IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 25, 2019, accepted October 9, 2019, date of publication October 14, 2019, date of current version October 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947111

Adaptive Douglas-Peucker Algorithm With Automatic Thresholding for AIS-Based Vessel Trajectory Compression

JINGXIAN LIU[1,2], HUANHUAN LI[1,2,3], (Student Member, IEEE), ZAILI YANG[3], KEFENG WU[4], YI LIU[1,2], AND RYAN WEN LIU[1,2], (Member, IEEE)

[1]Hubei Key Laboratory of Inland Shipping Technology, School of Navigation, Wuhan University of Technology, Wuhan 430063, China
[2]National Engineering Research Center for Water Transport Safety, Wuhan 430063, China
[3]Liverpool Logistics, Offshore and Marine Research Institute, Liverpool John Moores University, Liverpool L3 3AF, U.K.
[4]Beijing Electro-Mechanical Engineering Institute, Beijing 100074, China

Corresponding authors: Zaili Yang (z.yang@ljmu.ac.uk) and Ryan Wen Liu (wenliu@whut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51479156, Grant 51809207, Grant 51709219, and Grant 51179147, in part by the China Scholarship Council under Grant 201706950105, in part by the EU Project RESET under Grant H2020-MSCA-RISE-2016-730888, and in part by the EU Project GOLF under Grant H2020-MSCA-RISE-2017-777742.

ABSTRACT Automatic identification system (AIS) is an important part of perfecting terrestrial networks, radar systems and satellite constellations. It has been widely used in vessel traffic service system to improve navigational safety. Following the explosion in vessel AIS data, the issues of data storing, processing, and analysis arise as emerging research topics in recent years. Vessel trajectory compression is used to eliminate the redundant information, preserve the key features, and simplify information for further data mining, thus correspondingly improving data quality and guaranteeing accurate measurement for ensuring navigational safety. It is well known that trajectory compression quality significantly depends on the threshold selection. We propose an Adaptive Douglas-Peucker (ADP) algorithm with automatic thresholding for AIS-based vessel trajectory compression. In particular, the optimal threshold is adaptively calculated using a novel automatic threshold selection method for each trajectory, as an improvement and complement of original Douglas-Peucker (DP) algorithm. It is developed based on the channel and trajectory characteristics, segmentation framework, and mean distance. The proposed method is able to simplify vessel trajectory data and extract useful information effectively. The time series trajectory classification and clustering are discussed and analysed based on ADP algorithm in this paper. To verify the reasonability and effectiveness of the proposed method, experiments are conducted on two different trajectory data sets in inland waterway of Yangtze River for trajectory classification based on the nearest neighbor classifier, and for trajectory clustering based on the spectral clustering. Comprehensive results demonstrate that the proposed algorithm can reduce the computational cost while ensuring the clustering and classification accuracy.

INDEX TERMS Douglas-Peucker algorithm, trajectory compression, trajectory clustering, trajectory classification, maritime safety.

I. INTRODUCTION

The Automatic Identification System (AIS) network, including ships, AIS base stations, and shore stations, is one part of satellites and radars in navigation [1]. Based on the Very High Frequency (VHF) radios and mutual exchanging data communication [2], AIS is a self-reporting messaging system and originally conceived for collision avoidance via high-speed updates in ship-to-ship and ship-to-shore communication. It provides a vast amount of real-time information that can be used to support trajectory anomaly detection, coastal surveillance, maritime knowledge discovery, maritime situational awareness, and decision-making [3]. AIS can track, report, and locate vessels to enhance maritime supervision

The associate editor coordinating the review of this manuscript and approving it for publication was Zhixiong Peter Li.

and navigational safety [4]. Following the extensive installation and use of AIS equipment, the issues of vessel trajectory data storing, processing, and analysis arise as emerging research topics [5]. The accurate processing and extraction of massive trajectory data are vital for trajectory clustering, classification, and prediction [6]–[8].

The original AIS trajectory data contains massive noise and redundant information. The maritime navigational authorities need to manage and regulate vessels based on effective and real-time AIS data [9]. The visualization of vessel trajectories based on AIS data is conductive to detecting abnormal behaviors and aiding maritime surveillance [10]. The compression of AIS trajectory data is a valid data pre-processing way in practical applications, and also an effective method to visualize the massive trajectories. Moreover, the automatic and rational simplification threshold selection method is crucial in trajectory data compression. Therefore, effective data compressing algorithm and threshold selection method are proposed and improved to solve these problems while retaining the main features.

Trajectories are described as different types of curves with all sorts of linear features, and consist of many spatiotemporal points. A lot of classical algorithms (e.g. online and batched compression techniques [11]) are proposed and developed to compress the trajectories while preserving them with important geometrical properties. The online compression techniques include Reservoir Sampling (RS) algorithm [12], Sliding Window (SW) algorithm [13] and Normal Opening Window (NOW) algorithm [14]. The batched compression techniques mainly are associated with three algorithms, unformal sample [15], Douglas–Peucker (DP) [16] algorithm and Bellman algorithm [17]. The uniform sampling algorithm takes each i^{th} point in trajectory coordinates. The DP algorithm [18] is a classical simplification algorithm to preserve location, orientation, and shape of different trajectories based on the recursive and refinement approach of retaining the furthest vertexes. The Bellman algorithm is able to preserve the geometry feature of a certain number of points after their simplification as the original ones. The distances between points in the compression process are measured by two ways, Perpendicular Euclidean Distance (PED) and Time Synchronized Euclidean Distance (TSED) [19]. PED is the Euclidean Distance from one point to the line, and doesn't consider the temporal factor. TSED is the Euclidean Distance based on the time synchronised information, which takes the time interval ratio of different points as the weight to calculate the new projection location point.

Trajectory compression algorithms are widely used in various areas, such as maritime trajectory visualization, trajectory clustering, road traffic, pedestrian movement information, cartographic and map generalisation [20]. The theory of line compression has been widely used in trajectories processing. It's evident that the DP algorithm is one of the most effective methods to simplify and compress line data [21], and receives frequent usage [22].

Many different DP enhancements are proposed to compress trajectories. Saalfeld [23] discloses that the resulting simplified polyline by the DP algorithm is consistent with itself and adjacent features in the topology. Bertolotto and Zhou [24] develop the Saalfeld's algorithm to reduce the processing time, and integrate the new algorithm with a web-mapping system. Gudmundsson *et al.* [25] propose an extended DP algorithm to retain the geometry of self-crossing lines. The appropriate threshold interval [26] is selected from the experiment comparison results of different DP thresholds based on the AIS trajectory visualization quality. The Spatial QUalIty Simplification Heuristic Method (SQUSHM) is proposed by Muckell *et al.* [27] to reduce the computation time based on the selection of the local critical points. Chen *et al.* [28] put forward a fast polygonal approximation algorithm to simplify the GPS trajectories based on an integral square synchronous distance error criterion. Zhang *et al.* [29] present a new threshold selection method based on the minimum ship domain evaluation to define the threshold. Etienne *et al.* [30] propose an AIS trajectories simplification method based on the DP algorithm to reduce the computation time. However, the issue as to how the simplified threshold can be automatically determined remains unclear. A line simplification method is introduced in map generalisation, and Pallero [31] put forwards a robust and easy-to-implement DP algorithm to guarantee the lines without self-intersections. Birnbaum *et al.* [32] present a new trajectory compressing algorithm by splitting the trajectories into sub-trajectories based on their similarities. A new trajectory simplification algorithm namely Trajic is proposed by Nibali and He [33] based on the delta compression approach to achieve a good compression ratio and small error margin. Zhao and Shi [34] conduct clustering analysis based on the DP compression and the improved Density-Based Spatial Clustering of Applications with Noise (DBSCAN). However, all the improved DP algorithms are only based on the trajectory shape without changing the algorithm or automatically selecting the threshold.

Trajectory classification and clustering [35] are fundamental for trajectory prediction, anomaly detection and collision avoidance [36]. Trajectory classification and clustering are the important research methods of data mining, which are conducive to extracting pattern information and detecting anomaly behaviors [37], [38]. The classification and clustering processes are known as supervised and unsupervised learning methods respectively. Data pre-processing is the first step of trajectory classification and clustering, which can receive more effective information. The similarity measurement method can help calculate the distances between trajectories, which is used to measure their similarity. The distances between trajectories are a vital factor for trajectory classification and clustering [39]. There are many distance measurement methods from previous studies, for instance, simple Euclidean Distance (ED) [40], Hausdorff distance [41], HMM (Hidden Markov Model) [42], DTW (Dynamic Time Warping) [43], LCSS (Longest Common

Subsequence) [44] and so on. ED requires the equal length of all trajectories, and does not take into account the time information. Hausdorff distance is time-consuming. HMM distance sets a statistical model for each trajectory, however it has high time complexity. It has been proved that both Hausdorff and HMM have poor performance [45]. Compared with location similarity, LCSS involves more shape similarity and has high time cost. DTW can easily find the shape similarity of the trajectory, and warps the route from feature to feature [46]. Therefore, DTW is also adopted and developed in the process of similarity measurement.

The relevant literatures indicate that the DP algorithm has been widely studied and used in different fields. To the best of our knowledge, no research has been conducted on the development of automatic threshold selection and a single different threshold for each trajectory. The threshold in the original DP algorithm must be defined by its users to simplify the lines. Therefore, how to select the threshold automatically is one of the research challenges to be addressed in this work. Each time series trajectory is different from others. The other improvement is to automatically select an appropriate threshold for each trajectory. These two improvements can provide useful insights to guide and act as a solid foundation to develop future studies relating to time series trajectories. To address these two problems, we present an Adaptive DP (ADP) algorithm to select the threshold for each trajectory automatically according to the characteristics of different trajectories. Meanwhile, the classification and clustering experiments are carried out on different data sets to verify the effectiveness and robustness of the newly proposed ADP algorithm.

The remainder of the paper is organized as follows. The basic and improved algorithms are described in detail in Section II. Section III describes the proposed framework in this paper, which is used for classifying and clustering time series trajectories. The numerical experiments are carried out on different data sets to validate the effectiveness and reasonability of the ADP in the automatic threshold selection in Section IV. Finally, Section V concludes work together with future work.

II. BASIC ALGORITHMS AND IMPROVED ALGORITHMS

A. THE BASIC DOUGLAS-PEUCKER ALGORITHM

The classical DP algorithm is proposed by Douglas and Peucker, and its essence is that the line segments are used to approximate the original trajectory. The final simplified trajectory is topologically consistent with the original one, especially for the neighborhood characteristics in trajectories. The characteristic points are extracted, and then reconstructed the original trajectory which can approximate the original trajectory. The advantage of the basic DP is that it has translation and rotation invariance, the sampling results will be certain when the curve and threshold are given. However, the threshold must be pre-defined by the users to simplify the line. It is evident that the DP algorithm is able to compress

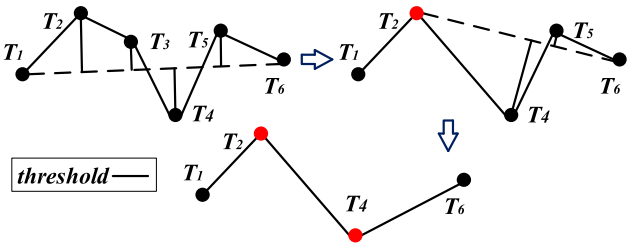


FIGURE 1. The schematic diagram of original DP algorithm.

trajectories effectively while preserving the main geometrical structures.

Suppose $T = (T_1, T_2, \cdots, T_i, \cdots, T_n)$ is the original trajectory. When the number of points is large enough, the original trajectory can be replaced by line segments $\overline{T_1T_2}, \overline{T_2T_3}, \ldots, \overline{T_{i-1}T_i}, \ldots, \overline{T_{n-1}T_n}$. To decrease the amount of trajectory points, we reconstruct the trajectory with fewer but more important points which are selected from the original point set T, $T' = (T_{k1}, T_{k2}, \cdots, T_{kj}, \cdots, T_{km})$, $T' \subseteq T$. If the characteristic points are extracted accurately, the new line segments $\overline{T_{k1}T_{k2}}, \overline{T_{k2}T_{k3}}, \ldots, \overline{T_{k(i-1)}T_{ki}}, \ldots, \overline{T_{k(m-1)}T_{km}}$ can substitute the original trajectory.

Fig. 1 is the schematic diagram of the original DP algorithm. The original trajectory is constructed by the line segments that connect 6 points $(T_1, T_2, \cdots, T_6)$. To preserve the main geometrical structure of the original trajectory and reduce the redundant trajectory points, it is necessary to extract the characteristic points from the original trajectory. The pre-defined threshold (i.e., tolerance) as a benchmark is selected to simplify the trajectory. The line $(\overline{T_1T_6})$ connecting the first point (T_1) and last point (T_6) is taken as the datum line (or a base line). Then the vertical Euclidean distance of each point to the datum line is calculated in the original trajectory. It can be seen that some of the vertical Euclidean distances are larger than the threshold, (e.g., T_2), the point related to the maximum vertical Euclidean distance will be selected to divide the original trajectory into two sub-trajectories (e.g. $\overline{T_1T_2}, \overline{T_2T_6}$). This procedure will be performed iteratively until there is no characteristic point which has a larger Euclidean distance than the threshold.

B. THE ADAPTIVE DOUGLAS-PEUCKER ALGORITHM

The threshold in the original DP algorithm must be set in advance to simplify the line. Currently, there is scanty studies on the selection of the best threshold in the literature. Therefore, this research pioneers the automatic selection of the threshold. Each time series trajectory is different from others, hence it is beneficial to select the appropriate threshold for each trajectory automatically. The success of such improvements will lay a solid foundation for the subsequent trajectory classification and clustering.

The original DP has only one threshold for all trajectories, and it is difficult to be determined. The ADP algorithm has a different threshold for each trajectory, and can automatically

select the appropriate thresholds for different trajectories. The essence of ADP is to calculate the thresholds automatically according to the distances and characteristics of all feature points. ADP can further extract and preserve key features based on the channel characteristics, trajectory characteristics, segmentation framework, and mean distance. The improved DBSCAN is an effective reprocessing method to remove the noise points. The innovation of the improved DBSCAN is that the circular neighborhood is changed into a square neighborhood. Then the square sliding window can handle all the points according to the coordinates, ε, and *MinPts*. All the points in a data set are reprocessed to extract more efficient points. The criterion is to determine whether the point coordinates are within the range of the square neighborhood. This improvement can avoid the data explosion and memory overflow. For instance, if there are 800,000 points, there will have 319,999,600,000 distance values between different points. The original DBSCAN algorithm will fail to solve the problem of this complexity.

The ADP algorithm is proposed based on the channel characteristics, trajectory characteristics, segmentation framework, and mean distance to select the threshold for each trajectory automatically.

The pseudo code of the ADP algorithm is listed as follows.

C. THE DTW ALGORITHM

From the statistical point of view, the spatio-temporal AIS trajectory is essentially a kind of time series. Suppose $Q = \{q_1, q_2, \cdots, q_m\}$ and $C = \{c_1, c_2, \cdots, c_n\}$ denote the two AIS trajectories (i.e., time series), q_i represents the value of the i^{th} point in series Q, c_j represents the value of the j^{th} point in series C, m and n indicate the length of the entire sequences of Q and C, respectively. $d\left(q_i, c_j\right)$ denotes the distance between q_i and c_j.

DTW is used to calculate the similarity between two time series. The process of DTW is described as follows. All points are sorted according to their time, then the users construct a matrix $A_{m\times n}$, and $a_{ij} = d(q_i, c_j) = \sqrt{(q_i - c_j)^2} \in A_{m\times n}$. A set of adjacent matrix elements in $A_{m\times n}$ is called a warping path, denoted by $W = \{w_1, w_2, \cdots, w_k, \cdots, w_K\}$, and $\max\{m, n\} < K \leq m+n-1$, the k^{th} point in W is represented by $w_k = \left(a_{ij}\right)_k$, the warping path must meet the following constraints:

(1) Boundary condition: $w_1 = a_{11}, w_k = a_{mn}$;

(2) Continuity and monotonicity:

if $w_{k-1} = a_{i'j'}, w_k = a_{ij}$, then $0 \leq i-i' \leq 1, 0 \leq j-j' \leq 1$. They together ensure that every coordinate in two trajectories can appear in W, and the dotted line between the trajectories does not intersect. Certainly, the time at each point is also monotonic in W.

DTW can find a path with a minimum of the cost of the optimal path based on dynamic programming [47]. The algorithm steps are described as follows:

Step1. Starting from the start point of the two sequences i, j to calculate the DTW distance $D(i, j)$ between the two

Algorithm 1 ADP Algorithm

Input: $(x_i^j \pm \varepsilon, y_i^j \pm \varepsilon)$, *MinPts*

//ε is the step size, $(x_i^j \pm \varepsilon, y_i^j \pm \varepsilon)$ is the square sliding window, *MinPts* is the number of points covered by the sliding window.

$T_i^j = (x_i^j, y_i^j, t_i^j) \in D,\ i = 1, \cdots, n,\ j = 1, \cdots, m,$
$T_1^j = (x_1^j, y_1^j, t_1^j),\ T_n^j = (x_n^j, y_n^j, t_n^j)$

//D is the trajectory data set, T_i^j is the i^{th} point in the j^{th} trajectory, T_1^j is the starting points of each trajectory, T_n^j is the ending points of each trajectory.

$(xx_t, yy_t) \in S,\ t = 1, \cdots, m \cdot n.$

//S is the point data set in ascending order.

Channel characteristics, trajectory characteristics

Output: θ^j, d_i^j, TT_i^j.

//θ^j is the automatic threshold of each trajectory, d_i^j is the Euclidean distances from all points to the baseline, TT_i^j is the simplified trajectories (the characteristic point set).

//Data preprocessing based on the improved DBSCAN.//

1: Set ε *and MinPts* based on the latitude and longitude in trajectory data set.
2: For $j = 1$ to $j = m$
3: For $i = 1$ to $i = n$
4: Save all the points in ascending order of the abscissa.
5: End
6: End

//Mark all points as the core points, boundary points and noise points.//

7: for $t = 1$ to $t = m \cdot n$
8: IF
$xx_{t+ii} \in [xx_t - \varepsilon,\ xx_t + \varepsilon],\ yy_{t+ii'} \in [yy_t - \varepsilon,\ yy_t + \varepsilon],\ ii, ii' \in Z^+$
9: $\min(ii, ii') < MinPts$
10: THEN mark this point as the core point.
11: IF $xx_{t+ii} = |xx_t + \varepsilon|,\ yy_{t+ii'} = |yy_t + \varepsilon|,\ ii, ii' \in Z^+$
12: THEN mark this point as the boundary point.
13: ELSE
14: THEN mark this point as the noise point;
15: End
16: Delete the noise points.

//The automatic threshold selection of each trajectory based on the segmentation framework.//

17: For $j = 1$ to $j = m$
18: $y = \frac{y_n^j - y_1^j}{x_n^j - x_1^j}(x - x_1^j) + y_1^j = k(x - x_1^j) + y_1^j$;
19: // The baseline equation calculation of each trajectory.
20: $d_i^j = \frac{\left|k(x-x_1^j)+y_1^j-y\right|}{\sqrt{1+k^2}}$;
// The Euclidean distances from all points to the baseline are calculated;
21: IF the dataset is straight trajectory,
22: THEN $\theta^j = |k| \cdot \sum_{i=2}^{n-1} d_i^j/(n-2)$
// The automatic threshold of each trajectory is calculated.
23: ELSE IF the dataset is curved trajectory,

Algorithm 1 *(Continued.)* ADP Algorithm

24: THEN $\theta^j = \sum_{i=2}^{n-1} d_i^j/(n-2)$

// The automatic threshold of each trajectory is calculated.

25: ELSE the dataset is complex trajectory,

26: THEN divide the complex trajectory into straight and curved trajectory based on the datum line and max d_i^j, and the channel characteristics. Return to step 21.

// The segmentation framework is formed based on the channel and trajectory characteristics.

27: End

//The compression process of each trajectory.//

28: For $j = 1$ to $j = m$

29: For $i = 2$ to $i = (n-1)$

30: // The starting point and ending point of each trajectory must be preserved.

31: IF $d_i^j > \theta^j$

32: THEN point i must be preserved.

33: ELSE

34: point i should be deleted.

35: End

36: End

37: End

38: The reserved points of each trajectory constitute the compressed point sets TT_i^j.

// The compression ratio are calculated. //

39: For $j = 1$ to $j = m$

40: $\sigma^j = \sum preserve(i)/n$

41: End

sequences.

$$\begin{cases} D(1,1) = d_{11} \\ D(i,j) = d_{ij} + \min\{D(i-1,j-1), \\ \qquad\qquad D(i,j-1), D(i-1,j)\} \end{cases} \tag{1}$$

$$d_{ij} = d(q_i, c_j) = \sqrt{(q_i - c_j)^2} \in D_{m\times n} \tag{2}$$

where $i = 2, 3, \cdots, m, j = 2, 3, \cdots n$, and $d(q_i, c_j)$ denotes the Euclidean distance.

Step2. The distance $D(i, j)$ of the end point in the two sequences is the DTW distance of the two sequences.

The time complexity of the Euclidean distance and DTW are $O(n)$ and $O(n^2)$ respectively. DTW does not require that the two sequences are equal.

III. THE PROPOSED METHOD FRAMEWORK

The proposed ADP algorithm can automatically select a threshold for each trajectory, and hence significantly compress the trajectories, and calculate the compression rate according to the characteristic of each trajectory. It can reduce the amount of data, save the follow-up calculation time and preserve the important structural properties well. The ADP and DTW algorithms can accelerate the data processing and similarity measurement between massive time series.

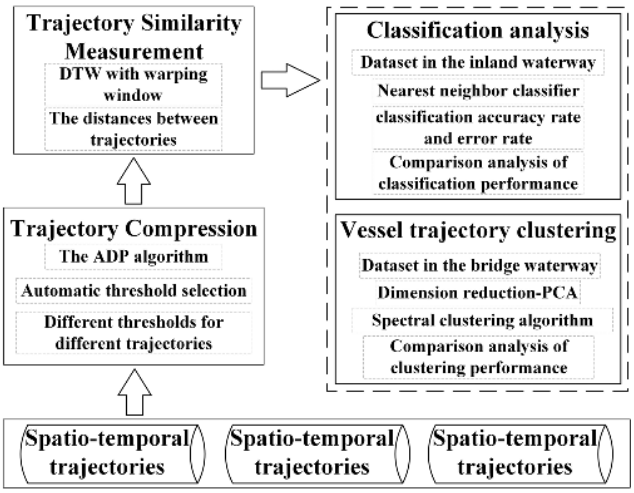


FIGURE 2. The proposed method and the experiment flowchart.

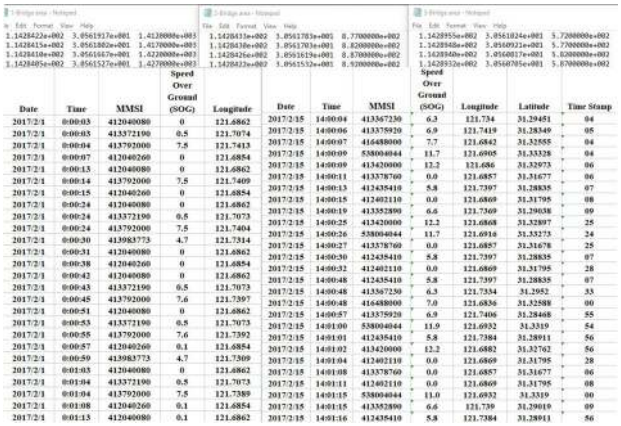


FIGURE 3. The visualization of data sets.

The ADP algorithm is proposed to compress the time series data sets, and the DTW algorithm with a warping window is introduced to calculate the distances between time series. Then the classification and clustering analysis are carried out in two different time series data sets to verify the validity and effectiveness of the proposed algorithms. The experiment flowchart is shown as follows.

The threshold is the main factor that determines the trajectory compression quality. When its value becomes too small, it will lead to a high calculation cost, while if it becomes too large, it will not capture the original feature of the trajectory. Manual selection of the best compression threshold is the shortcoming of the current research of trajectory compression. To solve this problem, a novel ADP algorithm is proposed to automatically select the thresholds while preserving the structural and geometric characteristics well. Moreover, DTW is chosen to calculate the distance between the time series accurately. This paper not only presents a new algorithm, but also analyses its validity and feasibility through different experiments in the ensuing sections.

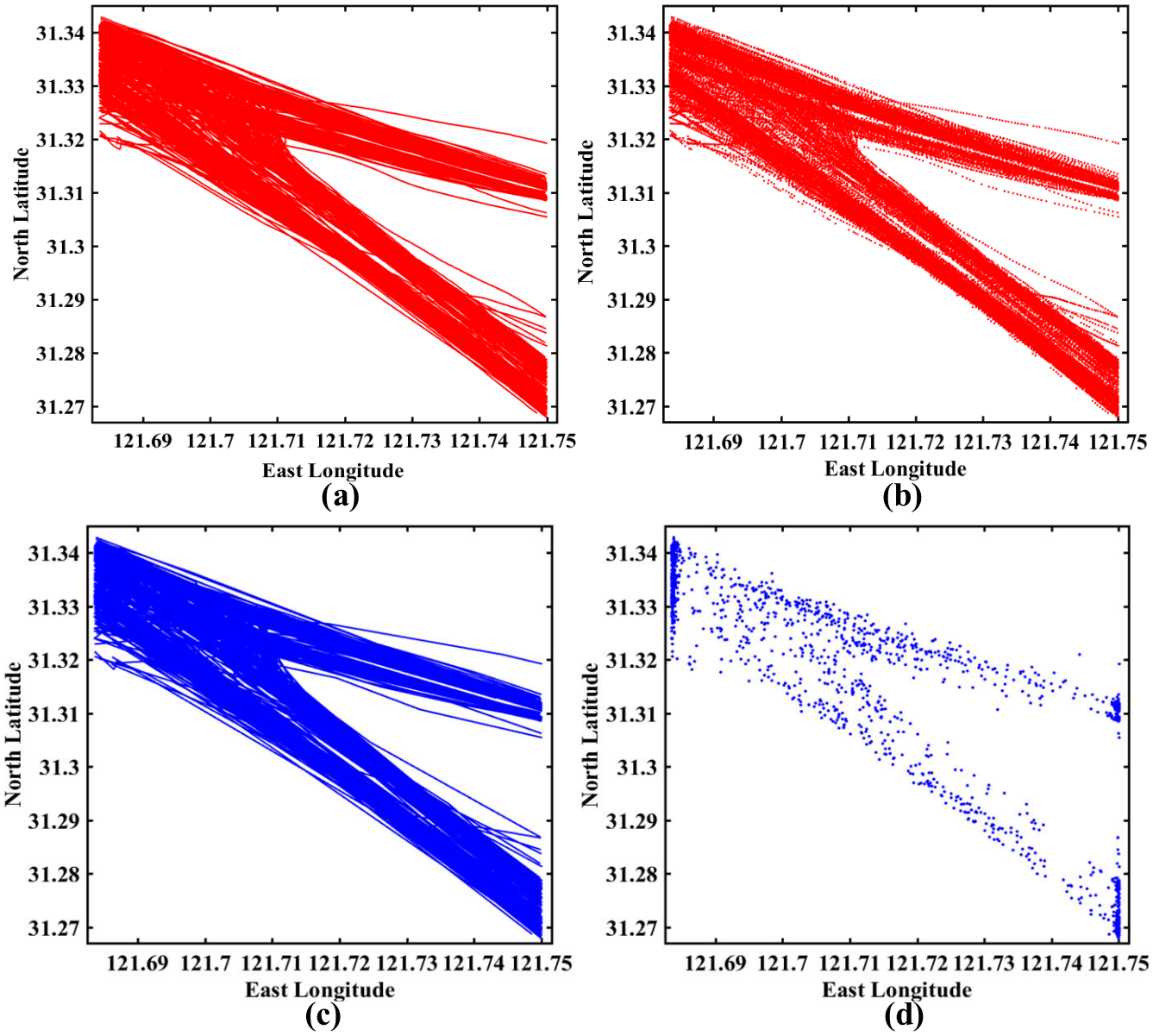


FIGURE 4. The original and compressed trajectories of inland waterway: (a) the original vessel trajectories; (b) the original point data; (c) the trajectories after compression by ADP; (d) the point data after compression by ADP.

IV. EXPERIMENT RESULTS AND EVALUATION OF TWO DATA SETS

A. EXPERIMENTAL SETUP AND DATA SETS

Two experiments are performed using 64-bit Windows 10 on a 2.60 GHz Intel Core i7-5600U CPU equipped with 8 GB memory. We implemented the proposed ADP, classification, and clustering methods using MATLAB R2016a, and DTW with a warping window algorithm using MATLAB R2016a and C language.

To verify the accuracy and efficiency of the proposed ADP algorithm, numerical experiments are implemented based on real AIS trajectory data of an inland waterway for classification and the bridge area waterway for clustering. The inland waterway data set is collected from Yangtze River, and has 404 trajectories with 74,263 points. The AIS trajectory data set in the bridge area waterway is the spatial-temporal trajectories with time, longitude, latitude and speed, etc. The AIS trajectory data sets in the bridge area waterway are three-dimensional time series. The experimental data are collected from the AIS base station in the Wuhan section of the Yangtze River. The bridge area waterway data set includes the AIS trajectory data of 377 vessels with 58,296 points. The visualization of data sets is shown in Fig. 3.

B. TRAJECTORY COMPRESSION RESULT OF ADP ON A CLASSIFICATION DATA SET

In this paper, the validity of the proposed ADP algorithm is demonstrated by the real vessel trajectory data set.

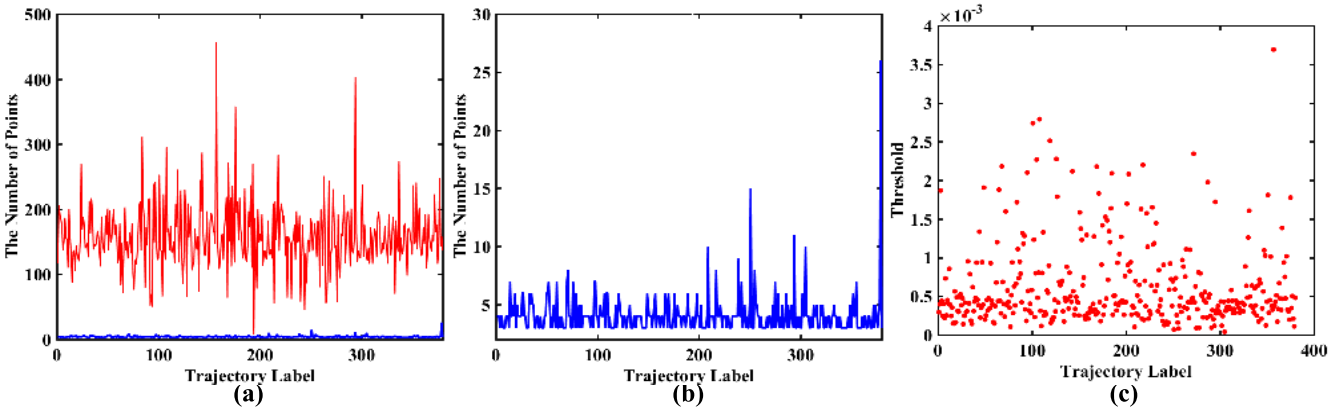


FIGURE 5. Visualization of the number of points and the threshold based on ADP: (a) the number of points before and after compression; (b) the number of points after compression; (c) the threshold of each trajectories.

Data cleansing is the basic step of trajectory visualization. The incomplete, repeated, redundant, and invalid trajectory data are deleted based on the trajectory acquisition time and time interval. The original data set includes 404 trajectories, and there are 380 trajectories with 59,888 points are preserved after data cleansing.

In the first step, the longitude range is [121.6830, 121.7502] and latitude range is [31.267, 31.3435] in the selected trajectory data set. Then the parameter ε is set to 0.0003 and the parameter *MinPts* is set to be 5 in the improved DBSCAN based on the longitude and latitude range of the trajectory points.

The proposed ADP algorithm is used for compressing the trajectories. The visualization of original and compressed trajectories are shown in Fig. 4. Fig. 4(a) and Fig. 4(c) are the original vessel trajectories and the compressed trajectories respectively. Meanwhile, Fig. 4(b) and Fig. 4(d) show the point data before and after compression respectively. It can be seen that from Fig. 4(b) and Fig. 4(d), the data volume is significantly reduced. The number of points on all trajectories is 1,553 after the trajectory compression.

The number of points and the threshold based on the ADP algorithm are shown in Fig. 5. Fig. 5 (a) displays the number of points before and after compression, where the red line expresses the number of points in original trajectories and the blue one is the number of points in compressed one. The number of points after compression is shown in Fig. 5 (b), which further clearly shows the number of points. The threshold of different trajectories is shown in Fig. 5 (c), and the range is $[0,\ 4 \times 10^{-3}]$. The threshold is automatically selected based on features of different trajectories. To show the classification performance more clearly, the vessel trajectories are further analysed based on their movement courses. The trajectories of up-bound and down-bound vessels are separated based on their different sailing directions. Then there are 201 up-bound and 179 down-bound vessels, respectively. The visualization of original and compressed trajectories of up-bound vessels are shown in Fig. 6. Fig. 6 (a) and Fig. 6 (c) are the original vessel trajectories and the ones after compression, respectively. Meanwhile, Fig. 6 (b) and Fig. 6 (d) show the point before and after compression respectively. It can be seen from Fig. 6 (b) and Fig. 6 (d) that the data volume is significantly reduced. There are 31,503 points on 201 up-bound trajectories, and only 828 points after using our ADP compression algorithm.

The number of points and the thresholds of up-bound trajectories are shown in Fig. 7. Fig. 7 (a) displays the number of points before and after compression, where the red line indicates the number of points in original trajectories and the blue one is the number of points of all trajectories after compression. The number of points after compression is shown in Fig. 7 (b), which further clearly shows the number of points. The thresholds of different trajectories are shown in Fig. 7 (c), and the range is $[0,\ 4 \times 10^{-3}]$. The threshold is automatically selected based on the features of different trajectories.

The visualization of original and compressed trajectories of the down-bound vessels are shown in Fig. 8. Fig. 8 (a) and Fig. 8 (c) are the original vessel trajectories and the ones after compression, respectively. Meanwhile, Fig. 8 (b) and Fig. 8 (d) show the points before and after compression respectively. It can be seen from Fig. 8 (b) and Fig. 8 (d) that the data volume is significantly reduced. There are 28,385 points on 179 down-bound trajectories, and only 725 points after using the ADP compression algorithm.

The number of points and the thresholds of the down-bound trajectories are shown in Fig. 9. Fig. 9 (a) displays the number of points before and after compression, where the red line represents the number of points in original trajectories and the blue one is that in compressed trajectories. The number of points after compression is shown in Fig. 9 (b), which further clearly shows the number of points. The thresholds of different trajectories are shown in Fig. 9 (c), and the range is $[0,\ 3 \times 10^{-3}]$. The thresholds are automatically selected based on the features of different trajectories.

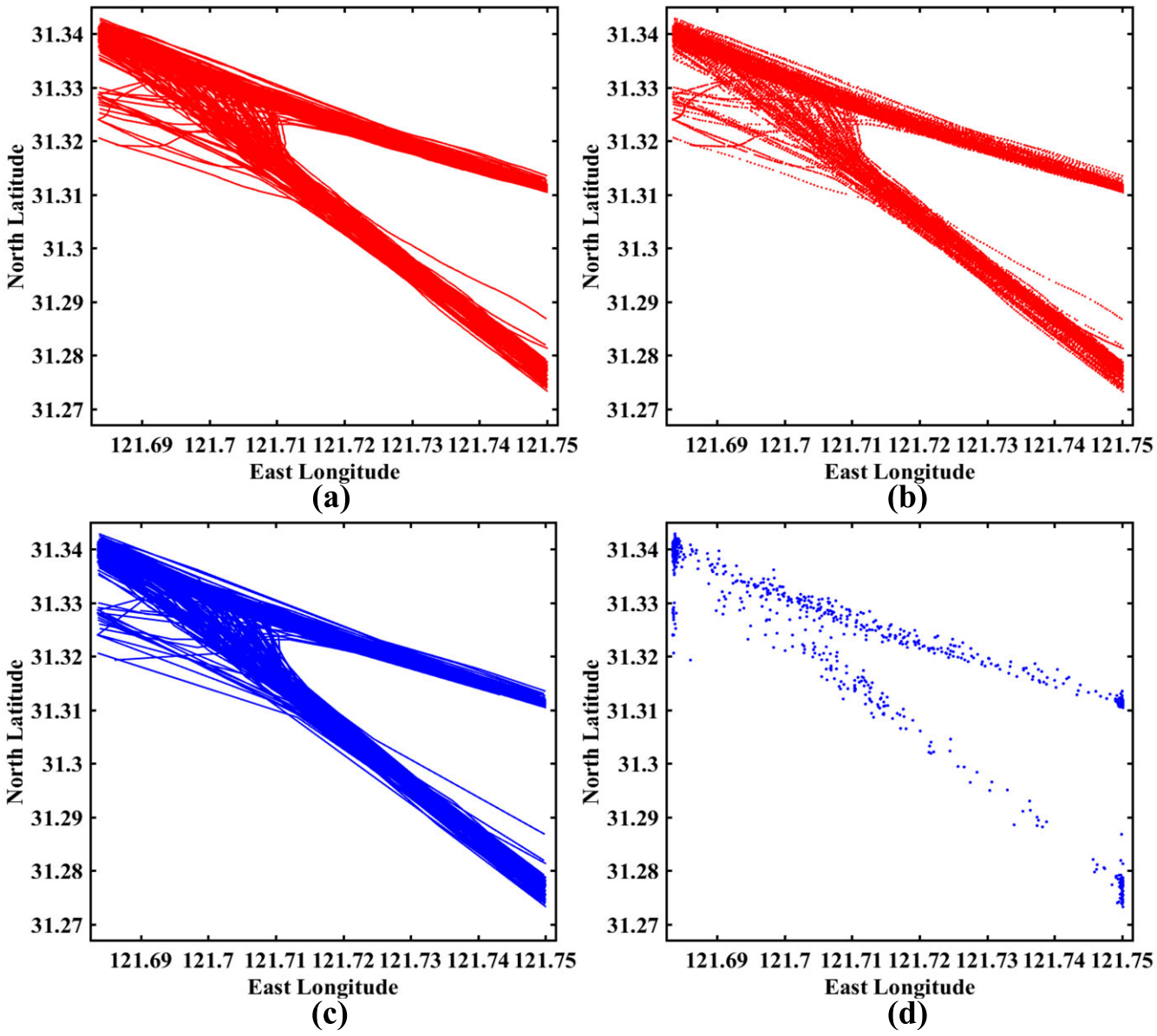


FIGURE 6. The original and compressed trajectories of up-bound vessels: (a) the original vessel trajectories; (b) the original point data; (c) the compressed trajectories by ADP; (d) the point data after compression by ADP.

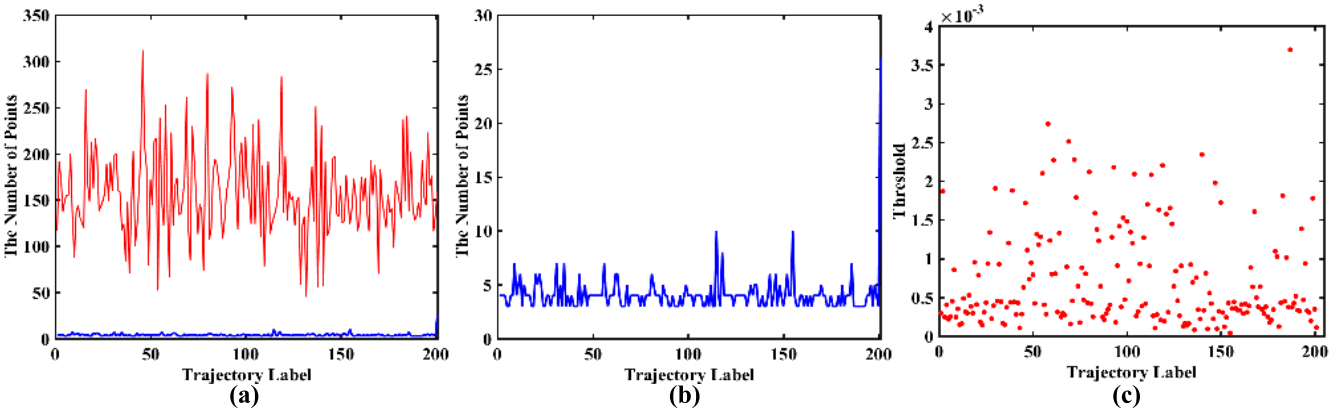


FIGURE 7. Visualization of the number of points and the threshold of up-bound trajectories: (a) the number of points before and after compression; (b) the number of points after compression; (c) the threshold of each trajectory.

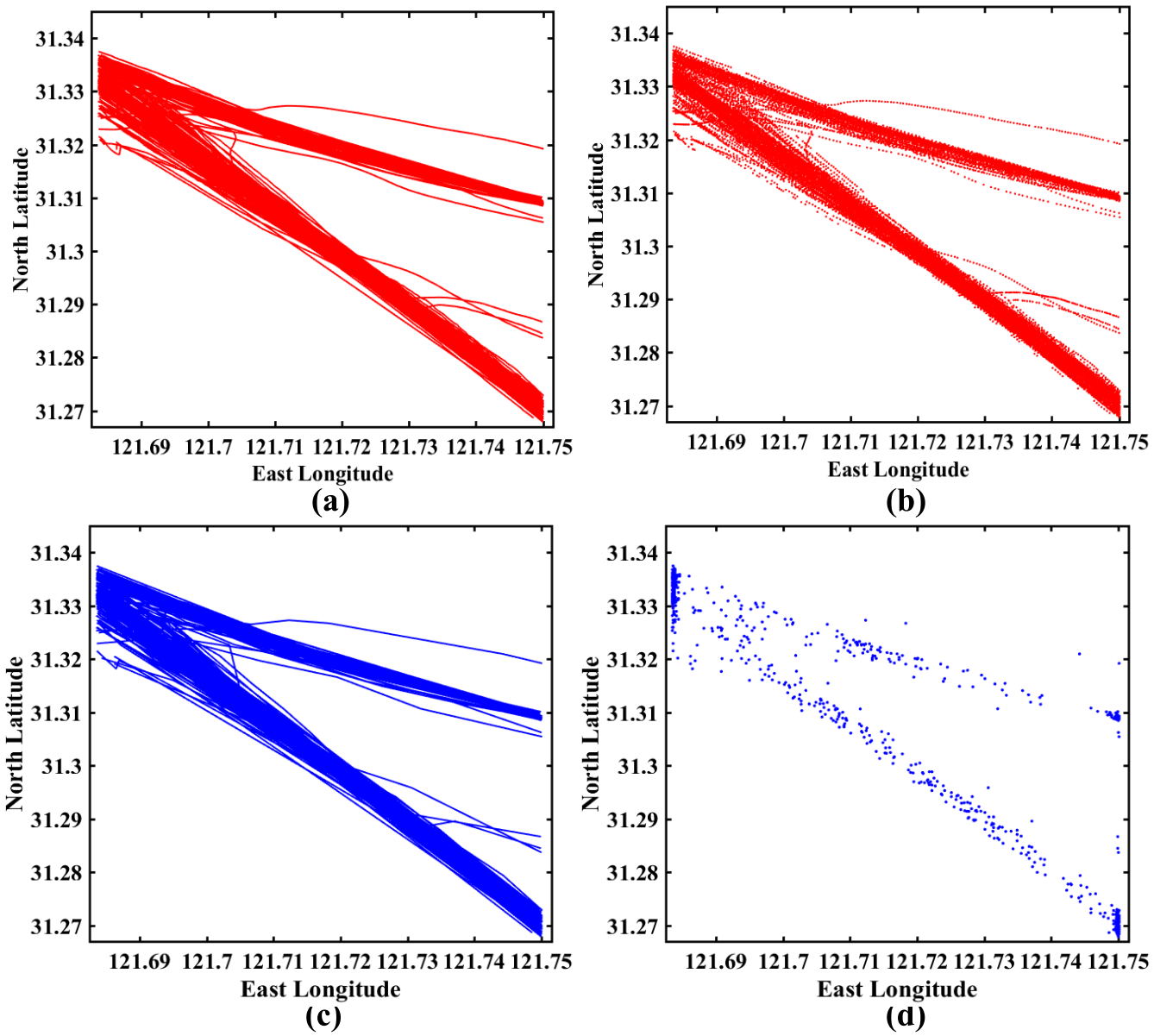


FIGURE 8. The original and compressed trajectories of down-bound vessels: (a) the original vessel trajectories; (b) the original point data; (c) the compressed trajectories by ADP; (d) the point data after compression by ADP.

C. TRAJECTORY COMPRESSION RESULT OF ADP ON CLUSTERING DATA SET

The data cleansing method used in this section is the same with the above process. The original data set includes 377 trajectories, and there are 324 trajectories with 25,678 points are preserved after data cleansing.

In the first step, the longitude range is [114.2746, 114.2919] and latitude range is [30.545, 30.562] in the selected trajectory data set. Then the parameter ε is set to 0.0006 and the parameter *MinPts* is set to be 4 in the improved DBSCAN based on the longitude and latitude range of trajectory points.

The visualization of the trajectories before and after compression based on the ADP algorithm are shown in Fig. 10. Fig. 10 (a) and Fig. 10 (c) are the original vessel trajectories and the ones after compression respectively. Meanwhile, Fig. 10 (b) and Fig. 10 (d) show the point data before and after compression respectively. It can be seen that from Fig. 10(b) and Fig. 10 (d), the data volume after compression is significantly reduced.

The number of points and the thresholds are shown in Fig. 11. Fig. 11 (a) displays the number of points before and after compression, where the red line expresses the number of points in original trajectories and the blue one is the number of points in the trajectories after compression. The number of points after compression is clearly shown in Fig. 11 (b). The thresholds of different trajectories are shown in Fig. 11 (c), the range is [0, 1×10^{-3}] and the

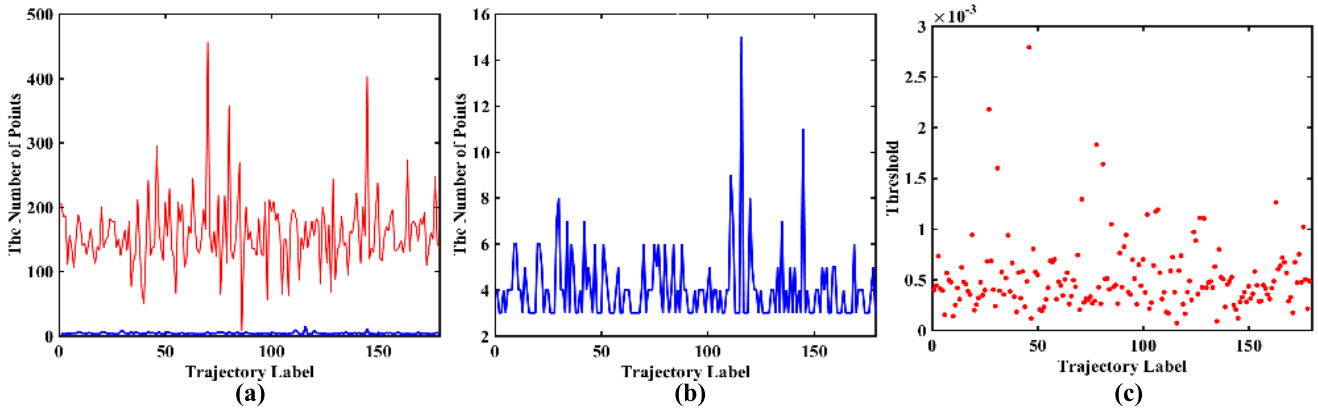


FIGURE 9. Visualization of the number of points and the threshold of down-bound trajectories: (a) the number of points before and after compression; (b) the number of points after compression; (c) the threshold of each trajectory.

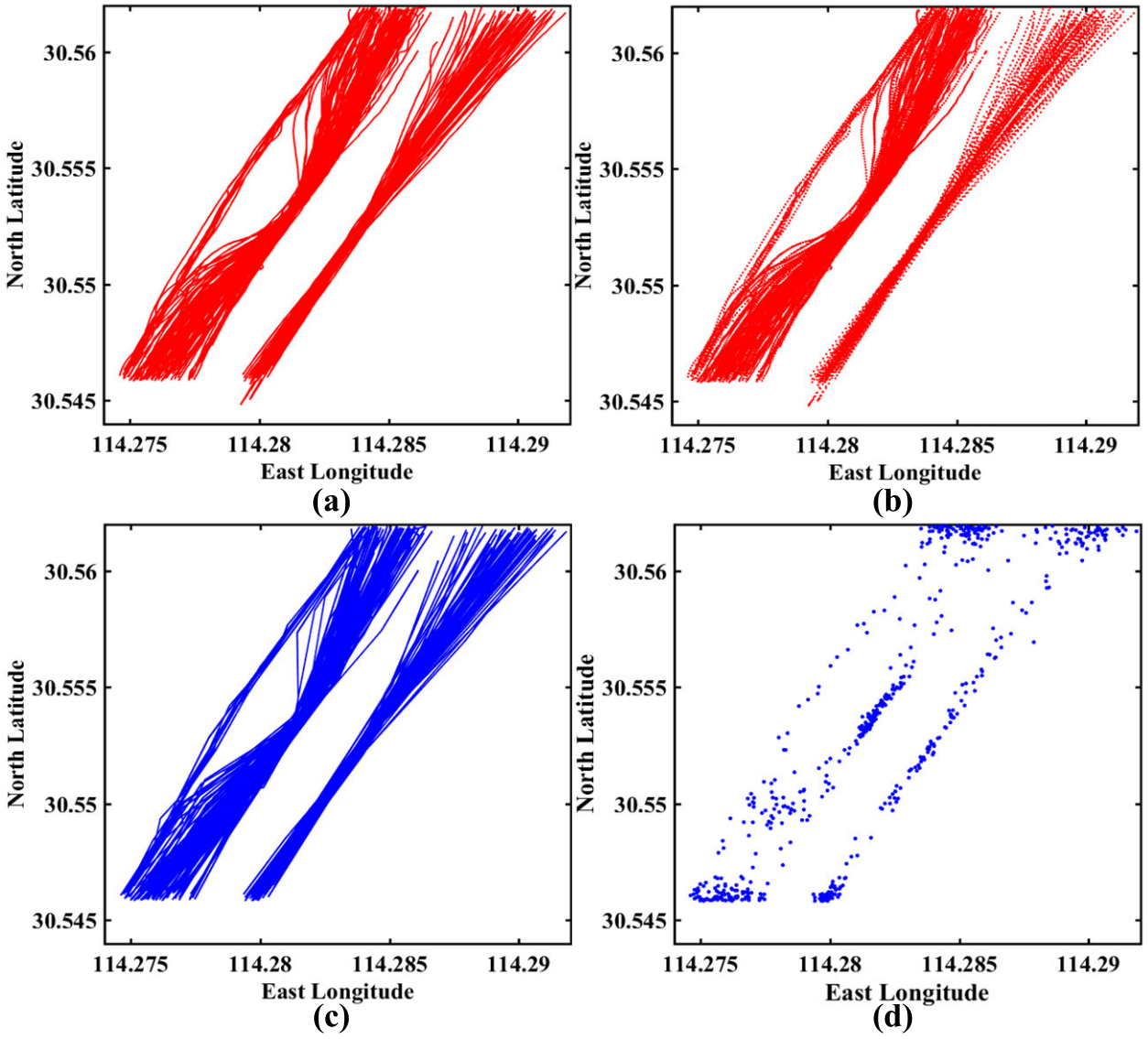


FIGURE 10. The original and compressed trajectories in bridge area waterway: (a) the original vessel trajectories; (b) the original point data; (c) the compressed trajectories by ADP; (d) the point data after compression by ADP.

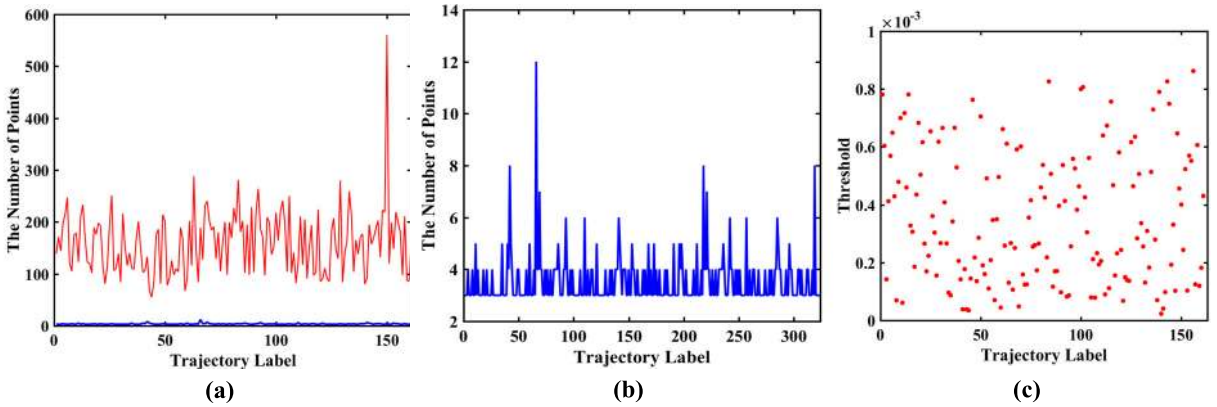


FIGURE 11. Visualization of the number of points and the threshold: (a) the number of points before and after compression; (b) the number of points after compression; (c)the threshold of each trajectory.

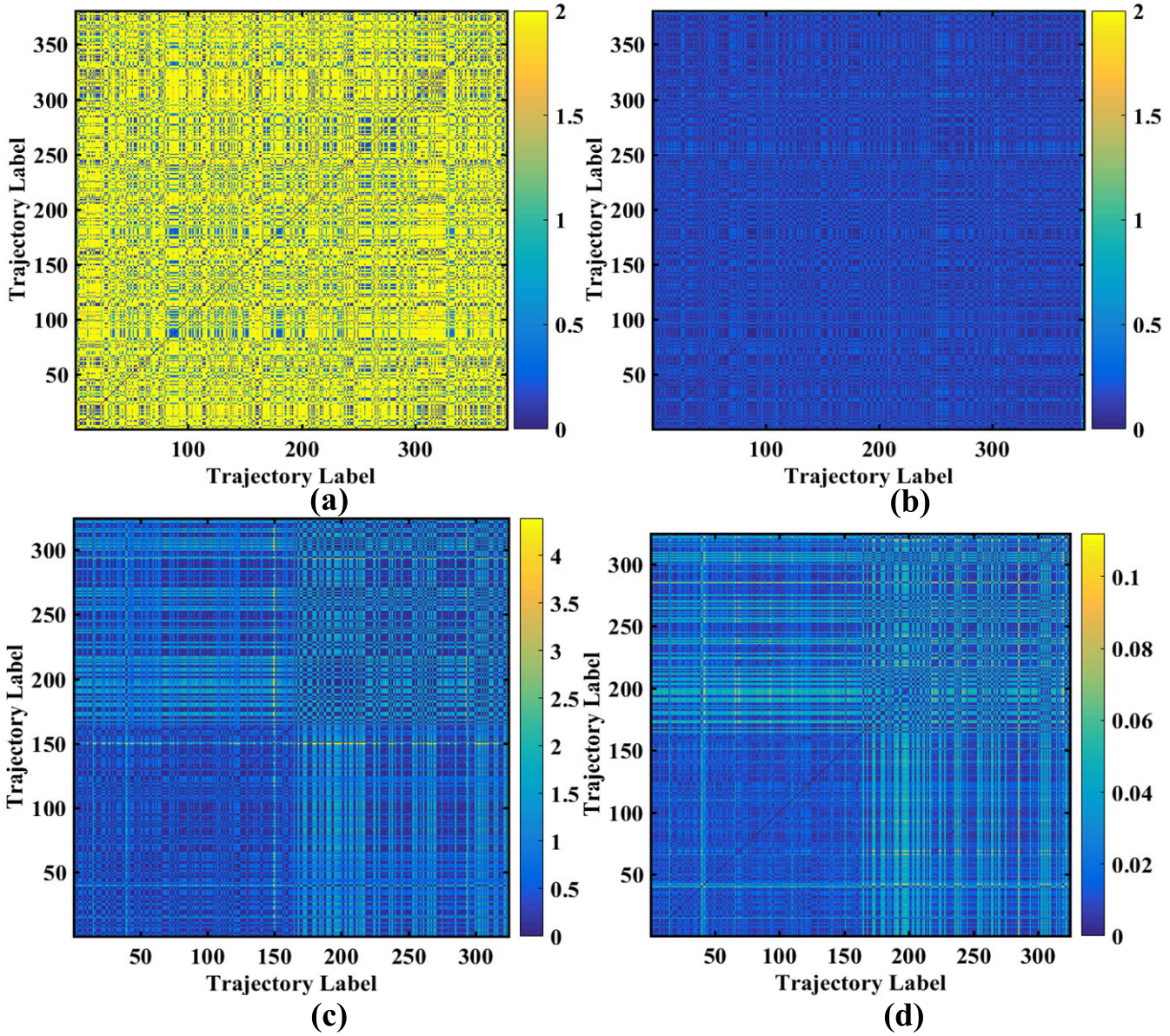


FIGURE 12. Visualization of the distance matrix: (a) 2D image visualization of the 380 × 380 distance matrix before trajectory compression; (b) 2D image visualization of the 380 × 380 distance matrix after trajectory compression; (c) 2D image visualization of the 324 × 324 distance matrix before trajectory compression; (b) 2D image visualization of the 324 × 324 distance matrix after trajectory compression.

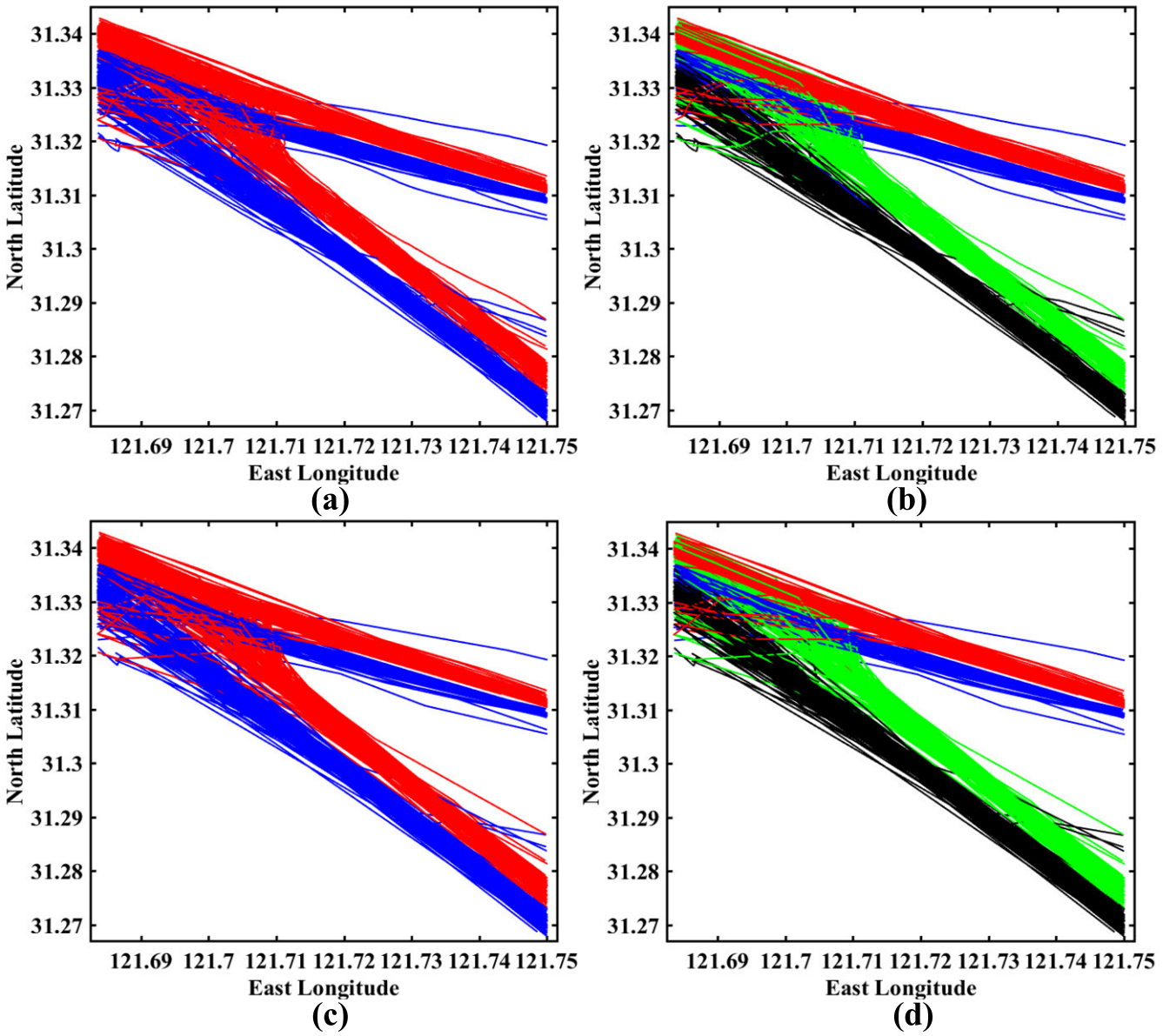


FIGURE 13. The classification results of data set in the inland waterway: (a) visualization of the original data set; (b) the classification results of the original data set; (c) visualization of the data set with ADP; (d) the classification results of the compressed data set.

thresholds are automatically selected based on trajectories characteristics.

D. TRAJECTORY SIMILARITY MEASUREMENT BASED ON DTW

There are 380 trajectories in the inland waterway data set, while 324 trajectories in the bridge area waterway data set after trajectory compression. The distances between the trajectories are calculated by DTW. The distance matrix visualization for different data sets is shown in Fig. 12. Fig. 12(a) and Fig. 12(b) are the 2D image visualization of the 380 $\times$ 380 distance matrix before and after trajectory compression, respectively. The 2D image visualization of the 324 $\times$ 324 distance matrix before and after trajectory compression are shown in Fig. 12 (c) and Fig. 12 (d), respectively.

E. CLASSIFICATION RESULTS

1) VISUALIZATION OF CLASSIFICATION RESULTS IN INLAND WATERWAYS

The classification results of the original and compressed data sets are shown in Fig. 13. The original trajectories are shown in Fig. 13(a), where the red lines are the trajectories of the up-bound vessels and the blue ones represent the one of the down-bound vessels. The classification result of the original data set is shown in Fig. 13(b), where the red, blue, green, and black colors represent different classes respectively. The blue line in the black one is the misclassification trajectory. Fig. 13(c) is visualization of the compressed data set, where the red and blue lines have the same meaning in Fig. 13(a). The classification result of the compressed data set is shown in Fig. 13(d), and the trajectories are clearly divided into

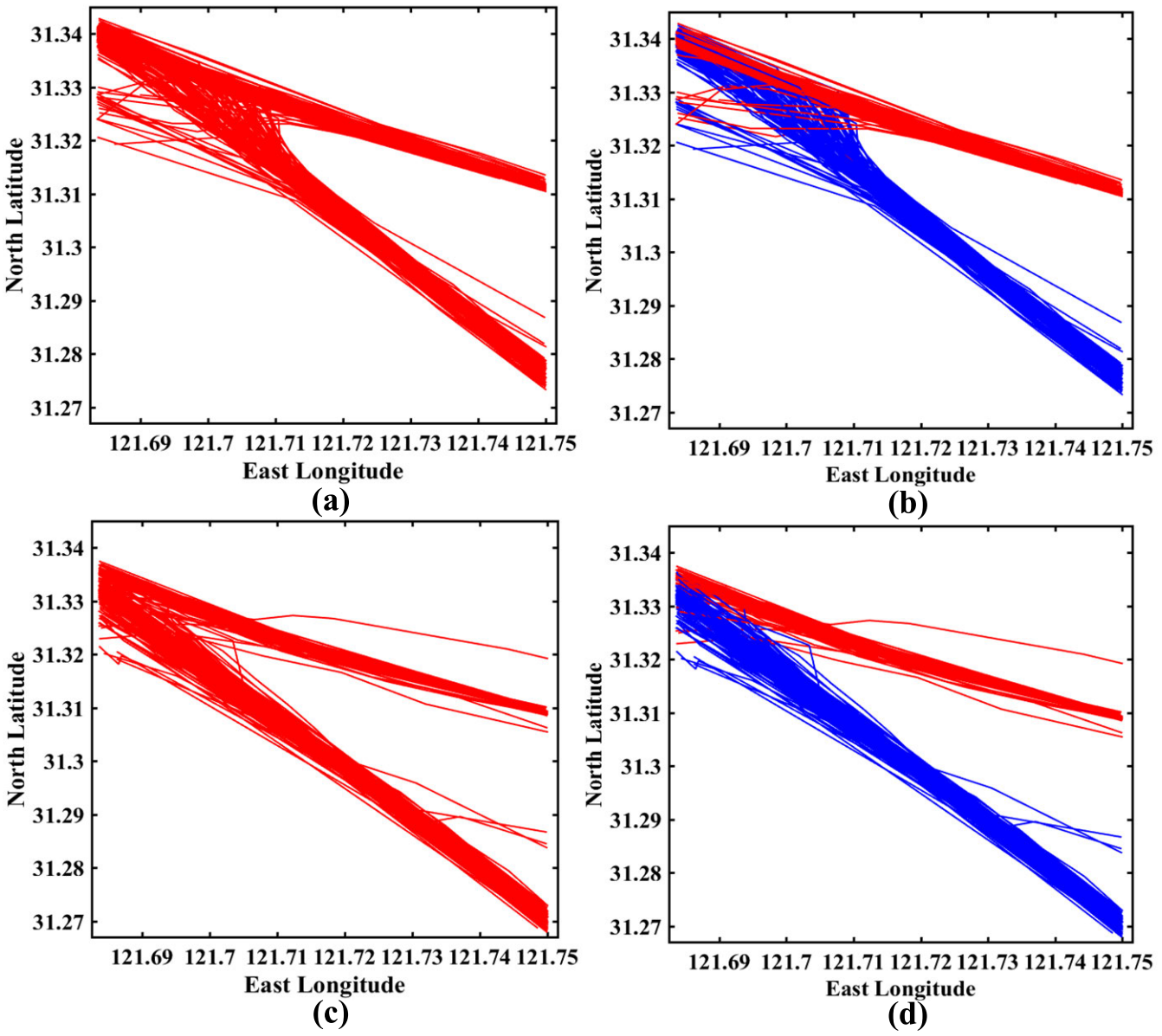


FIGURE 14. The classification results of the inland waterway data set based on different course: (a) visualization of the up-bound vessel trajectories; (b) the classification results of (a); (c) visualization of the down-bound vessel trajectories; (d) the classification results of (c).

four categories. The classification accuracy of these four categories is 100%. The original data set have 59,888 points, while the compressed data set only have 1,553 ones. The calculation time and processing time are significantly reduced, which provide theoretical basis and technical support for realizing big data research and analysis in future.

2) VISUALIZATION OF TRAJECTORY CLASSIFICATION RESULTS BY COURSE

The classification results of the data set in the inland waterway based on different courses are compared and shown in Fig. 14. Fig. 14 (a) is visualization of the up-bound vessel trajectories, and Fig. 14 (b) shows the classification result of the up-bound vessel trajectories. The classification accuracy of the up-bound vessel trajectories is 100%. The visualization of the down-bound vessel trajectories is shown in Fig. 14 (c), and the classification result of the down-bound vessel trajectories is shown in Fig. 14 (d). The classification accuracy of the down-bound vessel trajectories is also 100%.

F. CLUSTERING RESULTS

1) VISUALIZATION OF CLUSTERING RESULTS IN BRIDGE WATERWAYS

Spectral Clustering (SC) is based on the spectral graph partition theory, and its essence is to transform the clustering problem of a sample space into the optimal partition problem of graph. It can divide the graph into several subgraphs, which have no intersections between each other. The points have the highest similarity in the same subgraph and the lowest

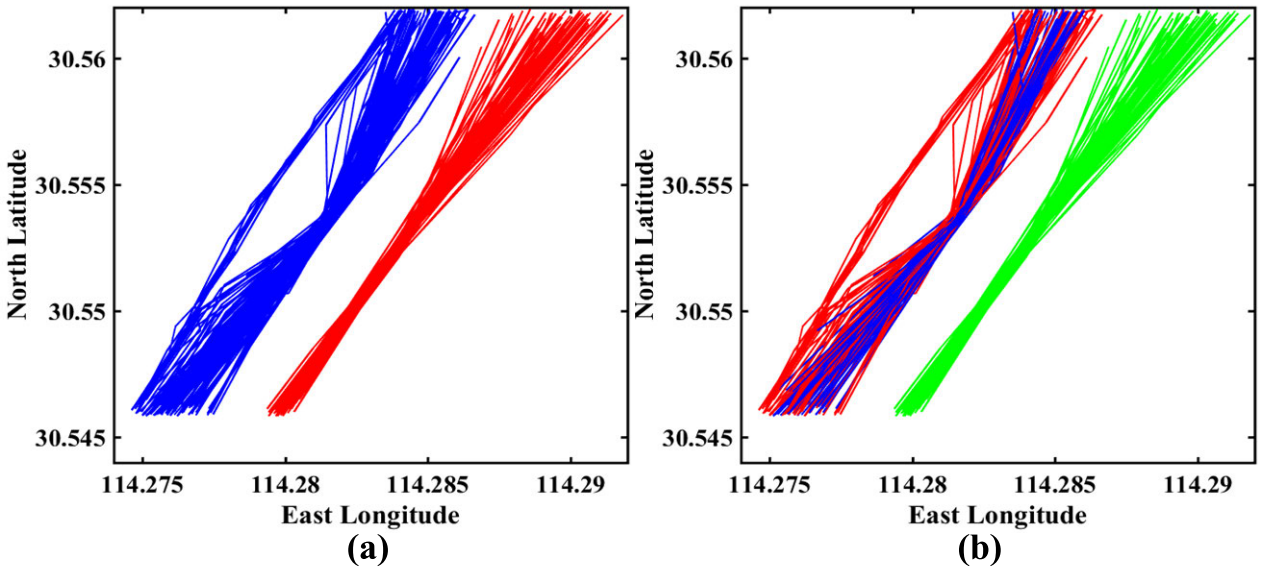


FIGURE 15. The clustering results of the bridge area waterway data set: (a) the clustering results based on spectral clustering (k = 2); (b) the clustering results based on spectral clustering (k = 3).

TABLE 1. The top 10 eigenvalues (EV) and the corresponding accumulative contribution rate (ACR) with PCA based on the original DTW.

EV	289.28	19.2698	12.3493	1.72216	0.771316	0.38507	0.265673	0.198458	0.124328	0.107943
ACR	89.28%	95.23%	98.73%	99.27%	99.50%	99.38%	99.70%	99.77%	99.80%	99.84%

similarity between different subgraphs. SC can identify the sample space with an arbitrary shape and converge to the global optimal solution. The basic idea of SC is to classify the feature vectors received by the feature decomposition based on the similarity matrix of the sample data.

The clustering results of data set in the bridge area waterway are shown in Fig. 15. Fig. 15 (a) is the clustering results based on SC when the number of clustering centers is 2. Fig. 15 (b) is the clustering results based on spectral clustering when the number of clustering centers is 3.

2) VALIDATION OF THE NUMBER OF CLUSTERS

Table 1 is the accumulative contribution rate of the top ten eigenvalues based on ADP and DTW, and the top two eigenvalues and the top three are 95.23% and 98.73%, respectively. The number of clusters is set to 2, and the performance analysis of two or three clustering centers are shown and analysed in the previous experiments. It can be clearly seen from Fig.15, the performance of two clustering centers is better than the three ones. The verification of the number of clustering centers further proves the effectiveness of the proposed compression algorithm and the clustering algorithm.

G. COMPARATIVE ANALYSIS OF TIME COMPLEXITY

The time complexity of the used methods in this work are as follows: DTW is $O(n^2)$, the nearest neighbor classification is $O(n)$, and spectral clustering is $O(n^2)$. In the above time complexity expressions, n represents the number of AIS trajectories. The comparison results before and after trajectory compression are listed in Table 2.

TABLE 2. Comparison results before and after trajectory compression.

Symbol	Original trajectory data set	Trajectory data set after preprocessing	Trajectory data set after compression
$N_{TRA\text{-}I}$	404	380	380
$N_{POINT\text{-}I}$	74,263	59,888	1,553
$T_{DTW\text{-}I}$	-	250.292s	183.762s
$T_{clas\text{-}I}$	-	6.348s	3.256s
$R_{clas\text{-}accu\text{-}I}$	-	99%, 99%	100%, 100%
$N_{TRA\text{-}B}$	377	324	324
$N_{POINT\text{-}B}$	58,296	25,678	1,154
$T_{DTW\text{-}B}$	-	174.246s	108.141s
$T_{clus\text{-}B}$	-	5.322s	3.818s
$R_{clus\text{-}accu\text{-}B}$	-	96.9%	100%

$N_{TRA\text{-}I}$ represents the number of trajectories in the inland waterway data set. $N_{POINT\text{-}I}$ indicates the number of points in the inland waterway data set. $T_{DTW\text{-}I}$ represents the running time of DTW in the inland waterway data set. $T_{clas\text{-}I}$ indicates the classification time in the inland waterway data set. $R_{clas\text{-}accu\text{-}I}$ represents the classification accuracy in the inland waterway data set. $N_{TRA\text{-}B}$ indicates the number of trajectories in the bridge area waterway. $N_{POINT\text{-}B}$ represents the number of points in the bridge area waterway. $T_{DTW\text{-}B}$ represents the running time of DTW in the bridge area waterway data set. $T_{clus\text{-}B}$ indicates the classification time in the bridge area waterway data set. $R_{clus\text{-}accu\text{-}B}$ represents the classification accuracy in the bridge area waterway data set.

In the inland waterway data set, there are 380 trajectories, consisting of 59,888 points and 1,553 points before and after compression. The running time of DTW before and after compression is 250.292 s and 183.762 s, respectively. The classification running time is 6.348 s and 3.256 s, respectively. Whether the course is considered or not, the classification accuracy after trajectory compression is always 100%.

The running time and the classification accuracy further verify the validity of the proposed trajectory compression algorithm.

The data set in the bridge area waterway includes 324 trajectories, consisting of 25,678 and 1,154 points before and after trajectory compression. The running time of DTW before and after compression is 174.246 s and 108.141 s, respectively. The clustering time is 5.322 s and 3.818 s, respectively. The clustering accuracy is 96.9% and 100%, respectively.

The accuracy of classification and clustering after trajectory compression is better than that before trajectory compression. The running time of different parts after trajectory compression is less than that before trajectory compression. The comparison results before and after trajectory compression have further prove the effectiveness and feasibility of our proposed trajectory compression algorithm.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel trajectory compression algorithm to extract valid trajectory features, accelerate the similarity measures between massive AIS trajectories, improve the accuracy of classification and clustering, and reduce the processing and running time. The quality of trajectory compression and the accuracy of similarity measurement are the key factors to determine trajectory classification and clustering. The traditional DP compression threshold needs to be set manually or selected by experimental comparison. The proposed method could significantly compress the AIS trajectories while maintaining the main geometrical structures, and also automatically calculate a different threshold for each trajectory. It is always important to guarantee the structural features and increase the compression quality in trajectory clustering and classification. Therefore, trajectory similarity measurement based on ADP, the classification accuracy, and the clustering accuracy could be significantly improved and accelerated in practical applications. It is of significance for realizing big data research in future. Numerous experiments of trajectory classification and clustering are implemented using different trajectory data sets to verify the effectiveness and feasibility of the new ADP.

To generalise the improved algorithm in future, we need to research the particular shape trajectories, then further realize big data analysis based on the proposed ADP algorithm. Thus, further studies should be conducted to investigate the threshold automatic selection method of special and chaotic trajectories. In addition, the automatic segmentation framework should also be further studied.

REFERENCES

[1] H. Li, J. Liu, K. Wu, Z. Yang, R. W. Liu, and N. Xiong, "Spatio-temporal vessel trajectory clustering based on data mapping and density," *IEEE Access*, vol. 6, pp. 58939–58954, 2018.

[2] Z. Xiao, X. Fu, L. Zhang, and R. S. M. Goh, "Traffic pattern mining and forecasting technologies in maritime traffic service networks: A comprehensive survey," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[3] G. Pallotta, M. Vespe, and K. Bryan, "Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction," *Entropy*, vol. 15, no. 6, pp. 2218–2245, 2013.

[4] R. Al-Zaidi, J. C. Woods, M. Al-Khalidi, and H. Hu, "Building novel VHF-based wireless sensor networks for the Internet of marine things," *IEEE Sensors J.*, vol. 18, no. 5, pp. 2131–2144, Mar. 2018.

[5] K. Patroumpas, E. Alevizos, A. Artikis, M. Vodas, N. Pelekis, and Y. Theodoridis, "Online event recognition from moving vessel trajectories," *GeoInformatica*, vol. 21, no. 2, pp. 389–427, 2017.

[6] Z. Feng and Y. Zhu, "A survey on trajectory data mining: Techniques and applications," *IEEE Access*, vol. 4, pp. 2056–2067, 2017.

[7] Z. Xiao, L. Ponnambalam, X. Fu, and W. Zhang, "Maritime traffic probabilistic forecasting based on vessels' waterway patterns and motion behaviors," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3122–3134, Nov. 2017.

[8] T. R. Hammond and D. J. Peters, "Estimating AIS coverage from received transmissions," *J. Navigat.*, vol. 65, no. 3, pp. 409–425, 2012.

[9] H. Li, J. Liu, R. W. Liu, N. Xiong, K. Wu, and T.-H. Kim, "A dimensionality reduction-based multi-step clustering method for robust vessel trajectory analysis," *Sensors*, vol. 17, no. 8, p. 1792, 2017.

[10] U. Demšar and K. Virrantaus, "Space–time density of trajectories: Exploring spatio-temporal patterns in movement data," *Int. J. Geographical Inf. Sci.*, vol. 24, pp. 1527–1542, Oct. 2010.

[11] Y. Zheng and X. Zhou, *Computing With Spatial Trajectories*. New York, NY, USA: Springer, 2011.

[12] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, 1985.

[13] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," in *Proc. IEEE Int. Conf. Data Mining*, Nov./Dec. 2001, pp. 289–296.

[14] N. Meratnia and A. de Rolf, "Spatiotemporal compression techniques for moving point objects," in *Proc. Int. Conf. Extending Database Technol.*, 2004, pp. 765–782.

[15] W. R. Tobler, "Numerical map generalization, and notes on the analysis of geographical distributions," Dept. Geography, Univ. Michigan, Ann Arbor, MI, USA, 1966.

[16] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica, Int. J. Geographic Inf. Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[17] R. Bellman and B. Kotkin, "On the approximation of curves by line segments using dynamic programming," *Commun. ACM*, vol. 4, p. 284, Feb. 1962.

[18] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Comput. Graph. Image Process.*, vol. 1, no. 3, pp. 244–256, Nov. 1972.

[19] H. Qian and Y. Lu, "Simplifying GPS trajectory data with enhanced spatial-temporal constraints," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 11, p. 329, 2017.

[20] K.-F. Richter, F. Schmid, and P. Laube, "Semantic trajectory compression: Representing urban movement in a nutshell," *J. Spatial Inf. Sci.*, vol. 2012, no. 4, pp. 3–30, 2012.

[21] J. Muckell, P. W. Olsen, Jr., J.-H. Hwang, C. T. Lawson, and S. Ravi, "Compression of trajectory data: A comprehensive evaluation and new approach," *GeoInformatica*, vol. 18, no. 3, pp. 435–460, 2014.

[22] C. K. Cheung and W. Shi, "Positional error modeling for line simplification based on automatic shape similarity analysis in GIS," *Comput. Geosci.*, vol. 32, pp. 462–475, May 2006.

[23] A. Saalfeld, "Topologically consistent line simplification with the Douglas-Peucker algorithm," *Cartogr. Geograph. Inf. Sci.*, vol. 26, no. 1, pp. 7–18, 1999.

[24] M. Bertolotto and M. Zhou, "Efficient and consistent line simplification for Web mapping," *Int. J. Web Eng. Technol.*, vol. 3, no. 2, pp. 139–156, 2007.

[25] J. Gudmundsson, J. Katajainen, D. Merrick, C. Ong, and T. Wolle, "Compressing spatio-temporal trajectories," *Comput. Geometry*, vol. 42, no. 9, pp. 825–841, 2009.

[26] Y. Li, R. W. Liu, J. Liu, Y. Huang, B. Hu, and K. Wang, "Trajectory compression-guided visualization of spatio-temporal AIS vessel density," in *Proc. 8th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2016, pp. 1–5.

[27] J. Muckell, J.-H. Hwang, V. Patil, C. T. Lawson, F. Ping, and S. S. Ravi, "SQUISH: An online approach for GPS trajectory compression," in *Proc. 2nd Int. Conf. Comput. Geospatial Res. Appl.*, Washington, DC, USA, 2011, Art. no. 13.

[28] M. Chen, M. Xu, and P. Franti, "A fast $O(N)$ multiresolution polygonal approximation algorithm for GPS trajectory simplification," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2770–2785, May 2012.
[29] S.-K. Zhang, Z.-J. Liu, Y. Cai, Z.-L. Wu, and G.-Y. Shi, "AIS trajectories simplification and threshold determination," *J. Navigat.*, vol. 69, no. 4, pp. 729–744, 2016.
[30] L. Etienne, T. Devogele, and A. Bouju, "Spatio-temporal trajectory analysis of mobile objects following the same itinerary," *Adv. Geo-Spatial Inf. Sci.*, vol. 10, pp. 47–57, Jul. 2012.
[31] J. L. G. Pallero, "Robust line simplification on the plane," *Comput. Geosci.*, vol. 61, pp. 152–159, Dec. 2013.
[32] J. Birnbaum, H.-C. Meng, J.-H. Hwang, and C. Lawson, "Similarity-based compression of GPS trajectory data," in *Proc. 4th Int. Conf. Comput. Geospatial Res. Appl.*, Jul. 2013, pp. 92–95.
[33] A. Nibali and Z. He, "Trajic: An effective compression system for trajectory data," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3138–3151, Nov. 2015.
[34] L. Zhao and G. Shi, "A trajectory clustering method based on Douglas-Peucker compression and density for marine traffic pattern recognition," *Ocean Eng.*, vol. 172, pp. 456–467, Jan. 2019.
[35] Y. Li, H. Liu, X. Zheng, Y. Han, and L. Li, "A top–bottom clustering algorithm based on crowd trajectories for small group classification," *IEEE Access*, vol. 7, pp. 29679–29698, 2019.
[36] L. Zhao and G. Shi, "A novel similarity measure for clustering vessel trajectories based on dynamic time warping," *J. Navigat.*, vol. 72, no. 2, pp. 290–306, 2019.
[37] G. Andrienko, N. Andrienko, G. Fuchs, and J. M. C. Garcia, "Clustering trajectories by relevant parts for air traffic analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 34–44, Jan. 2018.
[38] D. Zhang, K. Lee, and I. Lee, "Hierarchical trajectory clustering for spatio-temporal periodic pattern mining," *Expert Syst. Appl.*, vol. 92, pp. 1–11, Feb. 2018.
[39] Z. Hong, Y. Chen, and H. S. Mahmassani, "Recognizing network trip patterns using a spatio-temporal vehicle trajectory clustering algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2548–2557, Aug. 2018.
[40] F. K. P. Chan, A. W. C. Fu, and C. Yu, "Haar wavelets for efficient similarity search of time-series: With and without time warping," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 3, pp. 686–705, May 2003.
[41] D. P. Huttenlocher, W. J. Rucklidge, and G. A. Klanderman, "Comparing images using the Hausdorff distance under translation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1992, pp. 654–656.
[42] A. Krogh, B. Larsson, G. von Heijne, and E. L. L. Sonnhammer, "Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes," *J. Mol. Biol.*, vol. 305, no. 3, pp. 567–580, 2001.
[43] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 23, no. 1, pp. 67–72, Feb. 1975.
[44] D. S. Hirschberg, "Algorithms for the longest common subsequence problem," *J. ACM*, vol. 24, no. 4, pp. 664–675, Oct. 1977.
[45] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 312–319.
[46] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, "A review of moving object trajectory clustering algorithms," *Artif. Intell. Rev.*, vol. 47, no. 1, pp. 123–144, 2017.
[47] D. F. Silva, R. Giusti, E. Keogh, and G. E. A. P. A. Batista, "Speeding up similarity search under dynamic time warping by pruning unpromising alignments," *Data Mining Knowl. Discovery*, vol. 32, no. 4, pp. 988–1016, Jul. 2018.

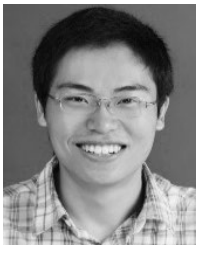
JINGXIAN LIU received the Ph.D. degree from the School of Energy and Power Engineering, Wuhan University of Technology (WUT), Wuhan, China, in 2009. He is currently a Full Professor with the School of Navigation (WUT). His current research interests include intelligent transportation systems, vessel traffic flow, vessel navigation safety, risk assessment, and intelligent traffic organization.

HUANHUAN LI received the M.Sc. degree from the School of Science, Wuhan University of Technology (WUT), Wuhan, China, in 2015, and the Joint Ph.D. Student in maritime and mechanical engineering from Liverpool John Moores University, U.K., in 2019. She is currently pursuing the Ph.D. degree with the School of Navigation, WUT. Her research interests include trajectory visualization, trajectory data mining, maritime transport, computational transportation science, and artificial intelligence.

ZAILI YANG received the B.Eng. degree in maritime transportation from Dalian Maritime University, China, in 2001, the M.Sc. degree in international transport from Cardiff University, U.K., in 2003, and the Ph.D. degree in maritime safety from Liverpool John Moores University (LJMU), U.K., in 2006. He is currently a Professor of maritime transport with LJMU. His research interests include analysis and modeling of safety, resilience, and sustainability of transport networks, especially maritime and logistics systems.

KEFENG WU received the B.Sc. degree from the School of Science, Wuhan University of Technology (WUT), China, in 2014, the M.Sc. degree from the School of Automation, Huazhong University of Science and Technology, Wuhan, China, in 2017. He is currently with the Beijing Electro-Mechanical Engineering Institute. His current research interests include path optimization and machine learning.

YI LIU received the Ph.D. degree from the Department of Civil, Architectural and Environmental Engineering, Illinois Institute of Technology (IIT), Chicago, IL, USA, in 2015. He is currently an Associate Professor with the School of Navigation, Wuhan University of Technology (WUT). His current research interests include vessel traffic flow theory, area wide traffic dynamics, transportation system optimization, and intelligent traffic organization.



RYAN WEN LIU (M'15) received the B.Sc. degree (Hons.) in information and computing science from the Department of Mathematics, Wuhan University of Technology, Wuhan, China, in 2009, and the Ph.D. degree in mathematical imaging from The Chinese University of Hong Kong, Hong Kong, in 2015. He was a Visiting Professor with the Agency for Science, Technology and Research (A∗ STAR), Singapore. He is currently an Associate Professor with the School of Navigation, Wuhan University of Technology. He is also a Visiting Scholar with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His current research interests include mathematical imaging, computer vision, trajectory data mining, and computational navigation sciences.