidance direction. From a collection of trajectories, Lee et al. (2021) designed an AIS data-driven approach to analyze the pattern of ship trajectories in ports using the DBSCAN algorithm. D'Afflisio et al. (2021) proposed a derivation of anomaly detection rules based on the Generalized Likelihood Ratio Test (GLRT) and the Model Order Selection (MOS) for collision avoidance and maritime surveillance. Singh & Heymann (2020) proposed a multi-class artificial neural network (ANN)-based anomaly detection framework to classify intentional and non-intentional AIS on-off switching anomalies. Iphar et al. (2020) presented a method based on expert knowledge for the risk assessment of cyberthreats in maritime transportation data. Zhen et al. (2017) proposed a method combining vessel trajectory clustering and a naive Bayes classifier to learn the typical vessel sailing pattern to detect anomalous vessel behavior in the maritime surveillance system. Based on exploring maritime trajectory data for anomalous behavior detection, Lei (2016) proposed a so-called MT-MAD framework for maritime trajectory modeling and anomaly detection. For vessel motion pattern recognition, Chen et al. (2019) put forward a method to classify collision avoidance situations, and Sun et al. (2018) tried to mine spatial–temporal motion patterns for vessel recognition. There are also similar studies in Le et al. (2016), Yan et al. (2020), and Fu et al. (2017). Because AIS data contains information such as vessel identity, position, speed, and course for vessels, most of these researchers studied trajectory data or derived motion patterns with certain statistical methods. In most cases, feature extraction is too simple to find vessel maneuvering actions or only to find limited types of anomalies.

With the rapid development of maritime traffic, over 580,000 vessels installed AIS receivers and emitted broadcast information continuously throughout the world. AIS data updates with a frequency of every 2 to 12 seconds while the vessel is underway and every 3 minutes while the vessel is anchored, which is shown in Figure 1.

Except for the anchored status, AIS updates every 4.83 seconds for the other six cases. Therefore, in the AIS aggregation process, the original data may be very large if the observation period is long

Figure 1.
AIS update frequency illustration



enough. For example, the whole sailing time for a vessel is six days, and the total amount of data for this vessel is approximately 107,329 records. To cope with the large amount of available AIS data, one quick dynamic Ramer–Douglas–Peucker is adopted to obtain a reduction of the original AIS data. Along with a normal course, some redundant locations can be discarded. Then, trajectories for different vessels can be extracted from the AIS database. With statistical analysis of vessel trajectories, sailing segment labeling of critical vessel motions, such as maneuvering actions of a sharp left turn or a sharp right turn, is completed. Next, candidate encountering vessel pairs are selected based on the candidate meeting vessel searching algorithm. Time alignment based on the interpolation operation is also adopted to prepare for the input of the CNN (Convolutional Neural Network, CNN). For each pair of potential encounter trajectories, the CNN is called to calculate the similarity of the trajectory in the vessel encounter situation knowledge base.

The rest of the paper is organized as follows. A step-by-step procedure for preprocessing vessel trajectories from AIS data is presented first. Then, sailing segment label modeling from the trajectory is given. The whole similarity matching framework from AIS trajectory data is described in the next section. Next, the experimental results and discussions are given. Finally, the conclusions of our work and future research directions are listed.

TRAJECTORY EXTRACTION FROM AIS DATA AND PREPROCESSING

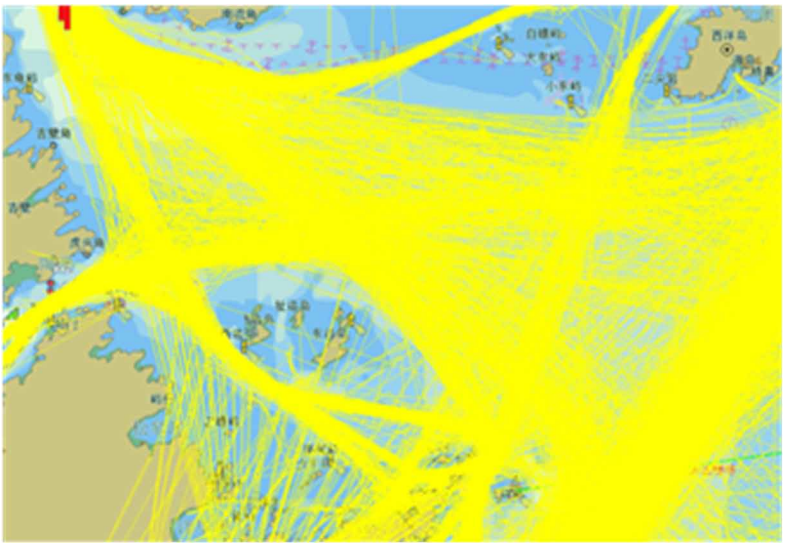
Figure 2 shows one AIS data visualization result in one marine area of China in one month. Without compression, it is difficult to deal with the overwhelming data for long time intervals.

DYNAMIC D-P ALGORITHM

Given a trajectory composed of line segments, the Ramer-Douglas-Peucker algorithm (RDP) attempts to find a similar trajectory with fewer points. The algorithm first joins the first and last vertices of the original trajectory to complete a single edge for crude simplification. It then computes the distance of all intermediate vertices to that edge. The vertex with the furthest distance (also larger than a specified tolerance) from that edge will be marked as a key point and added to the simplification. In the simplification process, the same operation will recur for each edge until all vertices of the original polyline are within the tolerance of the simplification results. The process is illustrated in Figure 3.

The algorithm recursively divides the trajectory. It first marks the first point p_1 and last point p_{13} to be kept. It then finds the point p_7 with the furthest distance from the trajectory segment p_1p_{13}.

Figure 2.
Visualization AIS data in one marine area in one month

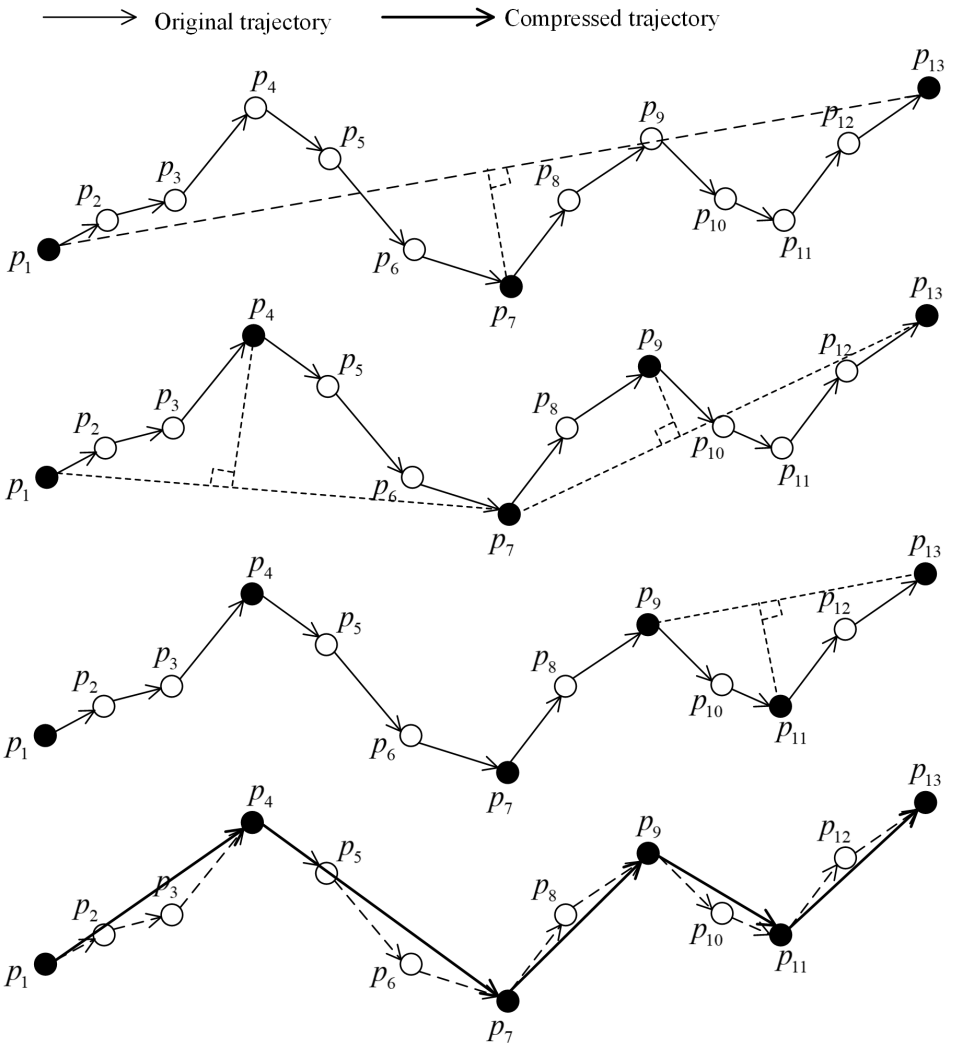


Subsequent operations are similar. The algorithm marks the first point and the furthest point to complete the next simplification. Then, it marks the furthest point and the last point to complete the same simplification. Certainly, the simplification result includes the furthest point being marked as kept. When the recursion is completed, the simplified result trajectory can be obtained by connecting the points being marked as kept. Here, the final simplified trajectory is $p_1 \rightarrow p_4 \rightarrow p_7 \rightarrow p_9 \rightarrow p_{11} \rightarrow p_{13}$.

The traditional D-P algorithm retains effective motion trajectory features for the vessel, but it fails to retain speed features for the vessel. For example, the vessel moves with varying speed along a line, and the traditional D-P algorithm fails to recognize the critical points with velocity change. In the simplification process, the velocity changing points are considered as the velocity keeping points, and they may be discarded. Alternatively, in the case of anchoring, the vessel is represented as one point in the 2D dimension. Because of the loss of the time feature for the anchored point, only a few points remain, which may lead to the loss of some anchored points. These critical points are very important for determining vessel status or providing decision support, which may be discarded by the traditional D-P algorithm. So the improved iterative point adaptive dynamic D-P (Douglas Peucker) algorithm is adopted, and the state stationary points with the course change exceeding the threshold value or the speed change exceeding the threshold value are added. No matter whether the distance between the state stationary point and the track line in the Douglas Puck algorithm is greater than the distance threshold, the point will be retained in the final compressed track to ensure that all critical points are retained in the final compressed trajectory.

Each trajectory point records the vessel identity, position, speed and course for the vessel. There is one *mmsi* (Maritime Mobile Service Identify, abbreviated *mmsi*) field in each AIS broadcast information, denoting the vessel identity. Without loss of generality, a vessel trajectory is an aggregation of its broadcasting AIS information, represented as $TV_{mmsi} = \{p_1, p_2, \cdots, p_n\}$, with *mmsi* as its unique ID. Here, each trajectory point p_i is defined as $p_i = \{\varphi_i, \lambda_i, t_i, SOG, COG\}$, where *SOG* refers to speed over ground and *COG* refers to course over ground. This is the representation of a trajectory with no compression. After compression, the new trajectory for the same vessel is denoted as $Q_{mmsi} = \{p_{k_1}, p_{k_2}, \cdots, p_{k_m}\}, m \leq n$, and the index set is $I = \{k_1, k_2, \cdots, k_m\}, k \in \{1, \cdots, n\}$.

Figure 3.
Illustration of traditional D-P algorithm



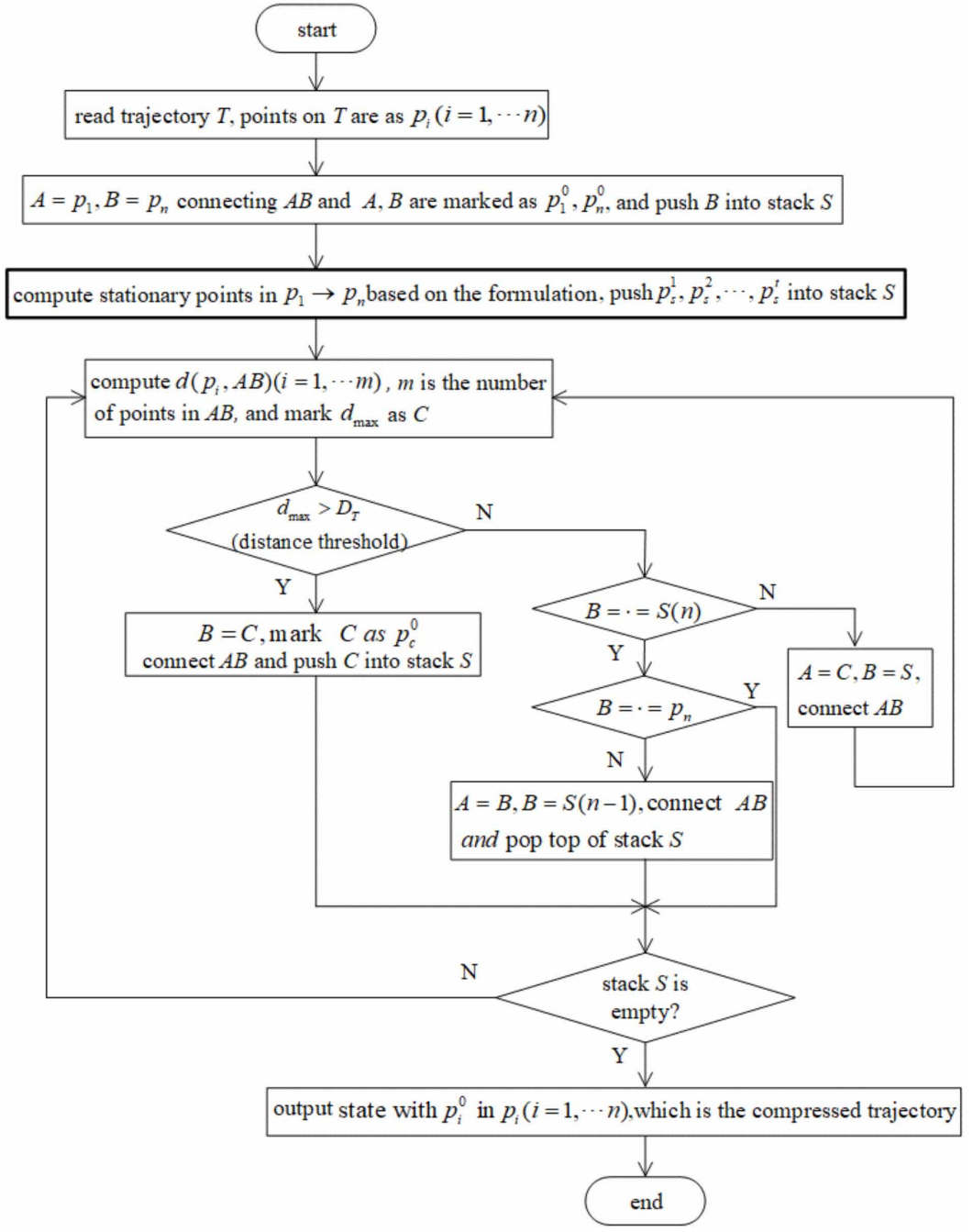
The new improved algorithm considers the data points of acceleration, deceleration and turning during the vessel's traveling, which are often the most important dynamic behavior points in the vessel's trajectory. The determination of speed or course change threshold is calculated according to the speed and course statistics of the current segment.

The computation process for our dynamic D-P algorithm is as follows, which is shown in Figure 4.

Step 1: According to the given starting point and ending point of the trajectory, all state stationary points in the trajectory are calculated. The start point, state stationary point and end point are recorded as the segment mark points.

Step 2: Connects the start point and end point of each segment trajectory and calculates the distance between all data points on the curve and the segment trajectory to find the maximum distance $D_{\max}$, comparing the maximum distance with the predefined distance threshold D_T.

Figure 4.
Flowchart of our dynamic D-P algorithm



Step 3: All the intermediate data points on the trajectory will be dropped if $D_{\max} < D_T$ and the starting point and ending point are connected directly. This line will be regarded as the approximation of the trajectory. Then, the processing is completed.

Step 4: The point is regarded as a key point on the result trajectory if $D_{\max} > D_T$, and the trajectory is split into two parts with the data point as the boundary. Repeat Steps 2 and 3 for the two parts of the curve until all segments satisfy $D_{\max} < D_T$.

Step 5: The trajectories formed by connecting the segmentation points in turn can be used as the approximation of the original trajectory when all the sub-trajectories are processed.

Figure 5 is an illustration of our dynamic D-P algorithm. In the figure, blue points are state stationary points. The first point p_1, the last point p_{13}, and the state stationary points p_3, p_7, p_{11} are first retained as key points. Then, based on the distance from the point to the trajectory segment, point p_4 and point p_9 are selected. The final simplified trajectory is $p_1 \rightarrow p_3 \rightarrow p_4 \rightarrow p_7 \rightarrow p_9 \rightarrow p_{11} \rightarrow p_{13}$.

Figure 6 is the application result of our dynamic D-P algorithm. With different thresholds, different compression results can be obtained. Although the computation cost is larger than that of the traditional D-P compression algorithm, our dynamic D-P algorithm performs better in maintaining key navigation points.

Figure 5.
Illustration of our dynamic D-P algorithm

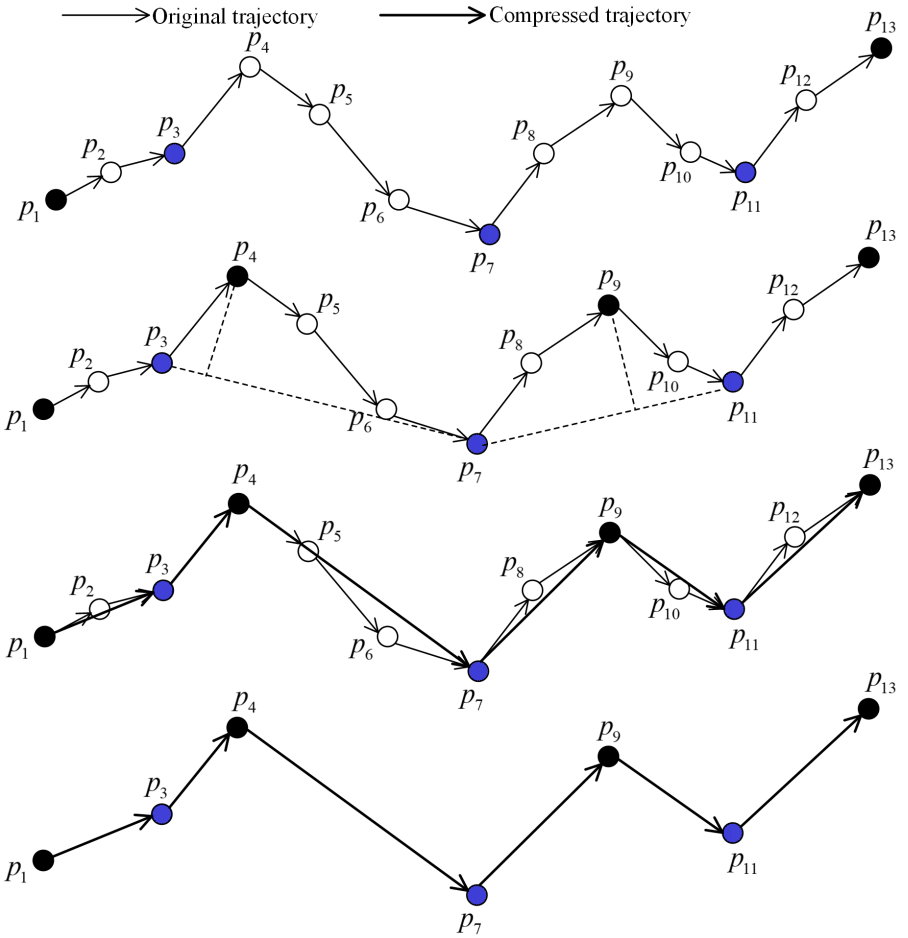


Figure 6.
Application result of our dynamic D-P compression algorithm

1. Original AIS trajectory data before compression
2. Compression result with a compression ratio of 90.39%
3. Compression result with a compression ratio of 29.06%
4. Compression result with a compression ratio of 5.12%

After data compression for the original AIS data, the following process is adopted. Data of the vessels are first ordered by *mmsi* and then by timestamp. The next point in the trajectory is found to make a line segment for every part of the trajectory. The distance traveled with speed for every segment is calculated. For one vessel, the query retrieves its trajectory as a set of line segments. For each segment, the query calculates the speed and distance.

Compared to the traditional D-P algorithm, our dynamic D-P algorithm retains key points for collision avoidance analysis. A comparison result of the proportion of key points in the compressed trajectories is shown in Figure 7. Here, DATA1 consists of 2355 AIS trajectories, DATA2 consists of 5632 AIS trajectories, DATA3 consists of 8345 AIS trajectories and DATA4 consists of 12356 AIS trajectories. Clearly, our method remains more key points than the traditional one.

LAGRANGE INTERPOLATION

To learn knowledge for meeting situations, the pair trajectories of two vessels are the basic requirements. However, some points for one vessel are missing because of the receiving frequency or the quality of AIS data. For the same time period, the same number of position points are needed, as shown in Figure 8.

There are 13 points in the first trajectory, but there are only 10 points in the second trajectory. Pair trajectory, that is, the same points in two trajectories, are used as the input for our trajectory similarity algorithm. This is the reason why interpolation is necessary for our preprocessing.

Before discussing how to call Lagrange interpolation, the first order difference quotient and second order difference quotient are listed as follows.

Figure 7.
Comparison result for the traditional D-P algorithm and our dynamic D-P compression algorithm

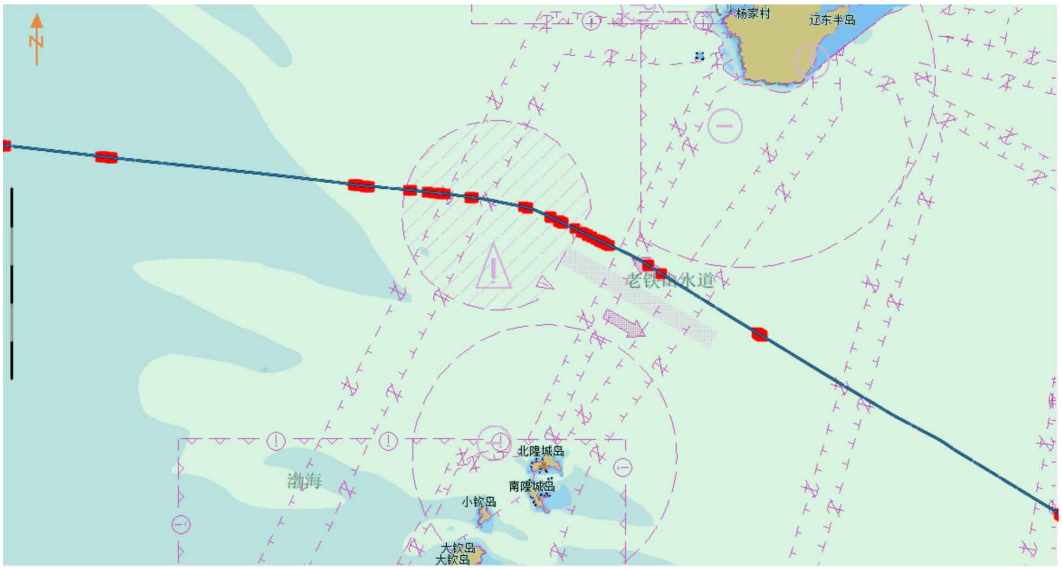
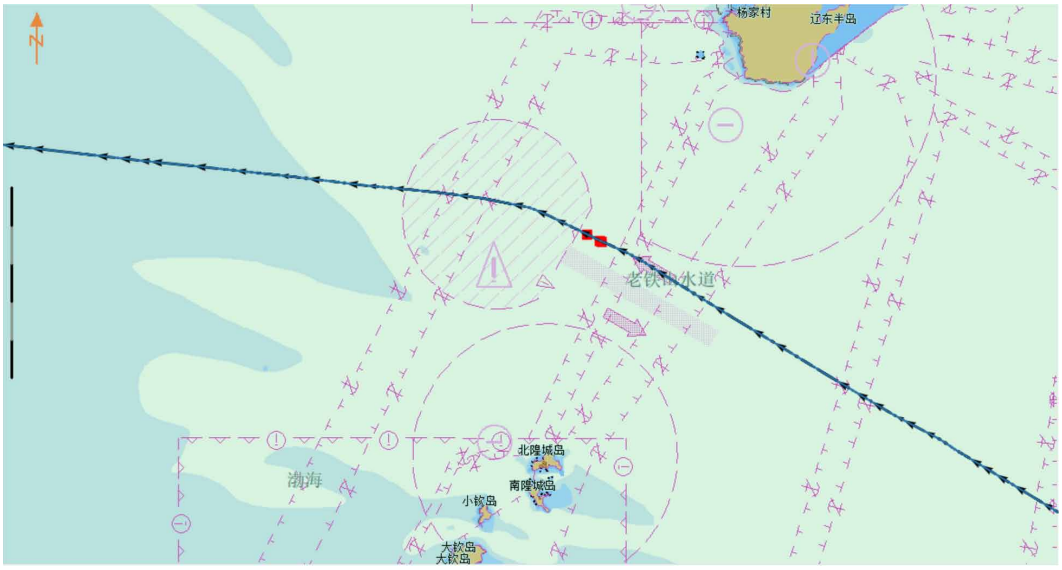


Figure 8.
An example of time alignment for two vessels



For three time points t_i, t_j, t_k satisfying $t_i < t_j < t_k$, the corresponding longitude and latitude are denoted as (φ_i, λ_i), (φ_j, λ_j), and (φ_k, λ_k). Its zero-order difference quotient is computed as Eq. (1):

$$\begin{aligned} f[t_i]_\lambda &= f(\lambda_i) = \lambda_i \\ f[t_k]_\lambda &= f(\lambda_k) = \lambda_k \\ f[t_j]_\lambda &= f(\lambda_j) = \lambda_j \\ f[t_i]_\varphi &= f(\varphi_i) = \varphi_i \\ f[t_k]_\varphi &= f(\varphi_k) = \varphi_k \\ f[t_j]_\varphi &= f(\varphi_j) = \varphi_j \end{aligned} \tag{1}$$

For its first-order difference quotient, the longitude velocity per unit time is computed as Eq. (2):

$$\begin{aligned} f[t_i, t_j]_\lambda &= \frac{f[t_j]_\lambda - f[t_i]_\lambda}{t_j - t_i} = \frac{\lambda_j - \lambda_i}{t_j - t_i} \\ f[t_j, t_k]_\lambda &= \frac{f[t_k]_\lambda - f[t_j]_\lambda}{t_k - t_j} = \frac{\lambda_k - \lambda_j}{t_k - t_j} \end{aligned} \tag{2}$$

Its latitude velocity per unit time is computed as Eq. (3):

$$\begin{aligned} f[t_i, t_j]_\varphi &= \frac{f[t_j]_\varphi - f[t_i]_\varphi}{t_j - t_i} = \frac{\varphi_j - \varphi_i}{t_j - t_i} \\ f[t_j, t_k]_\varphi &= \frac{f[t_k]_\varphi - f[t_j]_\varphi}{t_k - t_j} = \frac{\varphi_k - \varphi_j}{t_k - t_j} \end{aligned} \tag{3}$$

Its second-order difference quotient is computed with Eq. (4):

$$\begin{aligned} f[t_i, t_j, t_k]_\lambda &= \frac{f[t_j, t_k]_\lambda - f[t_i, t_j]_\lambda}{t_k - t_i} = \frac{\dfrac{\lambda_k - \lambda_j}{t_k - t_j} - \dfrac{\lambda_j - \lambda_i}{t_j - t_i}}{t_k - t_i} \\ f[t_i, t_j, t_k]_\varphi &= \frac{f[t_j, t_k]_\varphi - f[t_i, t_j]_\varphi}{t_k - t_i} = \frac{\dfrac{\varphi_k - \varphi_j}{t_k - t_j} - \dfrac{\varphi_j - \varphi_i}{t_j - t_i}}{t_k - t_i} \end{aligned} \tag{4}$$

Based on this assumption, the following formula holds:

$$t_k - t_i > 0, t_j - t_i > 0 \ and \ t_k - t_j > 0 \tag{5}$$

Therefore, the positive and negative sign of the second-order difference quotient is decided by its numerators $(\lambda_m - \lambda_n)(m > n)$ and $(\varphi_m - \varphi_n)(m > n)$ in Eq. (4).

If $(\lambda_k - \lambda_j) < 0, (\lambda_j - \lambda_i) > 0$, then $f[t_i, t_j, t_k]_\lambda < 0$, and the curve is convex.
If $(\lambda_k - \lambda_j) > 0, (\lambda_j - \lambda_i) < 0$, then $f[t_i, t_j, t_k]_\lambda > 0$, and the curve is concave.

If $(\varphi_k - \varphi_j) < 0, (\varphi_j - \varphi_i) > 0$, then $f[t_i, t_j, t_k]_\varphi < 0$, and the curve is convex.
If $(\varphi_k - \varphi_j) > 0, (\varphi_j - \varphi_i) < 0$, then $f[t_i, t_j, t_k]_\varphi > 0$, and the curve is concave.

Figure 9 is taken as an example to explain this.
From Figure 9, Eq. (6) to Eq. (8) can be obtained:

$$\lambda_i - \lambda_{i-1} < 0 \ \&\& \ \lambda_{i-1} - \lambda_{i-2} > 0 \Rightarrow \prod_{i-1} = \frac{\frac{\lambda_i - \lambda_{i-1}}{t_i - t_{i-1}} - \frac{\lambda_{i-1} - \lambda_{i-2}}{t_{i-1} - t_{i-2}}}{t_i - t_{i-2}} < 0 \tag{6}$$

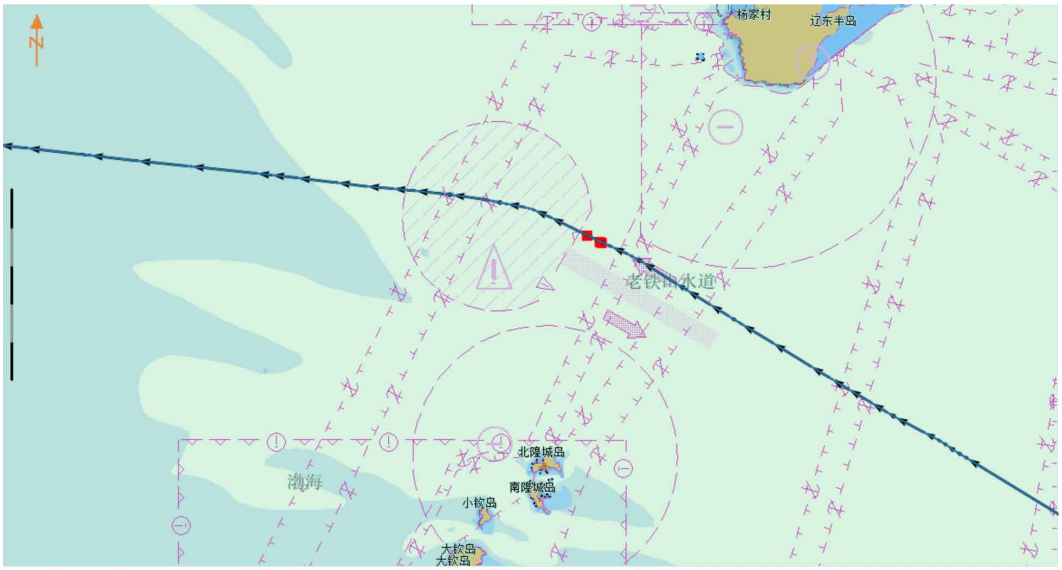
$$\lambda_{i+1} - \lambda_i > 0 \ \&\& \ \lambda_i - \lambda_{i-1} < 0 \Rightarrow \prod_i = \frac{\frac{\lambda_{i+1} - \lambda_i}{t_{i+1} - t_i} - \frac{\lambda_i - \lambda_{i-1}}{t_i - t_{i-1}}}{t_{i+1} - t_{i-1}} > 0 \tag{7}$$

$$\lambda_{i+2} - \lambda_{i+1} < 0 \ \&\& \ \lambda_{i+1} - \lambda_i > 0 \Rightarrow \prod_{i+1} = \frac{\frac{\lambda_{i+2} - \lambda_{i+1}}{t_{i+2} - t_{i+1}} - \frac{\lambda_{i+1} - \lambda_i}{t_{i+1} - t_i}}{t_{i+2} - t_i} < 0 \tag{8}$$

Therefore, the curve is convex at point p_{i-1}, concave at point p_i and convex at point p_{i+1}. The convexity of a curve is determined by the sign of its second-order difference quotient. If the sign is smaller than zero, the trajectory curve is convex. If the sign is larger than zero, the trajectory curve is concave. When the second-order difference quotient of a curve changes, there is a turning point. The reasoning of latitude direction is the same as above.

Here, the product of the second-order difference quotient of two adjacent points is used for determination. If the product is less than 0, there is a turning point. When two adjacent points have the same sign of second-order difference quotient, the vessel can be approximately considered as a straight-line motion, which can be directly solved by bilinear interpolation. When two adjacent points

Figure 9.
Illustration for solving the two-order difference quotient



have different signs of the second-order difference quotient, the vessel's running trajectory can be considered a quadratic curve, which can be solved by Lagrange quadratic interpolation.

According to the AIS data, when the second-order difference quotient of two adjacent points is the same sign, it can be approximately considered that the ship moves in a straight line. With trajectory data of the two points before and after known as $p_i(\varphi_i, \lambda_i, t_i, SOG_i, COG_i)$ and $p_j(\varphi_j, \lambda_j, t_j, SOG_j, COG_j)$, the interpolation is computed as Eq. (9) at time t_k:

$$\begin{aligned}
\varphi_k &= \frac{\varphi_j - \varphi_i}{t_j - t_i}(t_j - t_i) + \varphi_i \\
\lambda_k &= \frac{\lambda_j - \lambda_i}{t_j - t_i}(t_j - t_i) + \lambda_i \\
COG_k &= \frac{COG_j - COG_i}{t_j - t_i}(t_j - t_i) + COG_i \\
SOG_k &= \frac{SOG_j - SOG_i}{t_j - t_i}(t_j - t_i) + SOG_i
\end{aligned} \tag{9}$$

When the sign of the second-order difference between two adjacent points of AIS data is different, the ship's trajectory can be considered a quadratic curve, which can be solved by Lagrangian quadratic interpolation.

Suppose that the trajectory data of the three adjacent points are p_{i-1}, p_i and p_j, and the interpolation time is *t*. First, the Gauss Kruger projection transformation is carried out to obtain the plane coordinates $p'_{i-1}(x_{i-1}, y_{i-1}, t_{i-1}, SOG_{i-1}, COG_{i-1})$, $p'_i(x_i, y_i, t_i, SOG_i, COG_i)$ and $p'_j(x_j, y_j, t_j, SOG_j, COG_j)$.

Here, the latitude calculation is taken as an example. If $y(t_{i-1}) = y_{i-1}$, $y(t_i) = y_i$, and $y(t_j) = y_j$, the latitude calculation formula of interpolation time *t* is:

$$y(t) = la_0(t)y_{i-1} + la_1(t)y_i + la_2(t)y_j \tag{10}$$

The calculation formulas of $la_0(t)$, $la_1(t)$ and $la_2(t)$ are as follows:

$$la_0(t) = \frac{(t - t_i)(t - t_j)}{(t_{i-1} - t_i)(t_{i-1} - t_j)} y_{i-1} \tag{11}$$

$$la_1(t) = \frac{(t - t_{i-1})(t - t_j)}{(t_i - t_{i-1})(t_i - t_j)} y_i \tag{12}$$

$$la_2(t) = \frac{(t - t_{i\text{-}1})(t - t_i)}{(t_j - t_{i-1})(t_j - t_i)} y_j \tag{13}$$

A similar method is used for the longitude calculation. If $x(t_{i-1}) = x_{i-1}$, $x(t_i) = x_i$, and $x(t_j) = x_j$, the latitude calculation formula of interpolation time *t* is:

$$x(t) = lo_0(t)x_{i-1} + lo_1(t)x_i + lo_2(t)x_j \tag{14}$$

The calculation formulas of $lo_0(t)$, $lo_1(t)$ and $lo_2(t)$ are as follows:

$$lo_0(t) = \frac{(t-t_i)(t-t_j)}{(t_{i-1}-t_i)(t_{i-1}-t_j)} x_{i-1} \tag{15}$$

$$lo_1(t) = \frac{(t-t_{i-1})(t-t_j)}{(t_i-t_{i-1})(t_i-t_j)} x_i \tag{16}$$

$$lo_2(t) = \frac{(t-t_{i-1})(t-t_i)}{(t_j-t_{i-1})(t_j-t_i)} x_j \tag{17}$$

Finally, the plane coordinates are transformed into corresponding latitude and longitude coordinates, and the final interpolation results are obtained. One example is shown in Figure 10.

Among them, the hollow point is the original track point, and the solid point is the track point calculated by interpolation. It can be seen that the algorithm in this paper can obtain interpolation results close to the original trajectory, whether it is straight-line navigation or turning curve navigation.

Figure 11 is an example of the time alignment result based on interpolation computation for crossing, heading and overtaking cases.

1. Time alignment for the crossing situation.
2. Time alignment for heading situation.
3. Time alignment for the overtaking situation.

Figure 10.
Interpolation result based on our algorithm

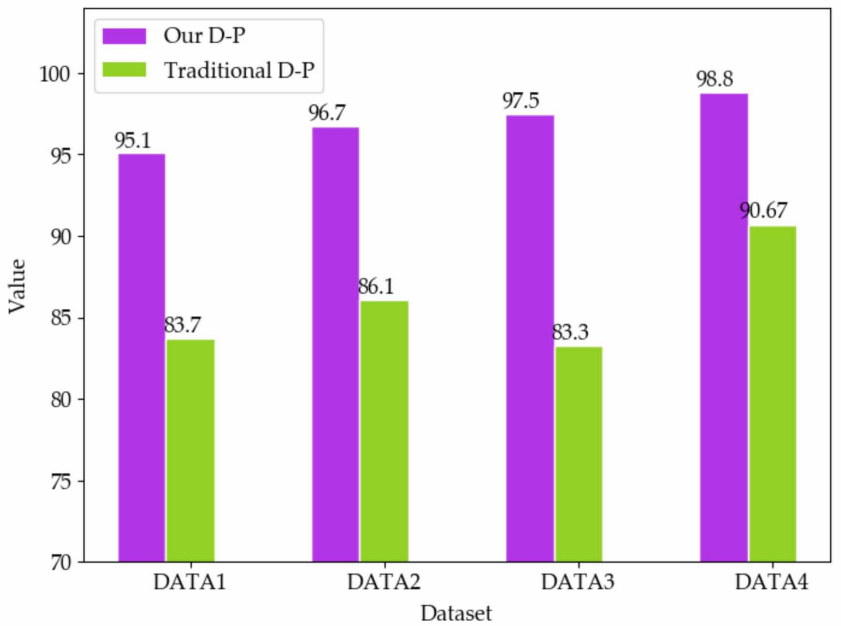
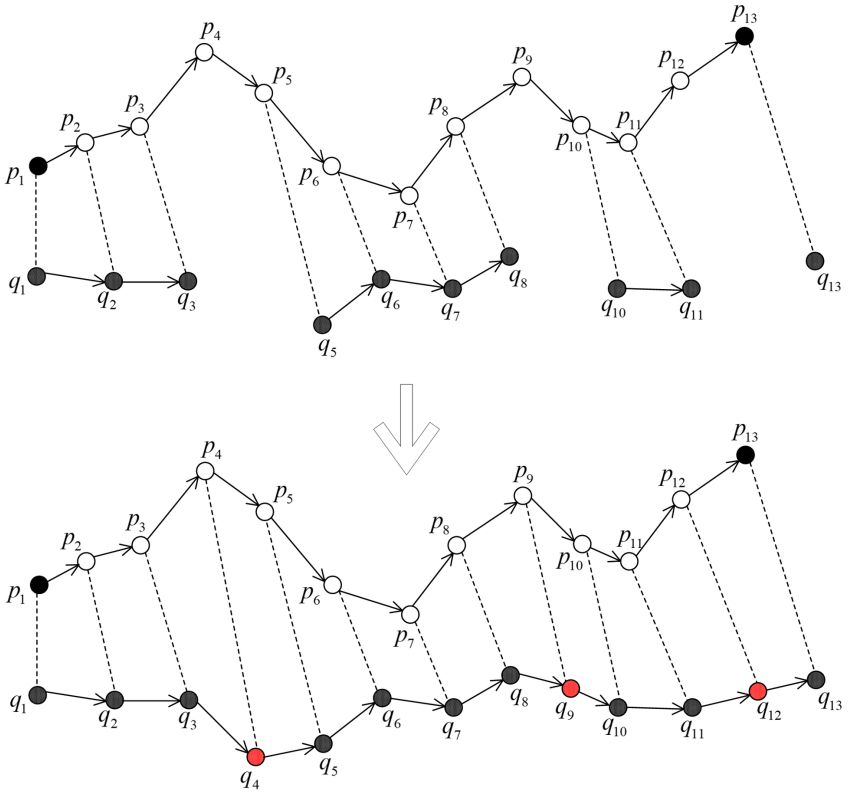


Figure 11.
Time alignment for different meeting situations



VESSEL TRAJECTORY SIMILARITY MEASUREMENT

There are two categories for actual vessel maneuvering patterns under various navigational environments, that is, course keeping and emergency maneuvers. In this paper, vessel maneuvering actions are classified by course keeping and course turning, which is shown in Figure 12.

There are sixteen vessel maneuvering actions defined here, including speed keeping, deceleration, acceleration, weak yaw of left rudder, weak yaw of left rudder with acceleration, weak yaw of left rudder with deceleration, weak yaw of right rudder, weak yaw of right rudder with acceleration, weak yaw of right rudder with deceleration, strong yaw of left rudder, strong yaw of left rudder with acceleration, strong yaw of left rudder with deceleration, strong yaw of right rudder, strong yaw of right rudder with acceleration, strong yaw of right rudder with deceleration and stopping.

The AIS trajectory for vessels can be categorized into different sailing segments based on different maneuvering actions. Each sailing segment corresponds to one or two maneuvering actions. Considering kinematic constraints, the AIS trajectory is classified as a normal sailing segment, acceleration sailing segment, deceleration sailing segment, stopping sailing segment, weak yaw sailing segment, weak yaw sailing segment with acceleration, weak yaw sailing segment with deceleration, strong yaw sailing segment, strong yaw sailing segment with acceleration, and strong yaw sailing segment with deceleration, as shown in Figure 13.

An illustration of sailing segment S_1, S_2, S_3 and S_4 is shown in Figure 14. Here, S_1 is a normal sailing segment. For the whole segment, the linear velocity of the vessel is larger than zero, while linear acceleration and angular acceleration are approximately zero. This sailing segment corresponds

Figure 12.
Classification of vessel maneuvering actions

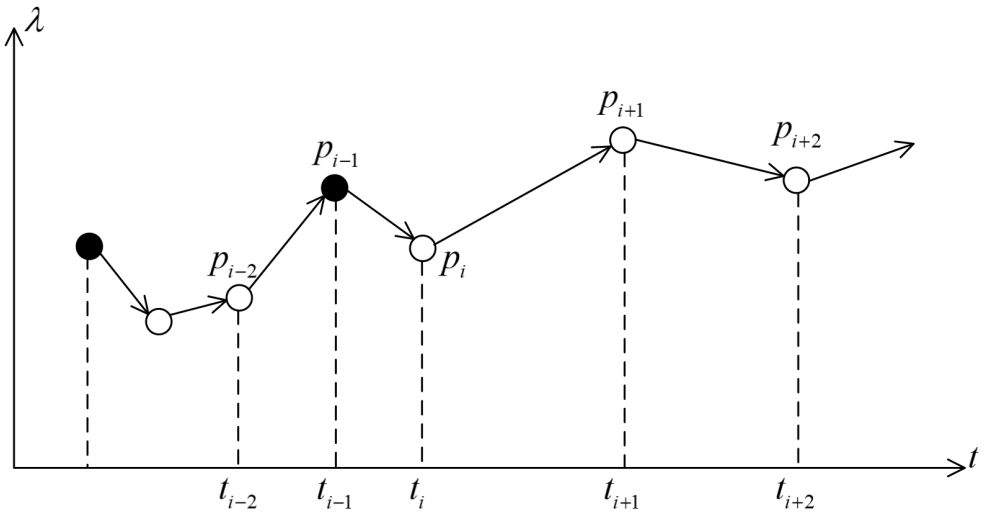
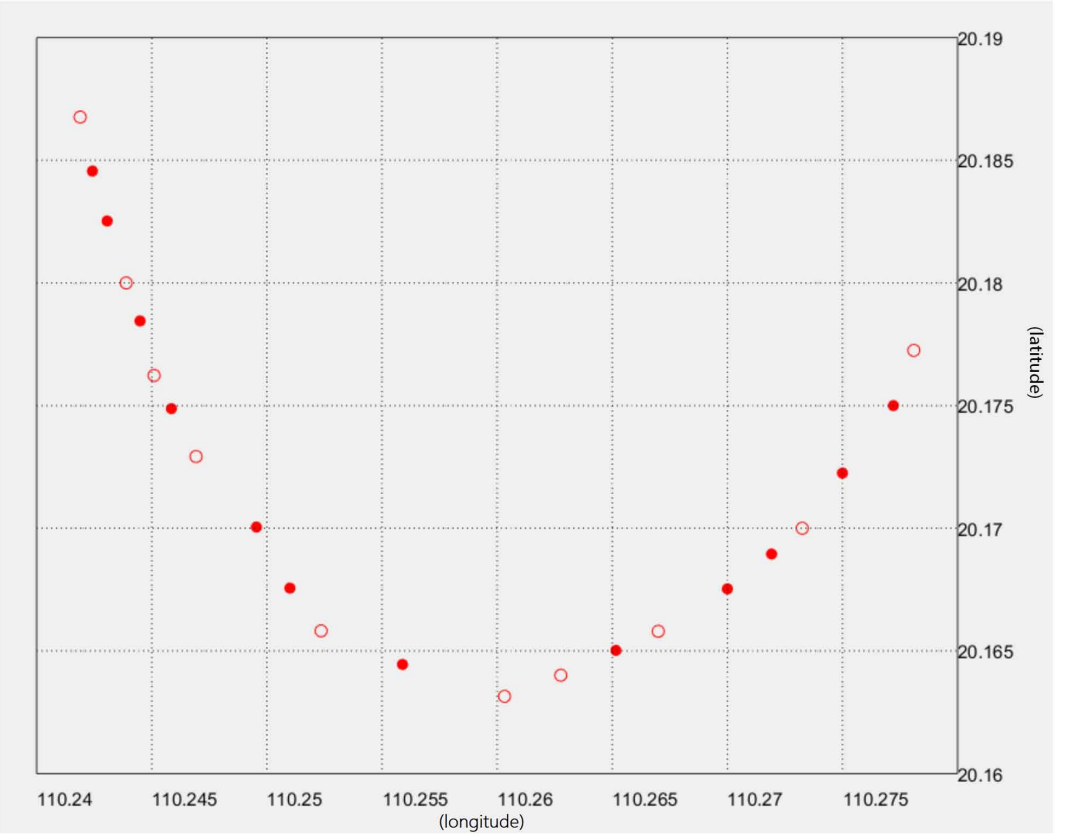


Figure 13.
Classification of vessel sailing segments



to speed-keeping maneuvering action. S_2 is the acceleration sailing segment. For the whole segment, linear acceleration of the vessel is larger than a positive threshold, which corresponds to acceleration maneuvering action. S_3 is the deceleration sailing segment. For the whole segment, linear acceleration of the vessel is lower than a negative threshold, which corresponds to deceleration maneuvering action. S_4 is the stopping sailing segment. For the whole segment, the linear velocity is approximately zero, which corresponds to stopping the maneuvering action.

S_5 is the weak yaw sailing segment. For the segment, the angular velocity is approximately one fixed value, and the course difference between the ending point and the starting point is within a certain threshold. If the course difference is positive, it corresponds to a weak yaw of the right rudder maneuvering action. If the course difference is negative, it corresponds to a weak yaw of the left rudder maneuvering action. S_6 is a weak yaw sailing segment with acceleration. It satisfies the sailing conditions of S_2 and S_5 and corresponds to weak yaw of the left or right rudder with acceleration maneuvering action. An illustration of the sailing segment S_5 and S_6 is shown in Figure 15.

S_7 is a weak yaw sailing segment with deceleration. It satisfies the sailing conditions of S_3 and S_5 and corresponds to the weak yaw of the left or right rudder with deceleration maneuvering action. An illustration of the sailing segment S_7 is shown in Figure 16.

Figure 14.
Illustration of sailing segment S_1, S_2, S_3 and S_4

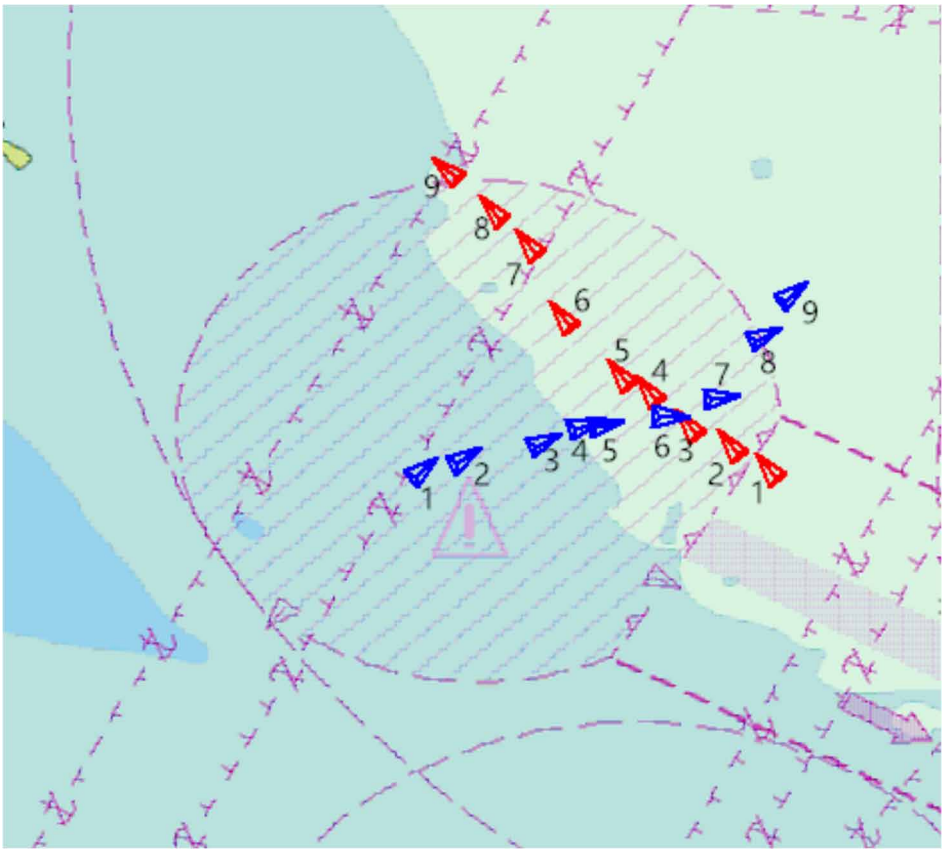


Figure 15.

Illustration of sailing segment S_5 and S_6

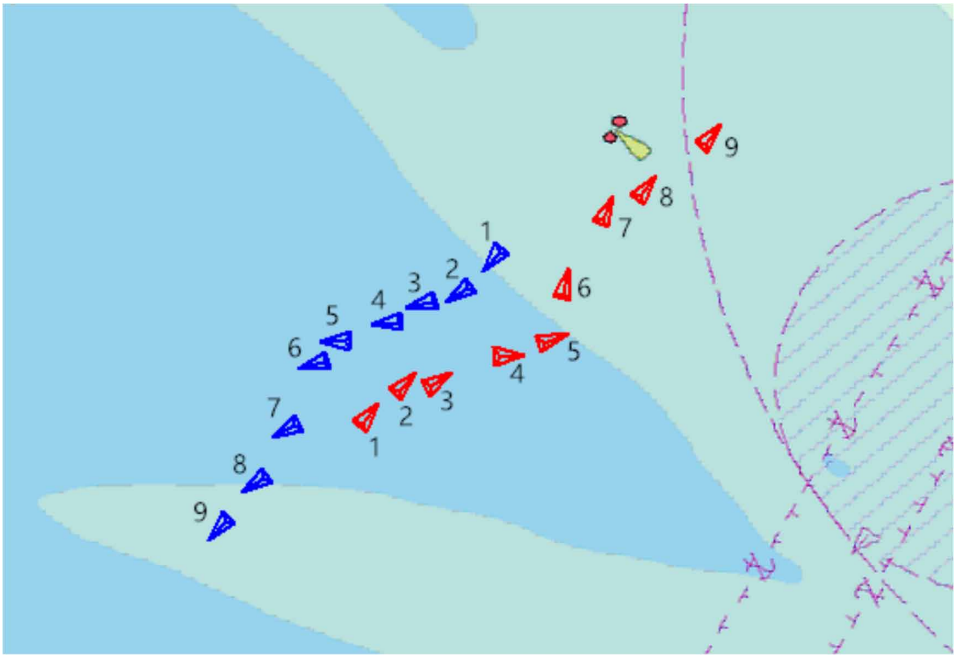
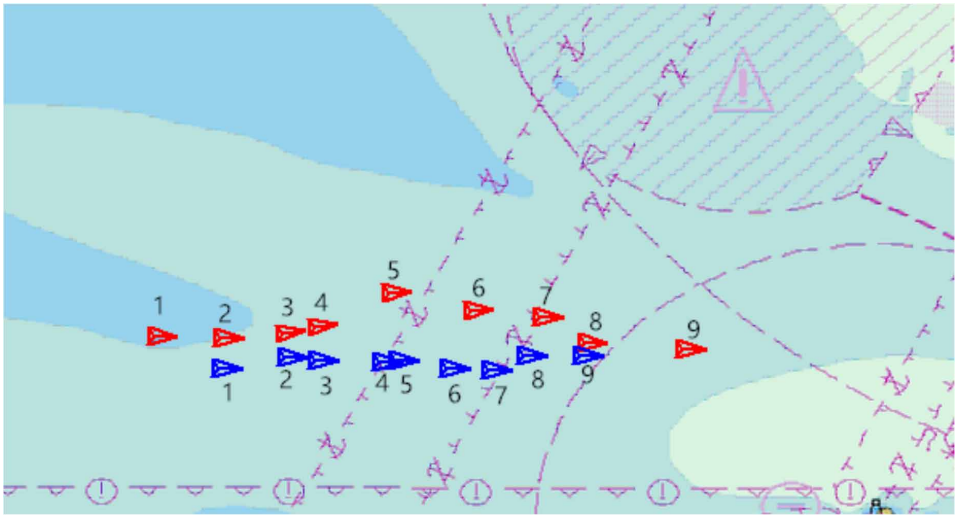


Figure 16.

Illustration of sailing segment S_7



S_8 is a strong yaw sailing segment. For the segment, angular velocity is around a certain fixed value, and the course difference between the ending point and starting point is larger than the threshold of weak yaw but lower than a higher certain threshold. If the course difference is positive, it belongs to the right strong yaw sailing segment, corresponding to the strong yaw of the right rudder maneuvering

action. If the course difference is negative, it belongs to the left strong yaw sailing segment, corresponding to strong yaw of the left rudder maneuvering action. S_8 is shown in Figure 17.

S_9 is a strong yaw sailing segment with acceleration. It satisfies the sailing conditions of S_2, S_8, and corresponds to strong yaw of the left or right rudder with acceleration maneuvering action.

S_{10} is a strong yaw sailing segment with deceleration. It satisfies the sailing conditions of S_3, S_8, and corresponds to strong yaw of the left or right rudder with deceleration maneuvering action. An illustration of the sailing segment S_9 and S_{10} is shown in Figure 18.

S_{11} is the other sailing segment. It is difficult to categorize the sailing segment into S_1 to S_{10} segments. This sailing segment is also called an irregular sailing segment.

Vessel sailing segment computation is based on vessel maneuvering actions. Vessel maneuvering action reflects the motion process of the vessel. Every three points in the trajectory imply one motion process. That is, every three trajectory points implies one vessel maneuvering action event. Vessel maneuvering action can be obtained based on three trajectory points of sampling data at a uniform time interval. Figure 19 shows an identification illustration for vessel maneuvering action.

Figure 17.

Illustration of sailing segment S_8

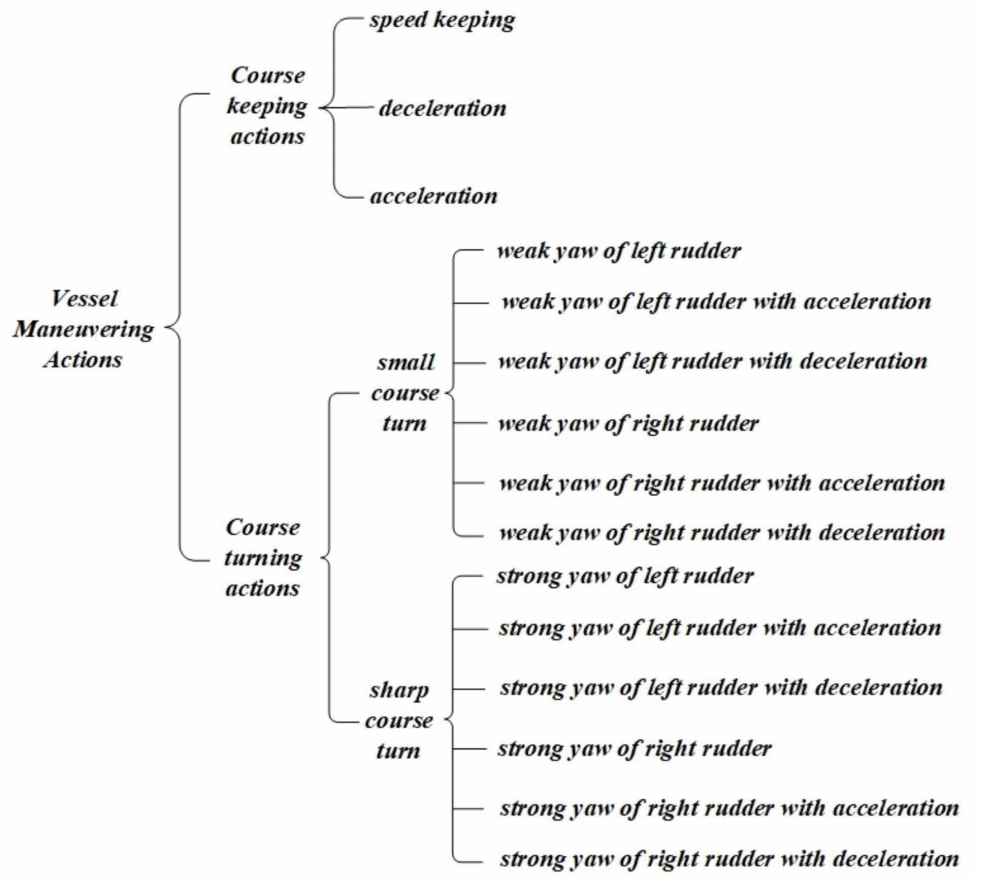
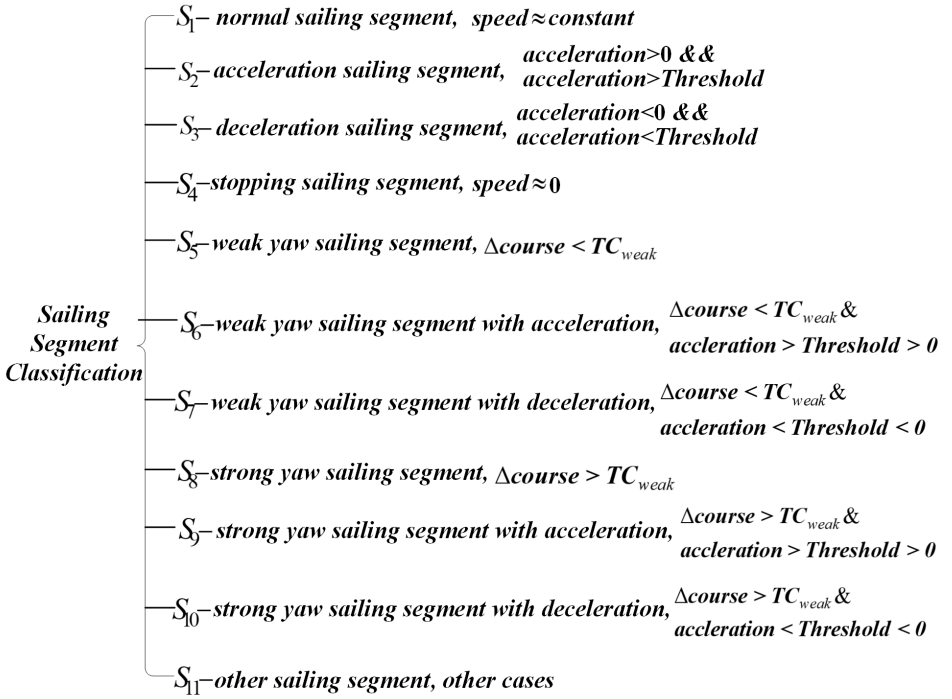


Figure 18.
Illustration of sailing segment S_9 and S_{10}



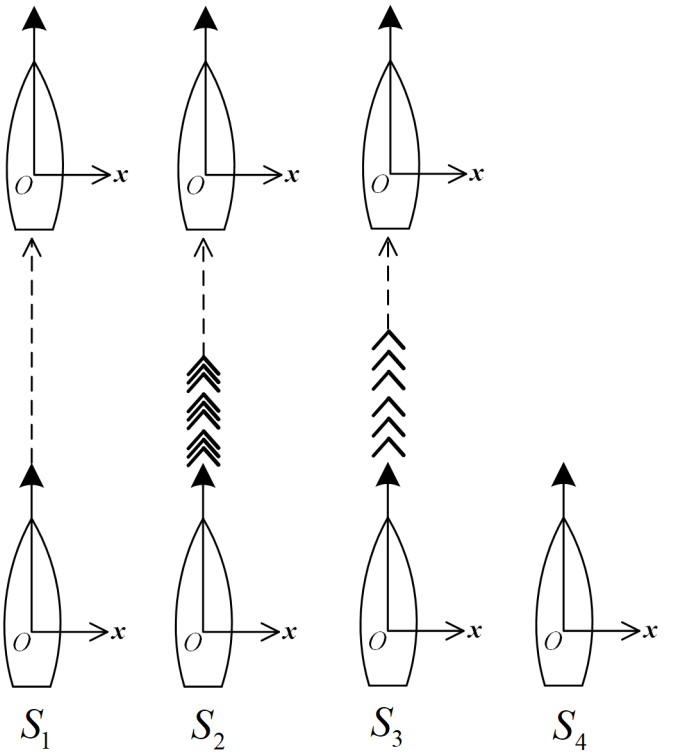
The time interval between p_i and p_{i+1}, p_{i-1} and p_i is Δt. $\overrightarrow{v_i^{in}}$ is the velocity vector of trajectory $p_{i-1}p_i$. $\overrightarrow{v_i^{out}}$ is the velocity vector of trajectory p_ip_{i+1}. $\overrightarrow{v_i^{in}}$ and $\overrightarrow{v_i^{out}}$ can be computed as Eq. (18):

$$\overrightarrow{v_i^{in}} = \frac{p_{i-1}p_i}{\Delta t},\ \overrightarrow{v_i^{out}} = \frac{p_ip_{i+1}}{\Delta t} \tag{18}$$

In the field of nautical science, velocity vectors can be represented with speed over ground (*SOG*) and course over ground (*COG*). That is, $\vec{v} = (V_{VOG}, C_{COG})$, which is shown in the right part of Figure 19.

To distinguish maneuvering action patterns, there are four thresholds, which are RC_{weak} and $RC_{sharp}, VT_{zero}, RV_{acceleration}$. Here, VT_{zero} is a small value around zero. When a vessel is anchored, there is still a small *SOG* influenced by wind or flow. Therefore, VT_{zero} is defined to determine the vessel status. If the vessel speed is lower than VT_{zero}, the vessel can be considered stopping. $RV_{acceleration}$ is vessel acceleration. When a vessel is in sailing status, there will be a slight change in its speed because of the influence of wind and flow. When the acceleration of a vessel is larger than $RV_{acceleration}$, the vessel can be considered to be in a speed-varying maneuvering state. RC_{weak}, RC_{sharp} are the rates of course change. The course change rate is the course change per second. When heading in the desired direction, the trajectory for a vessel will follow an S-shaped course because of the function of wind, current and waves. When there is a weak yaw for a vessel, it will turn to the desired

Figure 19.
Identification for vessel maneuvering action from trajectory data



direction slightly. When the rate of course change for a vessel is lower than RC_{weak}, the vessel can be considered as adopting S-shape swinging, approximately as linear motion. When the rate of course change for a vessel is between RC_{weak} and RC_{sharp}, the vessel can be considered to adopt small course turn motion. If the rate of course change is larger than RC_{sharp}, the vessel can be considered to adopt sharp turn motion. The four thresholds RC_{weak} and $RC_{sharp}, RV_{acceleration}$ are determined according to the speed and course statistics of the current segment and the derivation process is as follows. In our experiment, the values are also inferred by referring to navigation experience or navigation big data.

The rate of vessel speed change and rate of vessel course change are denoted as RV, RC. To compare maneuvering patterns between p_i and its adjacent up and down sailing segments, *RC* and *RV* are defined as Eq. (19):

$$RV = \frac{\Delta v}{\Delta t} = \frac{V_{SOG}^{i,out} - V_{SOG}^{i,in}}{\Delta t} \quad RC = \frac{\Delta C_{COG}}{\Delta t} = \frac{C_{COG}^{i,out} - C_{COG}^{i,in}}{\Delta t} \tag{19}$$

For the whole sailing segment, the global rate of course changes can be computed as Eq. (20):

$$\overline{GC} = \frac{\sum_{i=1}^{n} C_{COG}^{i}}{n}, \overline{RC} = \frac{\overline{GC}}{t_{n} - t_{1}} \tag{20}$$

Therefore, the threshold can be chosen as Eq. (21):

$$\frac{1}{L}\overline{RC} < RC_{weak} < \frac{1}{H}\overline{RC}\ ,\ \frac{1}{M}\overline{RC} < RC_{sharp} < \frac{1}{N}\overline{RC} \tag{21}$$

For the whole sailing segment, the global rate of speed change can be computed as Eq. (22):

$$\overline{GV} = \frac{\sum_{i=1}^{n} V_{VOG}^{i}}{n}, \overline{RV} = \frac{\overline{GV}}{t_{n} - t_{1}} \tag{22}$$

Therefore, the threshold can be chosen as Eq. (23):

$$\frac{1}{P}\overline{RV} < RV_{acceleration} < \frac{1}{Q}\overline{RV} \tag{23}$$

The identification rule for the sailing segment classification is as follows. Course change can be identified by Eq (24):

$$RC\begin{cases} > 0 & \begin{cases} \left|RC\right| > RC_{sharp} & ,strong\ yaw\ of\ right\ rudder \\ RC_{weak} < \left|RC\right| < RC_{sharp} & ,weak\ yaw\ of\ right\ rudder \end{cases} \\ < 0 & \begin{cases} \left|RC\right| > RC_{sharp} & ,strong\ yaw\ of\ left\ rudder \\ RC_{weak} < \left|RC\right| < RC_{sharp} & ,weak\ yaw\ of\ left\ rudder \end{cases} \end{cases} \tag{24}$$

Acceleration or deceleration can be identified by Eq (25):

$$RV\begin{cases}>0 & \begin{cases}\left|RV\right|>RV_{acceleration} & ,acceleration\\ \left|RV\right|<RV_{acceleration}\ \&\&\left|RC\right|<RC_{weak} & \\ & ,course\ \&\ speed\ keeping\\ \&\&\ V_{SOG}^{i,in}>VT_{zero}\ \&\&V_{SOG}^{i,out}>VT_{zero} & \end{cases}\\ <0 & \begin{cases}\left|RV\right|>RV_{acceleration} & ,deceleration\\ \left|RV\right|<RV_{acceleration}\ \&\&\left|RC\right|<RC_{weak} & \\ & ,course\ \&\ speed\ keeping\\ \&\&\ V_{SOG}^{i,in}>VT_{zero}\ \&\&V_{SOG}^{i,out}>VT_{zero} & \end{cases}\end{cases} \tag{25}$$

The right turn with acceleration or deceleration can be identified by Eq. (26):

$$RC>0\begin{cases}RV>0\ \&\&\ \left|RV\right|>RV_{acceleration} & \begin{cases}\left|RC\right|>RC_{sharp},strong\ yaw\ of\\ \quad right\ rudder\ with\ acceleration\\ RC_{weak}<\left|RC\right|<RC_{sharp},weak\ yaw\ of\\ \quad right\ rudder\ with\ acceleration\end{cases}\\ RV<0\ \&\&\ \left|RV\right|>RV_{acceleration} & \begin{cases}\left|RC\right|>RC_{sharp},strong\ yaw\ of\\ \quad right\ rudder\ with\ deceleration\\ RC_{weak}<\left|RC\right|<RC_{sharp},weak\ yaw\ of\\ \quad right\ rudder\ with\ deceleration\end{cases}\end{cases} \tag{26}$$

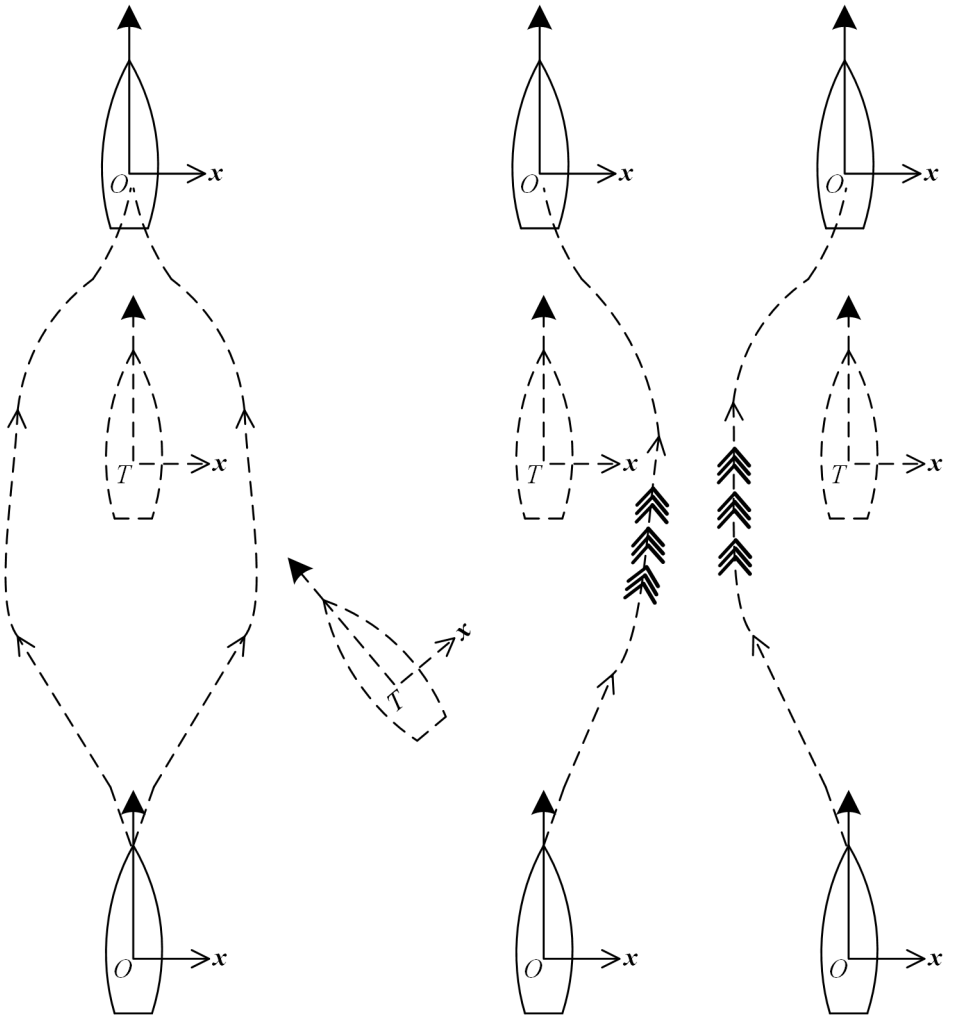
The left turn with acceleration or deceleration can be identified by Eq (27):

$$RC<0\begin{cases}RV>0\ \&\&\ \left|RV\right|>RV_{acceleration} & \begin{cases}\left|RC\right|>RC_{sharp},strong\ yaw\ of\\ \quad left\ rudder\ with\ acceleration\\ RC_{weak}<\left|RC\right|<RC_{sharp},weak\ yaw\\ of\ left\ rudder\ with\ acceleration\end{cases}\\ RV<0\ \&\&\ \left|RV\right|>RV_{acceleration} & \begin{cases}\left|RC\right|>RC_{sharp},strong\ yaw\ of\\ \quad left\ rudder\ with\ deceleration\\ RC_{weak}<\left|RC\right|<RC_{sharp},weak\ yaw\\ of\ left\ rudder\ with\ deceleration\end{cases}\end{cases} \tag{27}$$

Stopping can be identified if $V_{SOG}^{i,in}<VT_{zero}\ \&\&V_{SOG}^{i,out}<VT_{zero}$.

For each vessel in one sailing period, its speed change curve and course change curve can be obtained from its AIS trajectory data. For example, there are three vessels in Figure 20.

Figure 20.
Three vessels AIS trajectory data from August 6 to August 12, 2018 in one area

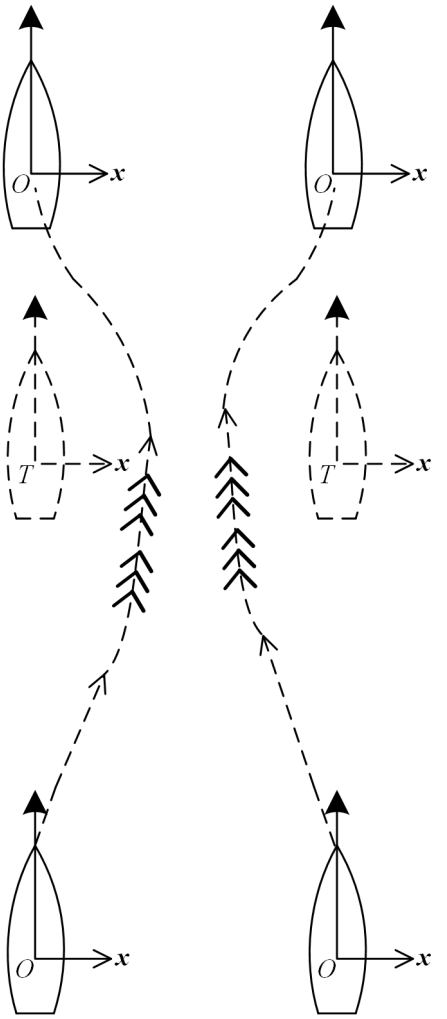


For the vessel of the red trajectory with 413436820 as its mmsi identifier, its speed change curve and course change curve are shown in Figure 21. The curve can be considered as a reference to identify the sharp course change time point and sharp speed change time point. The average course and average speed for the whole sailing period can also be obtained to determine the threshold for vessel maneuvering action identification.

Taking the Qiongzhou straits as an example, AIS data from 785 vessels were collected from July 20, 2018, to August 20, 2018. Based on preprocessing data mentioned above, vessel maneuvering actions such as speed keeping, deceleration, acceleration, weak yaw of left rudder, weak yaw of left rudder with acceleration, weak yaw of left rudder with deceleration, weak yaw of right rudder, weak yaw of right rudder with acceleration, weak yaw of right rudder with deceleration, strong yaw of left rudder, strong yaw of left rudder with acceleration, strong yaw of left rudder with deceleration, strong yaw of right rudder, strong yaw of right rudder with acceleration, strong yaw of right rudder with deceleration are identified.

The sampling time interval is 60 seconds, and

Figure 21.
Speed change curve and course change curve for vessels with MMSI=413436820 from August 6 to August 12, 2018



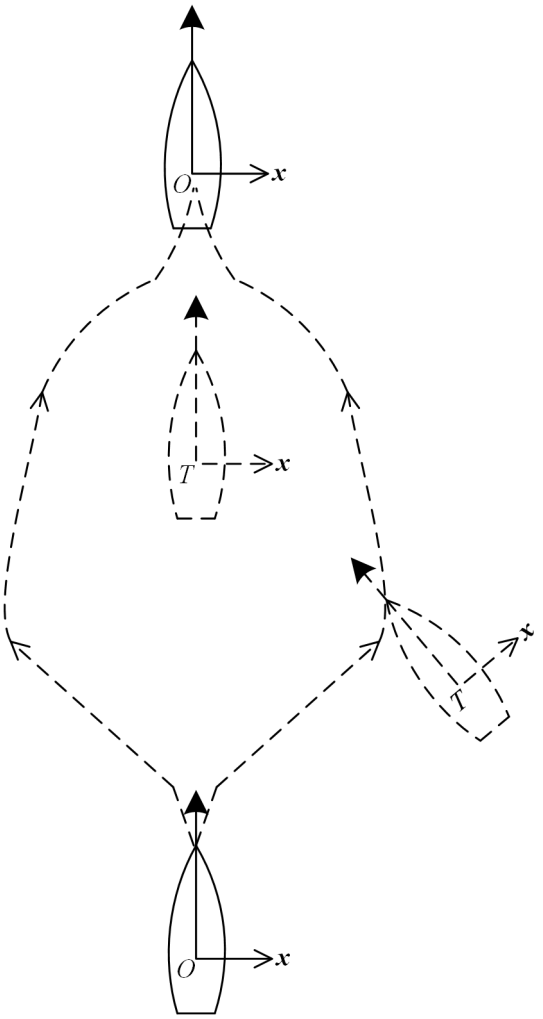
$$RC_{weak} = 2, RC_{sharp} = 10, VT_{zero} = 0.4, RV_{acceleration} = 1.2$$

are set to obtain the vessel trajectory with maneuvering actions. Two trajectories for two vessels are shown in Figure 22 and Figure 23. By analyzing trajectories, major changes in vessel movement can be detected, and critical points can be marked, such as a stop, a smooth turn, a sharp turn or slow motion.

In Figure 23, Point A refers to the weak yaw of the left rudder, point B refers to a stopping point, point C refers to the strong yaw of the left rudder with acceleration, point D refers to acceleration, and point E refers to deceleration.

With critical points as a reference, certain situations can be obtained, such as crossing situations or overtaking situations, which are very important for the analysis of collision avoidance.

Figure 22.
Two-vessel maneuvering action identification results in the Qiongzhou strait



TRAJECTORY SIMILARITY MAPPING BASED ON CNN FROM AIS TRAJECTORY DATA

The overall process can be divided into data preprocessing, candidate encountering vessel searching, and CNN-based trajectory similarity matching. As shown in Figure 24, data preprocessing starts from raw AIS data, and three threads of parsing AIS data, noise reduction and dynamic Ramer–Douglas–Peucker compression are called to create an AIS database. The aggregation of points for each vessel is retrieved from the database by mmsi and timestamp. Then, the trajectory can be represented as a set of line segments. The trajectory is also marked with different sailing segments based on feature extraction from every three points for the trajectory and identification rule for recognizing maneuvering actions.

For any vessel in the database, candidate encountering vessels will be searched based on the algorithm shown in Figure 25. For a given vessel, the searching algorithm will search the vessels near the same area in the same time period within the specified time period and location range. If the time points of AIS trajectory data of two vessels are different, an interpolation algorithm should be

Figure 23.
Zoomed-in result for Figure 23 in one small range to see maneuvering actions identification details

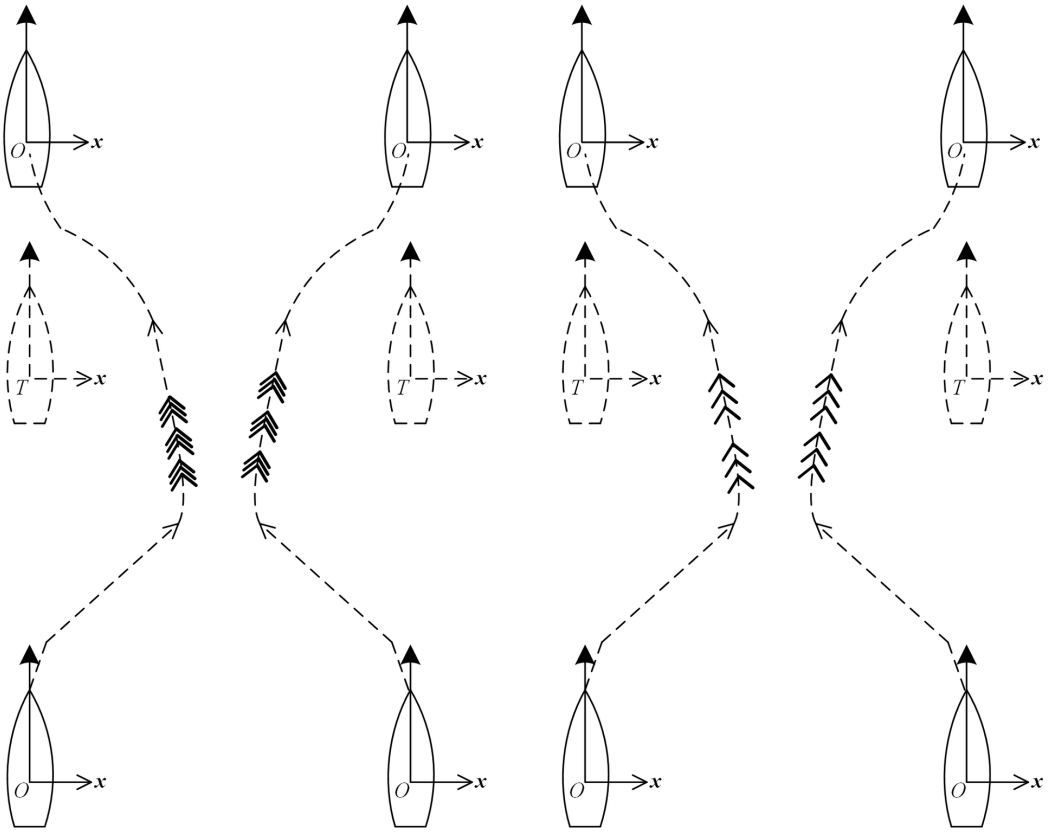
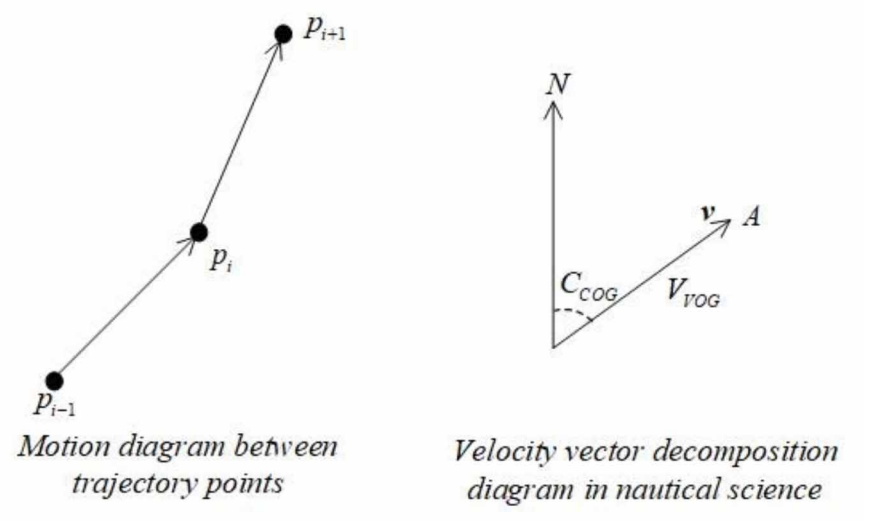


Figure 24.
Flowchart of data preprocessing



used for time alignment. The relative motion parameters, including heading difference and velocity difference, are calculated for the two vessel trajectories after time alignment. According to basic collision avoidance rules, the vessel encounter samples are extracted.

The CNN-based trajectory similarity matching algorithm is shown in Figure 26. The first part of the algorithm adds encountering labels for each data pair.

The second part matches the new input vessel pair with the known encountering AIS database to obtain the encountering situation result, which is shown in Figure 27.

Each input sample in the CNN consists of AIS trajectory data of two vessels according to time, and it is transformed into data of equal time intervals by interpolation processing and other operations. The features of each input sample are a 15-dimensional feature vector, including the time interval, vessel 1 longitude, vessel 1 latitude, vessel 1 speed, vessel 1 course, vessel 1 speed change rate, vessel 1 course change rate, vessel 1 sailing segment mark, vessel 2 longitude, vessel 2 latitude, vessel 2 speed, vessel 2 course, vessel 2 speed change rate, vessel 2 course change rate, vessel 2 course change rate, and vessel 2 segment label. The vector at time i is marked as Eq. (28):

$$\boldsymbol{x}_i^j(t_i,\varphi_i^1,\lambda_i^1,SOG_i^1,COG_i^1,RV_i^1,RC_i^1,S_i^1,\varphi_i^2,\lambda_i^2,SOG_i^2,COG_i^2,RV_i^2,RC_i^2,S_i^2) \tag{28}$$

There are three filters used in the convolution computing process, which are listed as Eq. (29), Eq. (30) and Eq. (31):

Figure 25.
Flowchart of candidate encountering vessel pairs searching

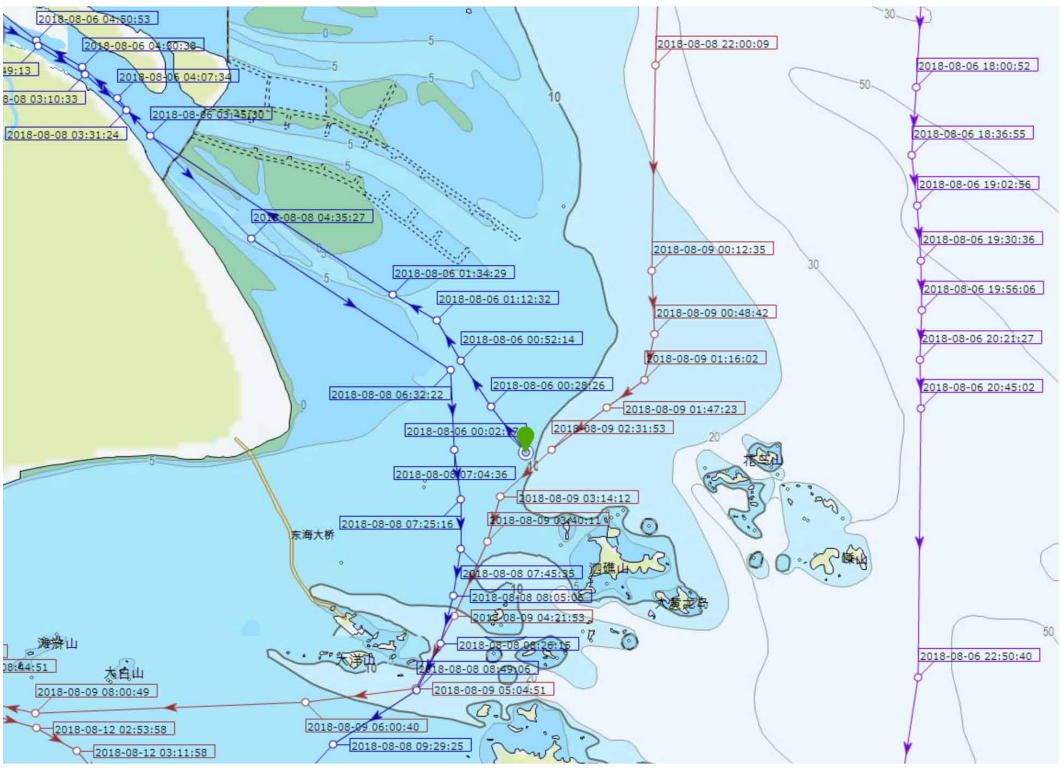
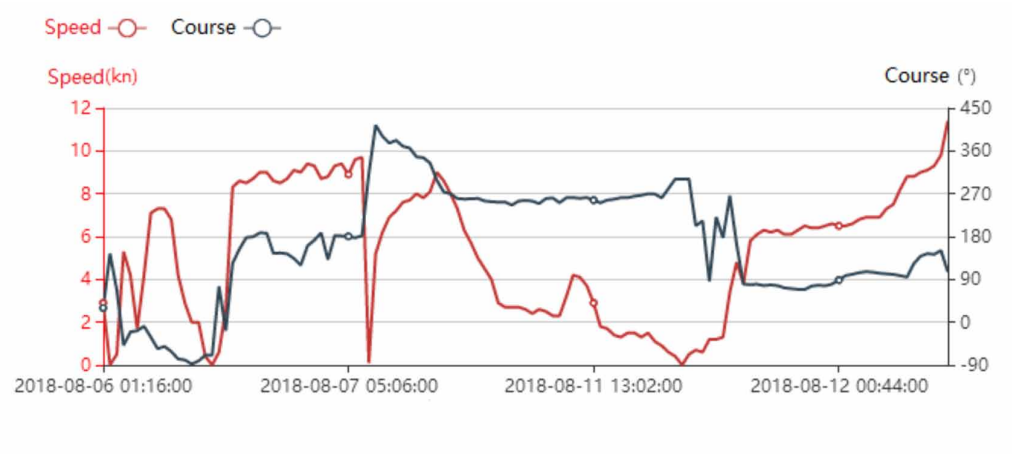


Figure 26.
Flowchart of the CNN-based trajectory similarity matching algorithm



$$G_{4*15} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ -1 & -1 & -1 & -1 & \cdots & -1 & -1 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ -1 & -1 & -1 & -1 & \cdots & -1 & -1 \end{bmatrix}}_{15} \tag{29}$$

$$G_{3*15} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & -1 & -1 & \cdots & -1 & -1 & -1 \end{bmatrix}}_{15} \tag{30}$$

$$G_{2*15} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\ -1 & -1 & -1 & \cdots & -1 & -1 & -1 \end{bmatrix}}_{15} \tag{31}$$

The convolution process is illustrated in Figure 28. Taking a filter of size 2 as an example, it adds the product with the first two lines of the input matrix, multiplies the corresponding elements, and then calculates the cumulative sum to obtain the first value of the result vector. Then, it moves the filter down one position to obtain the second value of the result vector, and the final feature map size is (15-2 + 1x1) = 14. Because the size of the feature map obtained by different size filters is different, a layer of max pooling is added to solve this problem, selecting a maximum value and combining the same size together. After the max pooling operation, the fixed length vector is given to sofamax to obtain the final matching degree. The encountering situation with the maximum matching degree is labeled for the new input vessel AIS trajectory pair. The loss function adopted is the conventional softmax cross entropy. Cross entropy describes the distance between the actual output (probability) and the expected output (probability). In other words, the smaller the cross entropy value is, the closer the two probability distributions are. It is defined as Eq. (32):

$$L_i = -\sum_{k=1}^{K} q_i(k) \cdot \log p_i(k), L = \frac{1}{N}\sum_i L_i \tag{32}$$

Figure 27.
Illustration of CNN-based trajectory similarity matching

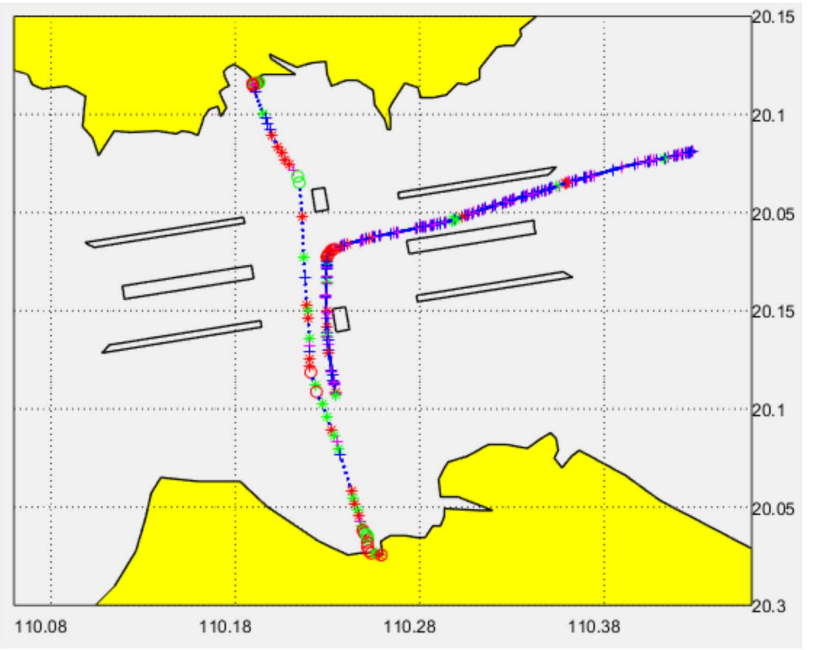
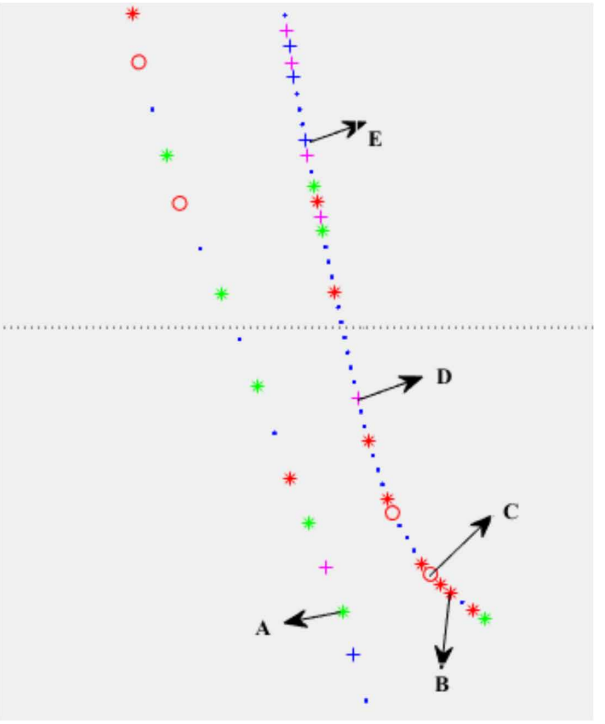


Figure 28.
Illustration of convolution computing in CNN



EXPERIMENTAL RESULTS AND ANALYSIS

Taking the Qiongzhou straits as an example, AIS data from 890 vessels were collected from May 20, 2019, to August 20, 2018. The features of the dataset used in the experiment are listed in Table 1. Configuration information for the hardware and software of our experiment is listed in Table 2. For the training data, a classification decision method combining weighted kNN algorithm and hypersphere support vector machine algorithm in Chen et al. (2018) is used to achieve accurate and efficient vessel encounter classification, extracting the trajectory data of collision avoidance process to build a concise version of the ship collision avoidance knowledge base. All the original supervised data has passed the experts' verification.

Because this is a classification problem, the universal precision, recall and F value are used as the evaluation indicators of the experimental results. The evaluation indicators are Precision, Recall and F1_score, which are represented in Eq. (33), Eq. (34) and Eq. (35):

$$P = \frac{\alpha}{\alpha + \varphi} \tag{33}$$

$$R = \frac{\alpha}{\alpha + \beta} \tag{34}$$

$$F_{\beta} = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R} \quad \beta = 1 \tag{35}$$

Here, α represents the number of test tracks correctly assigned to category c, β represents the number of test tracks incorrectly assigned to category c, and φ represents the number of test tracks incorrectly assigned to category. The comparison result between the LCS model in Chen et al. (2019), HSSVM in Chen et al. (2018) and our method is shown in Figure 29. Clearly, the performance of our

Table 1.
Features of the dataset

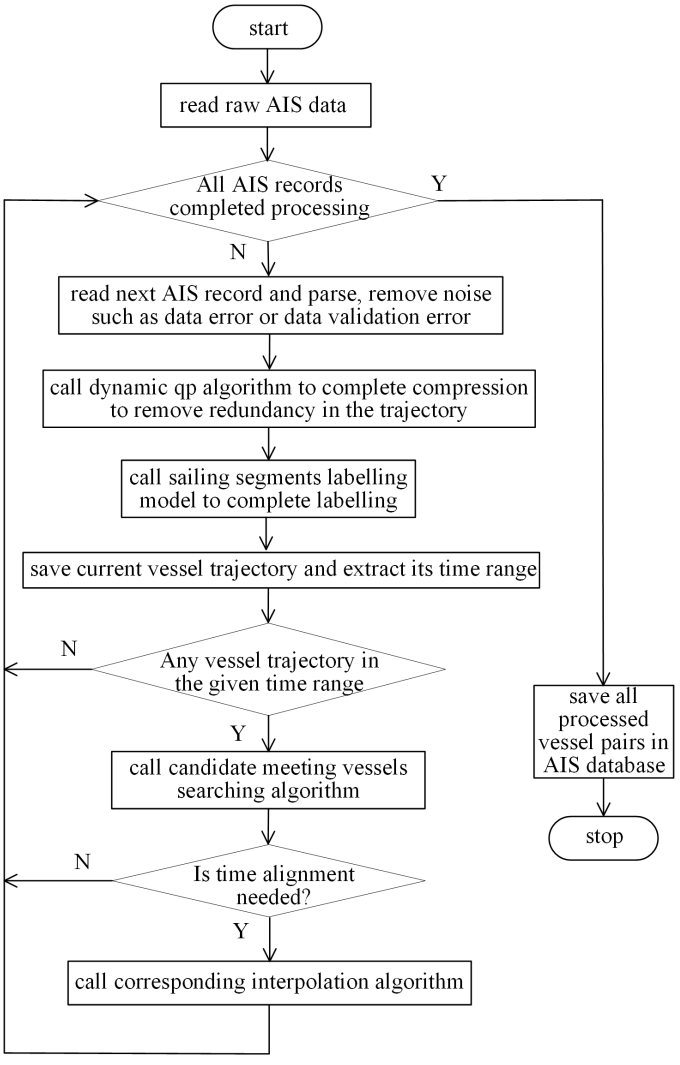
	Training(#)	Testing(#)
Crossing with small angle	8586	3546
Overtaking	7453	2564
Heading on	8125	3210
Vertical crossing	8432	2768
Crossing with big angle	7865	3012
Others	12098	5478

Table 2.
Configuration information for the hardware and software of our experiment

Configuration	Parameters
Operating system	Ubuntu 16.04
GPU	RTX2080Ti (11G)
Python version	3.7
Development tool	Pycharm
Deep learning Kit	Keras

method is better than that of the other two methods, with the highest precision 89.33%, the highest recall 92.1% and the F1_score 90.67%. For the hybrid hypersphere support vector machine proposed in Chen et al. (2018), training time complexity is $O(n^2)$. The time complexity of our CNN training is $O(E*D/B*T)$, where E is epochs, D is dataset size, B is batch Size, and T is the time complexity of a single item. Here, this T can be further decomposed into $O(T)\approx O(L*n)$, where L is the average time complexity of each layer, and n is the number of layers. This L can be further decomposed into $O(L)\approx O(M*N*K*K*H*W)$. Here, it is assumed that the calculation amount of each layer is concentrated in conv2d, M and N are the number of input and output channels respectively, K is the size of convolution kernel, and H and W are the space size of output feature map respectively. Although the time complexity of CNN is higher than that of other methods, it can process large-scale data efficiently, and it is not necessary to consider the time complexity too much when using GPU computing.

Figure 29.
Comparison result of three methods



With our CNN-based similarity matching method, Figure 30 to Figure 34 show different meeting situation matching results. Figure 30 shows one overtaking situation matching result. Figure 31 shows one crossing situation with a small angle matching result, and Figure 32 shows the vertical crossing situation matching result. Figure 33 shows one crossing situation with a large angle matching result. Figure 34 shows the situation matching result.

CONCLUSION

Starting from raw AIS data, one AIS database is created after noise reduction and dynamic Douglas–Peucker compression. Then, candidate encountering vessel pairs are selected based on the candidate

Figure 30.
Overtaking situation matching result based on our method

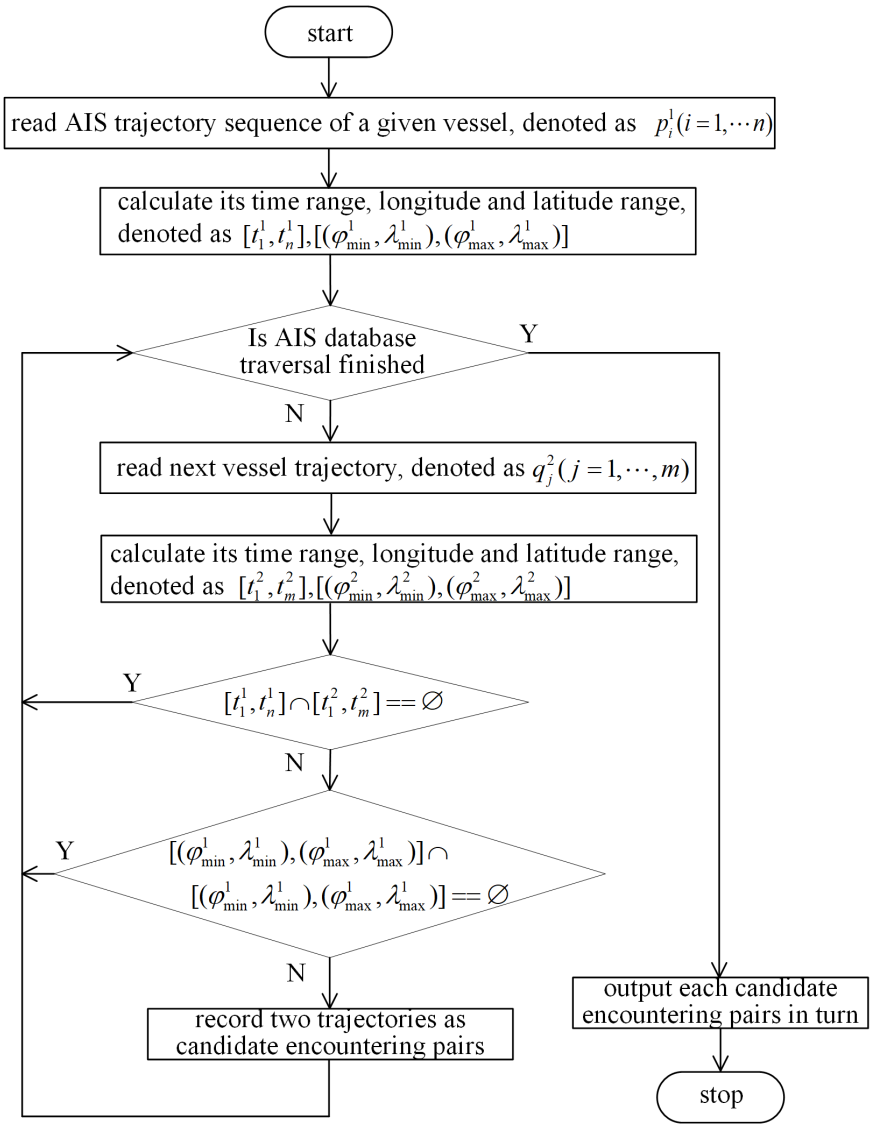
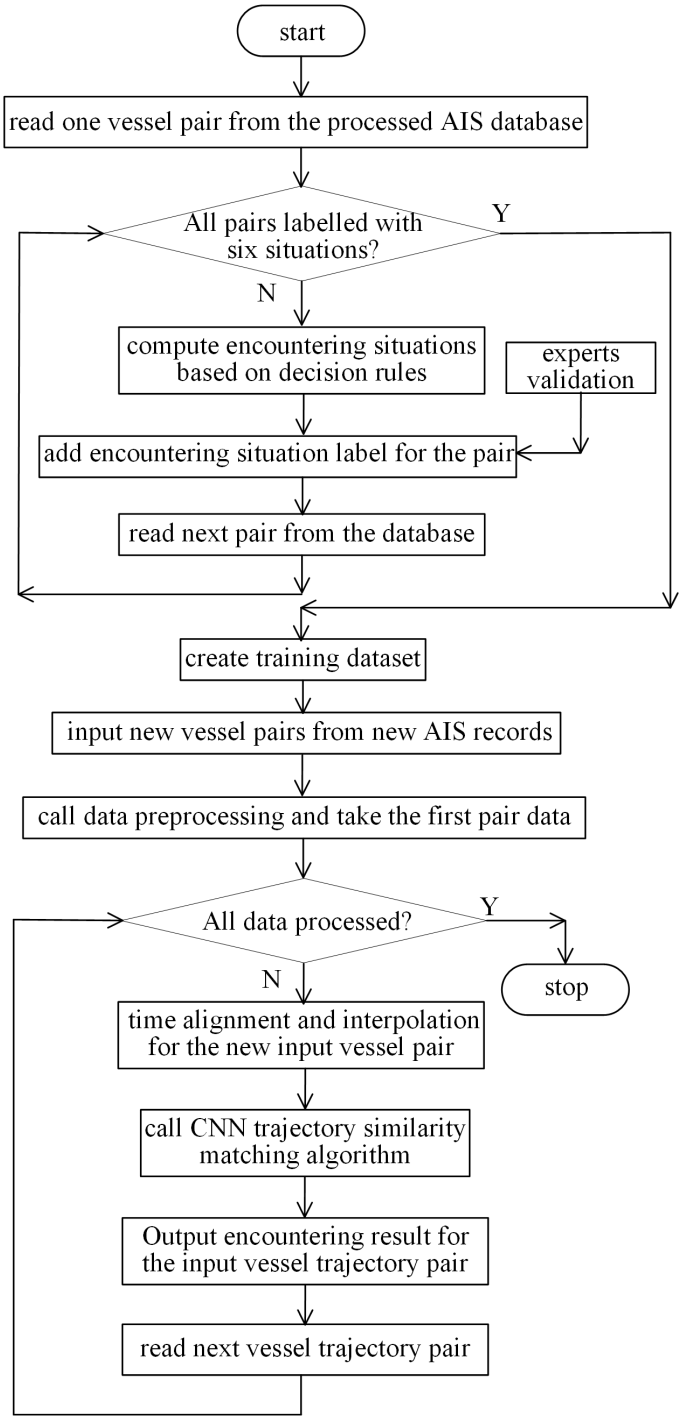
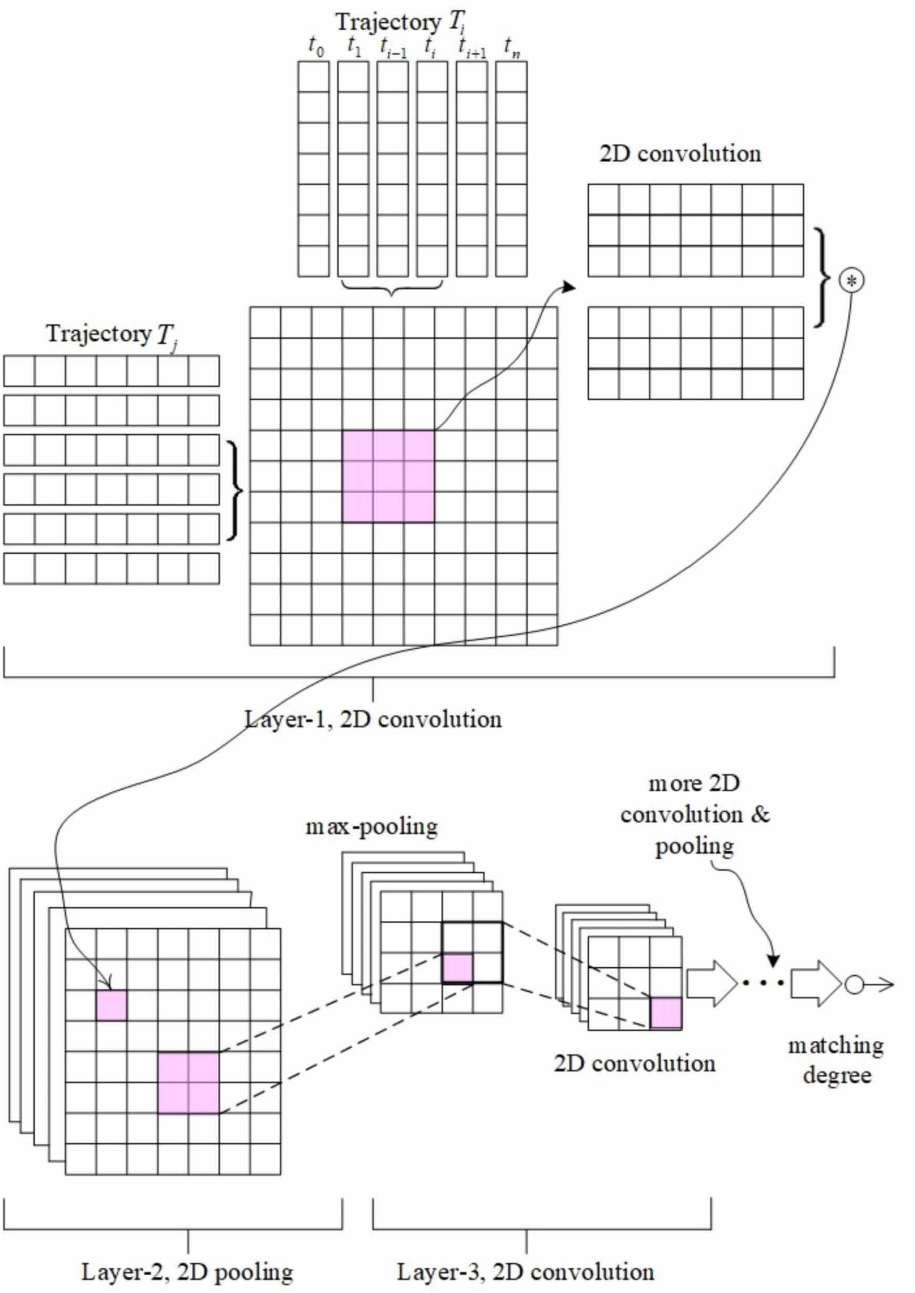


Figure 31.
Crossing situation with a large angle matching result



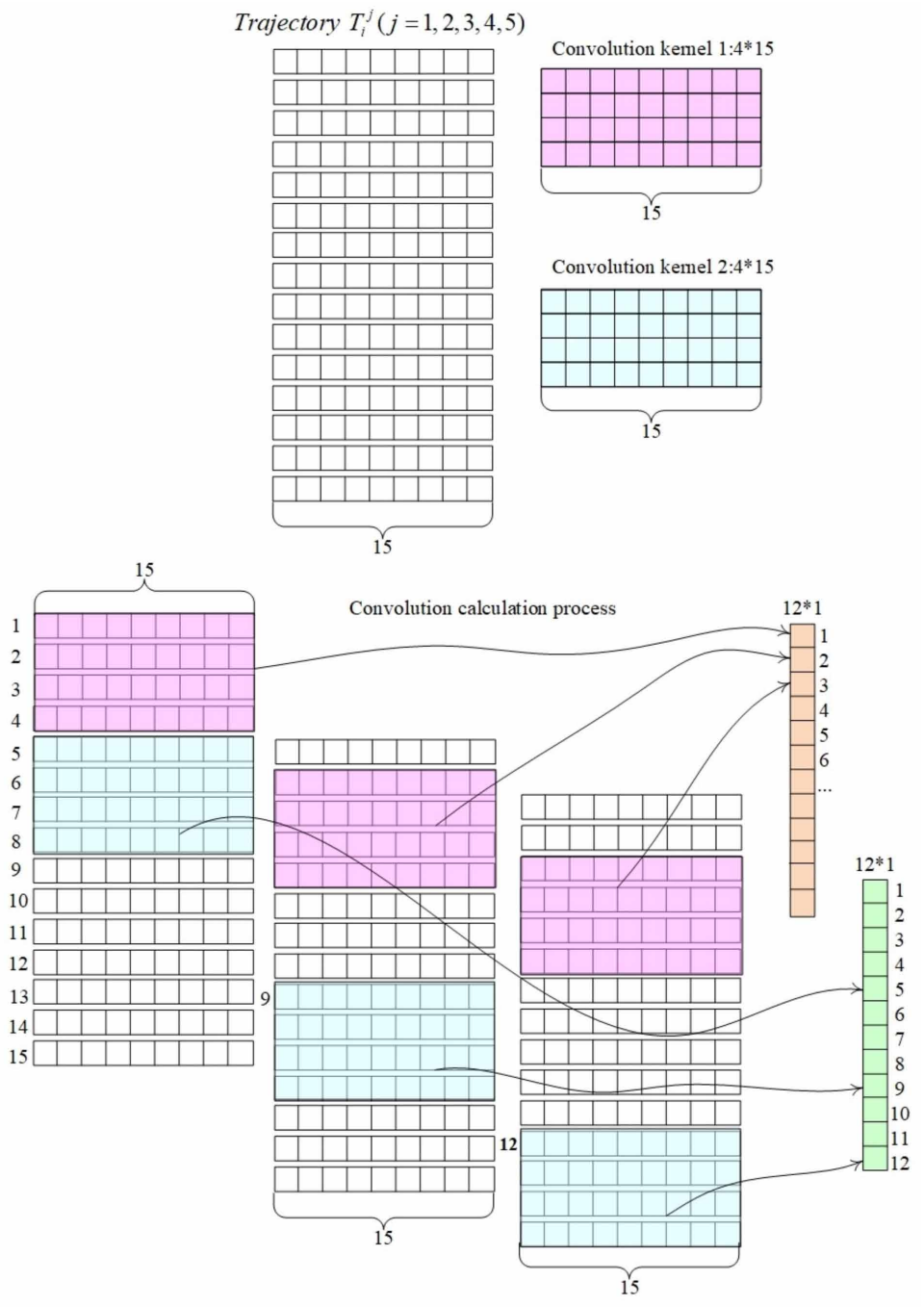
meeting vessel searching algorithm. To prepare for the input of the CNN, time alignment based on an interpolation operation is adopted for each candidate meeting vessel pair. With statistical analysis of

Figure 32.
Vertical crossing situation matching result



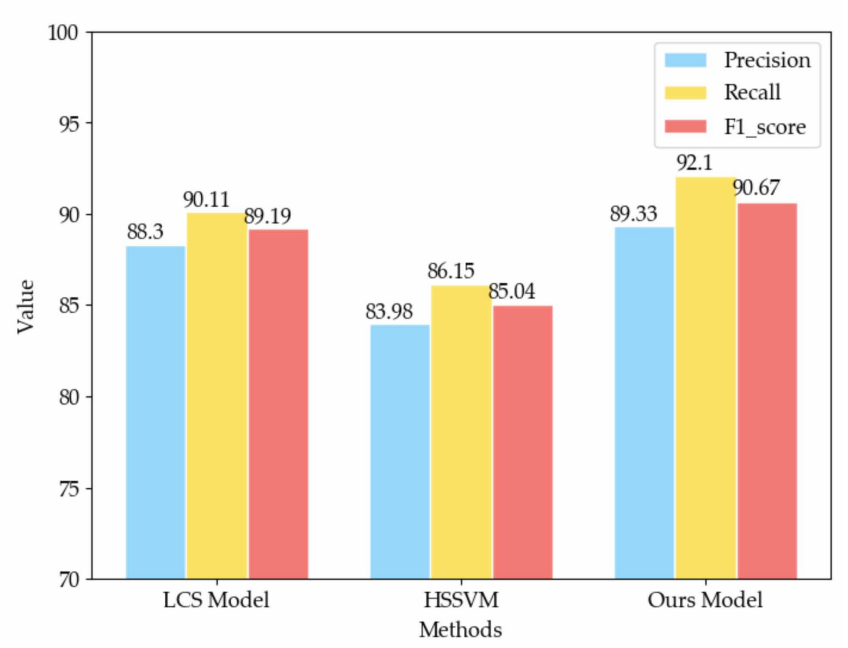
vessel trajectories, critical vessel motions, such as maneuvering actions of a sharp left turn or sharp right turn, can be identified. To ensure the accuracy of the CNN matching result, a sailing segment labeling algorithm is also adopted. For each pair of potential encounter trajectories, the convolution neural network is called to calculate the similarity of the trajectory in the vessel encounter situation knowledge base. Finally, the similarity matching results are obtained after the computation of the

Figure 33.
Crossing situation with small angle matching result



convolution layer, pooling layer, full connection layer and output layer of the CNN. From the compound features defined in the paper, the proposed method will help to discover vessel meeting knowledge from current trajectories and help to discover potential collision risk.

Figure 34.
Heading on situation matching result



AIS historical data processing belongs to massive data processing, and time stamp is one of the important features of ship trajectory. LSTM (Long Short Term Memory) can not only make full use of the time feature information, but also has unparalleled advantages in other methods when learning and predicting massive data. Therefore, our future work will focus on integrating different deep learning methods of CNN and LSTM to complete the unsupervised automatic knowledge discovery process of end-to-end AIS data.

REFERENCES

Burger, C. N., Grobler, T. L., & Kleynhans, W. (2020). Discrete Kalman Filter and Linear Regression Comparison for Vessel Coordinate Prediction. *2020 21st IEEE International Conference on Mobile Data Management (MDM)*. IEEE.

Chen, P., Shi, G. Y., & Liu, S. (2018). Classification Decision based on a Hybrid Method of Weighted *k*NN and Hyper-Sphere SVM. *International Journal of Performability Engineering*, *14*(5), 985–994. doi:10.23940/ijpe.18.05.p17.985994

Chen, P., Shi, G. Y., & Liu, S. (2019). Collision Avoidance Situation Matching with Vessel Maneuvering Actions Identification from Vessel Trajectories. *International Journal of Performability Engineering*, *15*(6), 1499–1507. doi:10.23940/ijpe.19.06.p1.14991507

D'Afflisio, E., Braca, P., & Willett, P. (2021). Malicious AIS Spoofing and Abnormal Stealth Deviations: A Comprehensive Statistical Framework for Maritime Anomaly Detection. *IEEE Transactions on Aerospace and Electronic Systems, PP*(99).

Dwg, A., Ysz, A., Jfz, B., Ykh, B., Ke, Y. A., & Bry, A. (2021). A Novel Mp-LSTM Method for Ship Trajectory Prediction based on AIS Data. *Ocean Engineering*, 228.

Fu, P., Wang, H., Liu, K., Hu, X., & Zhang, H. (2017). Finding Abnormal Vessel Trajectories using Feature Learning. *IEEE Access : Practical Innovations, Open Solutions*, *5*, 7898–7909. doi:10.1109/ACCESS.2017.2698208

Gao, M., & Shi, G. Y. (2020). Ship Collision Avoidance Anthropomorphic Decision-making for Structured Learning based on AIS with Seq-CGAN. *Ocean Engineering*, *217*(13–15), 107922. doi:10.1016/j.oceaneng.2020.107922

Greidanus, H., Alvarez, M., Eriksen, T., & Gammieri, V. (2016). Completeness and Accuracy of a Wide-Area Maritime Situational Picture based on Automatic Ship Reporting Systems. *Journal of Navigation*, *1*(1), 1–13. doi:10.1017/S0373463315000582

Guo, S., Mou, J., Chen, L., & Chen, P. (2021). Improved Kinematic Interpolation for AIS Trajectory Reconstruction. *Ocean Engineering*, *234*(8), 109256. doi:10.1016/j.oceaneng.2021.109256

Iphar, C., Napoli, A., & Ray, C. (2020). An Expert-based Method for the Risk Assessment of Anomalous Maritime Transportation Data. *Applied Ocean Research*, *104*(3), 102337. doi:10.1016/j.apor.2020.102337

Jadidi, M. (2021). Modeling Vessel Behaviours by Clustering AIS Data Using Optimized DBSCAN. *Sustainability*, 13.

Kopyt, P., Salski, B., Zagrajek, P., Janczak, D., Sloma, M., Jakubowska, M., Olszewska-Placha, M., & Gwarek, W. (2016). Electric Properties of Graphene-based Conductive Layers from DC up to Terahertz Range. *IEEE Transactions on Terahertz Science and Technology*, *6*(3), 480–490. doi:10.1109/TTHZ.2016.2544142

Le, Q. I., & Zheng, Z. (2016). Trajectory Prediction of Vessels Based on Data Mining and Machine Learning. *Journal of Digital Information Management*, *14*(1), 33–40.

Lee, H. T., Lee, J. S., Yang, H., & Cho, I. S. (2021). An AIS Data-Driven Approach to Analyze the Pattern of Ship Trajectories in Ports Using the DBSCAN Algorithm. *Applied Sciences (Basel, Switzerland)*, *11*(2), 799. doi:10.3390/app11020799

Lei, P. R. (2016). A Framework for Anomaly Detection in Maritime Trajectory Behavior. *Knowledge and Information Systems*, *47*(1), 189–214. doi:10.1007/s10115-015-0845-4

Li, S., Liang, M., Wu, X., Liu, Z., & Liu, R. W. (2020). AIS-Based Vessel Trajectory Reconstruction with U-Net Convolutional Networks. *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA 2020)* (pp. 157-161). IEEE.

Singh, S. K., & Heymann, F. (2020). Machine Learning-Assisted Anomaly Detection in Maritime Navigation Using AIS Data. *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE.

Sun, L., Zhou, W., Guan, J., & He, Y. (2018). Mining Spatial–temporal Motion Pattern for Vessel Recognition. *International Journal of Distributed Sensor Networks*, *14*(5), 155014771877956. doi:10.1177/1550147718779563

Sun, S., Chen, Y., Piao, Z., & Zhang, J. (2020). Vessel AIS Trajectory Online Compression based on scan-pick-move Algorithm added Sliding Window. *IEEE Access, PP*, *8*(99), 1–1. doi:10.1109/ACCESS.2020.3001934

Tang, C., Wang, H., Zhao, J., Tang, Y., & Xiao, Y. (2021). A Method for Compressing AIS Trajectory Data based on the Adaptive-threshold Douglas-Peucker Algorithm. *Ocean Engineering*, *232*(4), 109041. doi:10.1016/j.oceaneng.2021.109041

Wang, L., Chen, P., Chen, L., & Mou, J. (2021). Ship AIS Trajectory Clustering: An HDBSCAN-Based Approach. *Journal of Marine Science and Engineering*, *9*(6), 566. doi:10.3390/jmse9060566

Wei, Z., Xie, X., & Zhang, X. (2020). AIS Trajectory Simplification Algorithm Considering Ship Behaviours. *Ocean Engineering*, *216*(18), 108086. doi:10.1016/j.oceaneng.2020.108086

Yan, Z., Xiao, Y., Cheng, L., He, R., & Ran, B. (2020). Exploring AIS Data for Intelligent Maritime Routes Extraction. *Applied Ocean Research*, *101*, 102271. doi:10.1016/j.apor.2020.102271

Zhang, R., & Furusho, M. (2020). Constructing an Autonomous Decision Approach in a Real AIS Environment Using Deep Reinforcement Learning. 日本航海学会誌(211), 37-38.

Zhang, S. K., Liu, Z. J., Cai, Y., Wu, Z. L., & Shi, G. Y. (2016). AIS Trajectories Simplification and Threshold Determination. *Journal of Navigation*, *1*(4), 1–16.

Zhen, R., Jin, Y., Hu, Q., Shao, Z., & Nikitas, N. (2017). Maritime Anomaly Detection within Coastal Waters based on Vessel Trajectory Clustering and Naive Bayes Classifier. *Journal of Navigation*, *70*(3), 648–670. doi:10.1017/S0373463316000850

Peng Chen, born in Xuzhou, Jiangsu, China, received his master degree of computer science from Liaoning Shihua University in 2003. Now he works as a professor at the department of software engineering in Dalian Neusoft University of Information. And he got his PhD degree at Dalian Maritime University in 2020. His research interests include network security and machine learning algorithms. His academic papers are published both national and international journals and conferences such as Microelectronics & Computer (2015), ICIC Express Letters, Part B: Applications (2015), Information Technology Journal (2013) and Sensors & Transducers (2015).

Shuang Liu, born in Jinzhou, Liaoning, China, received her PhD degree in traffic information engineering & control from Dalian Maritime University in 2006. She finished her postdoctoral research on computer science and technology at Dalian University of Technology in April 2015. She currently works as an associate professor at the School of Computer Science & Engineering, Dalian Minzu University. Her research interests include computer education, intelligent transportation systems and machine learning algorithms.

Niko Lukač is an assistant professor at Faculty of Electrical Engineering and Computer Science, University of Maribor. His research interests include modelling and simulation, LiDAR, remote sensing, pattern recognition, general-purpose computing on graphics processing units, and scientific visualization. He got his Ph.D in computer science in 2016 from University of Maribor.