

Received June 3, 2018, accepted August 1, 2018, date of publication August 21, 2018, date of current version October 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2866364

# Spatio-Temporal Vessel Trajectory Clustering Based on Data Mapping and Density

HUANHUAN LI<sup>1,2,3</sup>, JINGXIAN LIU<sup>1,2</sup>, KEFENG WU<sup>4</sup>, ZAILI YANG<sup>3</sup>,  
RYAN WEN LIU<sup>1,2</sup>, AND NAI XUE XIONG<sup>5</sup>, (Senior Member, IEEE)

<sup>1</sup>Hubei Key Laboratory of Inland Shipping Technology, School of Navigation, Wuhan University of Technology, Wuhan 430063, China

<sup>2</sup>National Engineering Research Center for Water Transport Safety, Wuhan 430063, China

<sup>3</sup>Liverpool Logistics, Offshore and Marine Research Institute, Liverpool John Moores University, Liverpool L3 3AF, U.K.

<sup>4</sup>Beijing Electro-mechanical Engineering Institute, Beijing 100074, China

<sup>5</sup>School of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA

Corresponding authors: Ryan Wen Liu (wenliu@whut.edu.cn) and Naixue Xiong (xionгнаixue@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51479156, Grant 51609195, Grant 51709219, and Grant 51179147 and in part by the China Scholarship Council.

**ABSTRACT** Automatic identification systems (AISs) serve as a complement to radar systems, and they have been installed and widely used onboard ships to identify targets and improve navigational safety based on a very high-frequency data communication scheme. AIS networks have also been constructed to enhance traffic safety and improve management in main harbors. AISs record vessel trajectories, which include rich traffic flow information, and they represent the foundation for identifying locations and analyzing motion features. However, the inclusion of redundant information will reduce the accuracy of trajectory clustering; therefore, trajectory data mining has become an important research direction. To extract useful information with high accuracy and low computational costs, trajectory mapping and clustering methods are combined in this paper to explore big data acquired from AISs. In particular, the merge distance (MD) is used to measure the similarities between different trajectories, and multidimensional scaling (MDS) is adopted to construct a suitable low-dimensional spatial expression of the similarities between trajectories. An improved density-based spatial clustering of applications with noise (DBSCAN) algorithm is then proposed to cluster spatial points to acquire the optimal cluster. A fusion of the MD, MDS, and improved DBSCAN algorithms can identify the course of trajectories and attain a better clustering performance. Experiments are conducted using a real AIS trajectory database for a bridge area waterway and the Mississippi River to verify the effectiveness of the proposed method. The experiments also show that the newly proposed method presents a higher accuracy than classical ones, such as spectral clustering and affinity propagation clustering.

**INDEX TERMS** AIS network, data mapping, DBSCAN, trajectory similarity, trajectory clustering, maritime transport.

## I. INTRODUCTION

As an important type of navigational equipment, radar can be used to aid navigation, supervise ships and monitor vessel traffic flow via a vessel traffic service (VTS). However, identifying individual targets requires different technologies. Very high frequency (VHF) radios are among the most important communication and safety devices for ship operators and managers and can realize boat-to-boat to boat-to-shore communication [1]. Based on a VHF data communication scheme, automatic identification systems (AISs) provide functionalities for SOLAS (AIS Class A) vessels only in limited environments defined by radio propagation

properties. The AIS system is installed and widely used on ships to enhance their ability to identify targets.

The AIS network consists of several AIS base stations or shore ones that are connected and constructed to improve maritime surveillance and traffic management. AISs provide an important complementary data source to radar, especially for large, co-operating ships [2]. The temporal resolution of the AIS signal is commonly enhanced through marine radar (data fusion). The received AIS data are sent to the data fusion system and can then be correlated with raw sensor data, allowing vessel trajectories to be tagged with useful information [3].

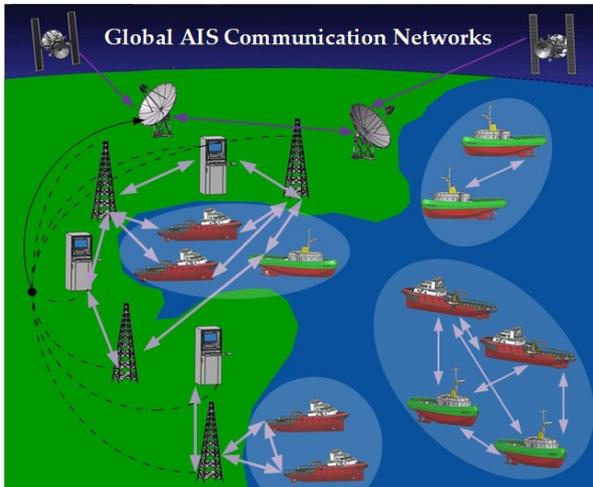


FIGURE 1. The global AIS communication networks.

As a digital communication system, a shipborne AIS is able to send and receive important information. AISs use VHF radios to transmit data, such as basic parameters and cargo information, maritime mobile service identities (MMSIs), accurate positions, routes, speeds, sea gauges, heading statuses and other information. This information can be transmitted to surrounding ships to improve maritime traffic and ensure navigational safety [4]. The global AIS communication networks are shown in Fig. 1.

The International Maritime Organization (IMO) defines an AIS as a ship and shore broadcast system based on wireless information that can be used to assist in identifying ships, tracking targets, exchanging information and avoiding collision between ships [5]. AISs are self-reporting messaging systems originally conceived for collision avoidance via high-speed updates. AISs can process 2,000 reports per minute and update information every two seconds to further guarantee reliable and stable ship-to-ship and ship-to-shore operations. AIS data provide a vast amount of real-time information that can be used to support decision-making and management [6].

Maritime traffic patterns are important for judging maritime conditions, researching navigational features, classifying and predicting ship activities [7]. AIS data can be effectively used to infer different levels of contextual information from characteristics to spatial and temporal distributions of routes [8]. A novel spatio-temporal vessel trajectory clustering method is proposed to extract the maritime movement patterns and further assist in making decisions. The proposed method represents a basic step toward being able to detect anomalies and make projections from current trajectories and patterns for automatic forecasting. Therefore, AIS data research and processing have become a prime focus of scholars [9].

Clustering is among the most important research methods in data mining and is often applied to large point databases, and it can be used to obtain the pattern information of vessels [10], [11]. The clustering process is known as an

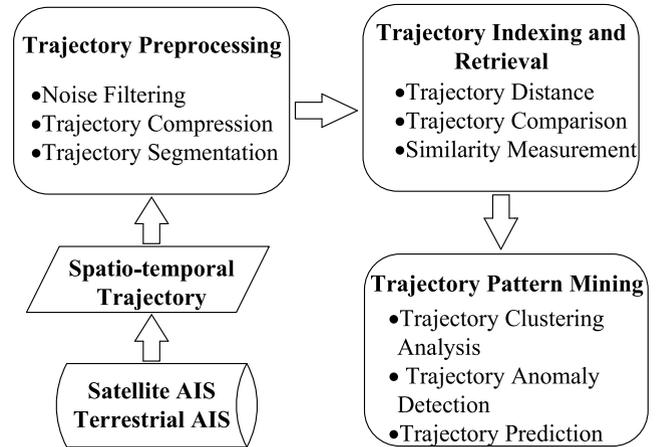


FIGURE 2. Trajectory clustering framework.

unsupervised learning method in which a priori knowledge about the dataset is not available. Many clustering algorithms, such as the K-means [12], density-based spatial clustering of applications with noise (DBSCAN) [13], Ordering Points To Identify the Clustering Structure (OPTICS) [14], and balanced iterative reducing and clustering using hierarchies (BIRCH) [15] algorithms, have been proposed and many still being under further research. However, trajectory data are different from traditional point data and cannot be directly clustered by classical clustering algorithms [16]. Other issues pertaining to trajectory data remain unresolved, including 1) whether to consider the whole trajectory length, 2) how to extract useful traffic flow information, 3) how to choose appropriate points in the trajectory, and 4) how to directly measure the similarity between trajectories [17], [18]. The literature indicates that trajectory clustering is composed of three components, and the framework for trajectory clustering is shown in Fig. 2.

Trajectories consist of many points and are usually not straight lines; therefore, methods of measuring the distances between points cannot be simply used directly to measure the distance between trajectories. The current clustering algorithms are all based on points, which cannot be directly used for trajectory clustering. Therefore, a fusion between multidimensional scaling (MDS) and the improved DBSCAN algorithm is proposed to realize vessel trajectories based on a representation of abstract points. The proposed methodology can clearly differentiate information on different vessel trajectories.

The purpose of the fusion method is to identify traffic flow patterns and customary routes from immense traffic movement trajectories and then to discern abnormal trajectories. To improve the accuracy of the associated calculations, the merge distance (MD) has been applied to measure the similarity between trajectories [19]. In particular, this method is able to effectively calculate the robust similarities between trajectories. When the distances between all trajectories are received, they must be transformed into the distances

between points. As a useful data mapping and dimensional reduction method, MDS is used to solve this problem. Thereafter, the improved DBSCAN is used to cluster abstract points. This fusion approach, using MD, MDS and the improved DBSCAN algorithm, is not sensitive to undesirable noise and can also aid a better clustering performance.

The traffic flow patterns detected by the clustering method from raw trajectory data have provided a solid foundation for further research on trajectory visualization and safety route planning. The remainder of the paper is organized as follows. Section II reviews the main clustering algorithms. Section III introduces the trajectory similarity measurement MD method, the data mapping algorithm, the theory behind MDS and the improved DBSCAN method. Section IV provides a comprehensive description of numerous experimental processes and a complete evaluation analysis of different algorithms in different data sets. Finally, Section V presents the discussion, and Section VI provides conclusions and future work.

## II. BRIEF REVIEW OF CLUSTERING ALGORITHMS

Clustering algorithms are used in various fields, such as pattern recognition, image processing, data mining, statistical analysis, and other business applications [20]. Recently, different types of clustering methods have been proposed and further developed by scholars worldwide. Until now, clustering methods could be roughly divided into six categories: partitioning methods [21] (e.g., K-means and K-medoids), hierarchical methods (e.g., Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH)), density-based methods [22] (e.g., DBSCAN), grid-based methods [23] (e.g., STatistical INformation Grid (STING)), model-based clustering algorithms [24] and fuzzy clustering algorithms [25].

The partition-based clustering algorithm divides data objects into different clusters, and it also demands that each object belongs to a cluster only until the optimal clustering result is obtained [26]. Generally, the criterion used in partitioning is that the within-cluster similarity of objects should be as large as possible and that the between-cluster similarity should be as small as possible. Partition-based clustering methods mainly include the K-means, K-medoids, Clustering Large Applications (CLARA), and Clustering Large Application based upon RANdomized Search (CLARANS) algorithms, etc. K-means and K-medoids need to determine the number of clusters in advance; however, choosing an appropriate value of  $k$  is scientifically not straightforward. The K-medoids algorithm is more robust, having a higher time complexity and is superior to K-means for isolated points. However, it is only suitable for small data sets. Partition-based clustering algorithms always assign points to the nearest cluster; therefore, these algorithms cannot discover non-spherical clusters.

Hierarchy-based clustering algorithms [27] can be further divided into three types: bottom-up or condensation algorithms [28] (e.g., Merging of Adaptive Finite

IntervAls (MAFIA) and ENtropy-based CLUStering (ENCLUS)), top-down or decomposition algorithms [29] (e.g., PROjected CLUStering (PROCLUS) and Oriented projected CLUStering (ORCLUS)) and compound algorithms (e.g., BIRCH and Clustering Using REpresentatives (CURE) [30]). The theory of the bottom-up algorithm is that each point is taken as a single cluster, and the closest two clusters are merged into one cluster. Top-down algorithms are the opposite of the bottom-up algorithms, where all object points are placed in the same class, which is then divided into smaller classes according to predetermined rules. Hierarchy-based clustering algorithms can easily define similarities in distance and the number of clusters do not need to be set in advance; however, these algorithms are time sophisticated, and the results of clustering are likely to be a chain.

The essence of density-based clustering algorithms [31], [32] is to separate high-density areas from the low ones. Density-based algorithms are not equivalent because of their different definitions for high- and low-density areas. The DBSCAN and OPTICS algorithms are typical examples. DBSCAN can discover clusters of arbitrary shapes and handle noise points (outliers) automatically and effectively, and it has two parameters ( $Eps$  and  $MinPts$ ) and involve low time complexity [33]. Density-based clustering algorithms can automatically find the number of clusters, and they are also suitable for clustering unknown and skewed data sets. However, density-based clustering algorithms are not effective when handling data sets with unobvious density differences.

After the original DBSCAN was proposed in 1996, many different DBSCAN enhancements were proposed for clustering studies, such as VDBSCAN (Varied DBSCAN), EDBSCAN (Enhanced DBSCAN), IDBSCAN (Improved DBSCAN), FDBSCAN (Fast DBSCAN), GRIDBSCAN (GRId DBSCAN), KNNDBSCAN (K-Nearest Neighbors DBSCAN), ST-DBSCAN (Spatial-Temporal DBSCAN), GMDBSCAN (Grid and Multi-density DBSCAN) and so on. VDBSCAN and EDBSCAN mainly choose appropriate density threshold values. IDBSCAN reduces the clustering time, but it still requires the users to customize the initial parameters. FDBSCAN reduces the clustering time and improves the accuracy of clustering by the Kernel function. GRIDBSCAN introduces a three-level mechanism to improve the clustering accuracy, but it has high time. KNNDBSCAN can determine the threshold values based on  $k$ . ST-DBSCAN introduces spatial, non-spatial, temporal and density functions to cluster spatial-temporal data sets. GMDBSCAN introduces the local density to cluster the data set, and determines local  $MinPts$  with grid-density. However, it does not discuss the time complexity. The reverse nearest neighbor approaches, such as RECORD, IS-DBSCAN, ISBDBSCAN, and RNN-DBSCAN, detect observation density with the reverse nearest neighbors. All these approaches require a single parameter  $k$ , the number of nearest neighbors to define the density.

Grid-based clustering algorithms [34] mainly include STING, clustering with wavelets [35] (WaveCluster), and CLustering In QUest (CLIQUE). STING has a higher efficiency, and the grid structure is conducive to parallel processing and incremental updates; however, it may reduce the quality and accuracy of the clusters. WaveCluster can effectively manage large data sets, find clusters of arbitrary shapes, and easily identify isolated points; however, it requires a higher condition requiring knowledge of mathematical modeling and is not able to achieve a superior clustering performance for high-dimension data sets. CLIQUE integrates the advantages of density-based and grid clustering methods and divides a space into sparse and dense regions, after which it finds the global distribution pattern of the datasets. Grid-based clustering algorithms are suitable for large data sets; however, they are sensitive to the input parameters, which increases the difficulty of finding an effective method in theory. The problem that needs to be solved is how to select the appropriate number of units to achieve a balance between the data expression and computational complexity. In addition, a large number of grid cells must be generated so that the original spatial information is retained for high spatial dimensions.

Model-based clustering algorithms [36] are robust clustering methods that can automatically obtain the number of clusters through standard statistical methods, and they can reflect the distribution of data points by constructing an effective density function. Model-based clustering methods mainly include statistics-based ones, which assume that data sets are consistent with the basic probability distribution, a typical example of which is COBWEB. COBWEB is a simple and widely used incremental clustering method. CLASSIT is an improved method of COBWEB, and it can perform incremental clustering of numerical attributes and save the corresponding continuous normal distribution for each node. However, CLASSIT and COBWEB are not suitable for large data sets.

Traditional clustering analyses are based on a hard division concept in which each point belongs to only one cluster out of all. However, many fuzzy attributes cannot be measured by strict standards; thus, the relative soft division concept has been proposed. Fuzzy set theory is introduced into the cluster analysis to solve the cluster with the association rule. Subsequently, multiple fuzzy clustering analysis methods [37] have been proposed, such as those based on similarity relations, fuzzy relationships, fuzzy equivalence relations, convex decomposition of data sets, and dynamic programming. These fuzzy clustering algorithms are mathematically sophisticated and hence often time consuming in calculation. Fuzzy clustering algorithms based on objective functions have become the focus of fuzzy clustering because they present advantages in terms of the results and time. The most commonly used method is the fuzzy C-means clustering algorithm.

### III. A NEW SPATIO-TEMPORAL AIS TRAJECTORY CLUSTERING METHOD

#### A. PROPOSED CLUSTERING ALGORITHM

All of the clustering algorithms discussed in the previous section are suitable for point clustering. Each trajectory consists of many points, and trajectory clustering is different from traditional point clustering. Trajectory clustering can extract useful information and is an efficient method for investigating vessel movement patterns. However, clustering methods that can directly cluster the whole trajectory are not currently available.

Each trajectory is composed of a series of points and has a linear or nonlinear structure over time. The trajectories of vessels have different unique characteristics; therefore, trajectory clustering is more complex than traditional point clustering. The literature indicates that two strategies are primarily used to research trajectory clustering: one method considers the trajectory as a whole and the other partitions the trajectory into a set of line segments [38].

Although a number of studies have considered the trajectory as a whole [39], [40], developing a method for selecting the most suitable similarity measurement method is still a critical problem. Moreover, developing a selection method for the number of cluster centers and a clustering algorithm are also to be explored.

Partitioning methods can divide a trajectory into similar sub-trajectories based on geometric features and structural similarities. Then, the sub-trajectories can be clustered by traditional clustering methods. Many scholars have proposed methods of clustering small sections of a trajectory [41], [42]. However, the development of methods to determine the length of a sub-trajectory and selection of the critical points to represent complex solutions remain to be found.

To extract more useful information from vessel trajectories with a high accuracy and low computational costs, we propose a novel clustering method for AIS data exploration. In this paper, each trajectory is regarded as a whole, and the spatio-temporal trajectory is mapped into a low-dimensional spatial representation of points via MDS. Then, commonly used clustering algorithms can be used to handle these points, and complex trajectory clustering is transformed into a point clustering problem. This method can solve the difficulty of trajectory clustering and greatly reduce the time complexity without affecting the clustering effect. The purpose of this paper is to identify customary routes and discern abnormal trajectories based on data mapping methods and density clustering algorithms.

To improve the efficiency and effectiveness of trajectory clustering, a novel fusion trajectory-clustering algorithm is proposed. The distance between trajectories is measured via MD, which is a robust similarity measurement method used to measure the similarity between trajectories to obtain a distance matrix. Then, suitable data mapping and dimensional reduction methods need to be selected. MDS can construct a suitable low-dimensional space to obtain the spatial

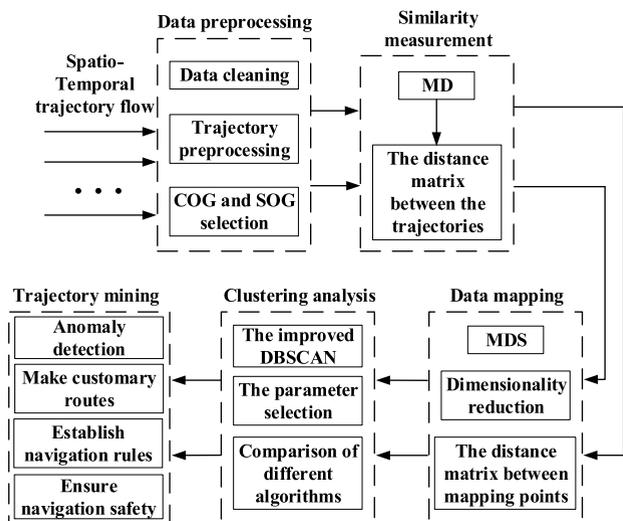


FIGURE 3. Flowchart of the proposed algorithm.

point expression of the similarity between trajectories. MDS is a classical data visualization and nonlinear data mapping method that can transform trajectory data into a low-dimensional point representation. The spatial representation of trajectories can be mapped into the low-dimensional representation of points, which reduces both the dimensionality and time complexity. After the spatial point expression is received, the most suitable clustering algorithm must be selected. DBSCAN is a representative density-based spatial clustering algorithm that can find clusters of arbitrary shapes and remove noisy data. Therefore, the improved DBSCAN algorithm is used to cluster spatial points to acquire the optimal clustering results. The fusion among MD, MDS and the improved DBSCAN algorithm is not sensitive to undesirable noise, and it can also achieve a better clustering performance.

The traffic flow patterns detected by the fusion method from the original trajectory data represent fundamental knowledge for further traffic flow analysis and decision making on traffic management. Experiments on real ordinary AIS trajectories in a bridge area waterway showed that our proposed method was more efficient when discovering traffic flow patterns and discerning customary routes.

The flowchart of the proposed algorithm is shown in Fig. 3.

**B. TRAJECTORY SIMILARITY MEASUREMENT**

Most of the trajectories describing moving objects are curves, and trajectory similarity measurements are mainly dependent on the distance between trajectories. The Euclidean distance represents the straight-line distance between two points, and it requires an equivalent number of points in different trajectories. Therefore, this method is not suitable for measuring the similarity of trajectories. Dynamic time warping (DTW) is an algorithm for measuring the similarity between two time series, which may differ in time or speed. The essence of

DTW is to warp the routes from feature to feature. However, DTW is not robust for subsampling and supersampling.

As a trajectory similarity measurement algorithm with strong robustness under subsampling and supersampling, MD represents the shortest merged distance between two trajectories and can be calculated in quadratic time [19]. Suppose that the trajectory  $p$  is composed of a sequence of time points  $(x_i, y_i, t_i)$ ,  $p_i = (x_i, y_i)$ , where  $p_i$  is the position coordinate point and  $t_i$ ,  $t_1 < \dots < t_n$  represents the corresponding time in a chronological order. Thus, the length of trajectory  $p$  is  $l(p) = \sum_i d(p_i, p_{i+1})$ , where  $d(p_i, p_{i+1})$  represents the distance between point  $p_i$  and  $p_{i+1}$ .

MD denotes the length of the shortest trajectory that is a supersequence of two trajectories. The detailed calculation process is as follows.

Suppose that  $a = \{a_1, \dots, a_m\}$ ,  $i = 1, \dots, m$  and  $b = \{b_1, \dots, b_n\}$ ,  $j = 1, \dots, n$  are trajectories and  $A_i^j (B_j^i)$  is the length of the shortest trajectory that is a supersequence of  $a$  and  $b$ , then

$$A_i^j = \begin{cases} \left( \sum_{k=1}^{j-1} d(b_k, b_{k+1}) \right) + d(b_j, a_1), & i = 1, 1 \leq j \leq n \\ \min \left( A_{i-1}^j + d(a_{i-1}, a_i), B_{i-1}^j + d(b_j, a_i) \right), & 2 \leq i \leq m, 1 \leq j \leq n \end{cases} \quad (1)$$

$$B_j^i = \begin{cases} \left( \sum_{k=1}^{i-1} d(a_k, a_{k+1}) \right) + d(a_i, b_1), & j = 1, 1 \leq i \leq m \\ \min \left( A_i^{j-1} + d(a_i, b_j), B_i^{j-1} + d(b_{j-1}, b_j) \right), & 2 \leq j \leq n, 1 \leq i \leq m \end{cases} \quad (2)$$

The time complexity of MD is  $O(mn)$ , and the length of the shortest supertrajectory is  $l(a, b) = \min(A_m^n, B_m^n)$ . Then,  $MD(a, b)$  is as follows.

$$MD(a, b) = \frac{l(a, b)}{(l(a) + l(b))/2} = \frac{2l(a, b)}{l(a) + l(b)}. \quad (3)$$

MD is invariant under rigid motions [18]. Thus, MD has great application potential for the similarity measurement of the AIS trajectory distance.

**C. MULTIDIMENSIONAL SCALING**

MDS is a classical nonlinear data mapping method and an important data visualization method. It can proportionally scale the relationships between all samples defined in a multidimensional space to a 2D or 3D space. Then, the similarity matrix between trajectories can be represented as the distance of points in two dimensions. The problem is to determine how to reconstruct their Euclidean coordinates when only the similarity matrix between objects is known. For example, if only the distance between each pair of many cities in a country is known but their latitude and longitude information is unknown, then we can express their locations as a 2D coordinate with MDS [43].

MDS mainly includes three types: classical MDS, metric MDS and non-metric MDS [44], [45]. Classical MDS is also called principle coordinates analysis, and it is used to solve the spatial representation of points when only the distance matrix between all points is known. Metric MDS regards the sample distance or (dis)similarity as a quantitative measurement, which ensures that the representation in low-dimensional space maintains this metric relation as far as possible. Non-metric MDS is also called order ordinary scaling, and it considers the sample distance or (dis)similarity relation as a qualitative relationship. Then, the representation in low-dimensional space only needs to maintain the order of this relationship.

Therefore, metric MDS is selected to transform the similarity measurement matrix into a quantitative representation in 2D. The MDS procedure is described as follows.

Find a set of points  $X$  that has the same distance as  $D$  based on the Euclidean constraints. To solve  $X$ , the matrix  $T$  is introduced, and  $T = XX^T$ .

*Step 1.*  $T$  is calculated from  $D$ , and it represents a positive semidefinite matrix, where  $d_{ij} \in D$ ;  $x_i, x_j \in X$ ;  $t_{ij} \in T$ . The average value of  $X$  is 0.

$$\begin{aligned}
 d_{ij}^2 &= (x_i - x_j)^2 = x_i^2 + x_j^2 - 2x_i x_j; \\
 t_{ij} &= x_i x_j \Rightarrow t_{ij} = -\frac{1}{2}(d_{ij}^2 - x_i^2 - x_j^2) \\
 \sum_j d_{ij}^2 &= n x_i^2 + \sum_j x_j^2 - 2x_i \sum_j x_j = n x_i^2 + \sum_j x_j^2; \\
 \sum_i d_{ij}^2 &= n x_j^2 + \sum_i x_i^2 - 2x_i \sum_i x_i = n x_j^2 + \sum_i x_i^2 \\
 \sum_{ij} d_{ij}^2 &= n \sum_i x_i^2 + n \sum_j x_j^2; \\
 t_{ij} &= -\frac{1}{2}(d_{ij}^2 - \frac{1}{n} \sum_k d_{ik}^2 - \frac{1}{n} \sum_k d_{kj}^2 + \frac{1}{n^2} \sum_{k,l} d_{kl}^2)
 \end{aligned} \tag{4}$$

*Step 2.*  $X$  is solved based on  $T$ , and then the eigenvector decomposition is used to construct the matrix  $X$ .

$T = U \Lambda U^T = U \Lambda^{1/2} \Lambda^{1/2} U^T = XX^T$ , where the matrix of eigenvalues is  $U$ ,  $\Lambda$  is the vector of eigenvalues, and  $X = U \Lambda^{1/2}$ .

MDS is able to transform the similarity matrix into the relative distance representation of spatial points. A greater input similarity corresponds to a closer (smaller) relative distance between points, and vice versa. The trajectories are mapped into points, and the time complexity is greatly reduced. For example, the 187 trajectories composed of 29015 points are mapped into 187 points with MDS. When the abstract point representation is received, the most suitable point clustering algorithm should be chosen to solve the clustering problem.

#### D. IMPROVED DBSCAN

Clustering analysis is a common data mining technique, which aims to group a set of points into several clusters so that points in the same cluster are more similar to each other

than to those in other clusters. DBSCAN can find clusters of arbitrary shapes and remove noisy data; it needs to search only once to obtain the final clustering result, and thus, it has a higher efficiency. The main concept underlying DBSCAN is that for each point in a cluster, the neighborhood of a given radius has to contain at least the minimum number of points.

The essence of density-based clustering algorithms is that the density of the points in the same cluster is larger than that of the points in different clusters [46]. The density in the area of noise is lower than the density of any other clusters. To find a cluster, DBSCAN starts searching at an arbitrary object  $p$  at all research points and retrieves all points that are density-reachable from  $p$  with respect to  $Eps$  and  $MinPts$ . If  $p$  is a core object, then the neighborhood of  $p$  can be obtained with respect to  $Eps$  and  $MinPts$ , where  $p$  and the points in the neighborhood of  $p$  belong to a cluster. These points become seeds in the next circle to expand the cluster, and this procedure generates a cluster with respect to  $Eps$  and  $MinPts$ . If  $p$  is a border point and no points are density-reachable from  $p$ ,  $p$  is temporarily assigned to noise. Then, DBSCAN handles the next point in the database.

DBSCAN [47] requires the global parameters  $Eps$  and  $MinPts$  to be set in advance. The original DBSCAN algorithm selects the optimal parameters  $Eps$  and  $MinPts$  via only the  $k$ -nearest neighbor of each point. The improved DBSCAN algorithm, in contrast, chooses the optimal  $Eps$  and  $MinPts$  based on the  $k^{th}$  distance curves and the first derivative of the  $k^{th}$  distance curves.

Each row in the distance matrix  $D$  is sorted in an ascending order; then, the new distance matrix  $S$  is received. The  $k^{th}$  distance curve is the ascending order graph of the column  $k$  in  $S$ . The  $k^{th}$  distance curve is one of the most important parameter selection criteria. The first derivative of the  $k^{th}$  distance curve is the distance difference graph. The distance differences between the adjacent distances of the column  $k$  in  $S$  are computed and sorted in an ascending order. The sorted  $k^{th}$  distance values of every point and the sorted distance difference are visualized, and then  $Eps$  and  $MinPts$  are quickly and accurately calculated. The y-coordinate value of the inflection point is the optimal  $Eps$ , and the  $k$  value is the optimal  $MinPts$ . The comparative analysis results of different  $Eps$  and  $MinPts$  values are displayed in this paper, and the optimal parameters  $Eps$  and  $MinPts$  are chosen.

The pseudocode of the improved DBSCAN is described as follows.

#### IV. EXPERIMENTS AND EVALUATION

The development of bridges in navigational routes (e.g. waterways) is fast, and the inland river navigational environment also changes with the development of bridges, which creates certain restrictions and adverse effects for the navigation of inland waterway vessels. The hydrodynamic interaction between a bridge and a vessel makes the waterway beneath a bridge to be a high-risk area. The bridge across a channel may represent an obstacle to a ship, and ship-bridge collisions are often catastrophic. Therefore, vessel

**Algorithm 1** Improved DBSCAN

```

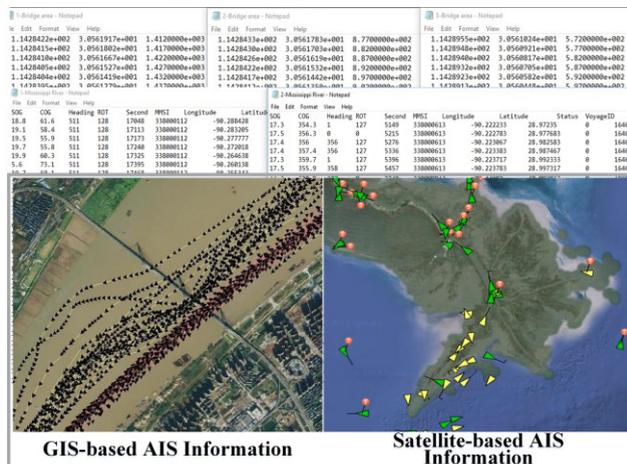
Input:  $D(i, j)$ ,  $i = 1, \dots, m$ ;  $j = 1, \dots, n$ 
        //the distance matrix between the trajectories;
         $MP$  // the mapping data set;
Output: the clustering results.

// Optimal parameter selection. //
Step 1.  $S(\bar{i}, \bar{j})$ ,  $\bar{i} = 1, \dots, m$ ;  $\bar{j} = 1, \dots, n$  is received.
        // the ascending order of each row in  $D$ ;
Step 2.  $T(\bar{i}, \bar{j})$ ,  $\bar{i} = 1, \dots, m$ ;  $\bar{j} = 1, \dots, n$  is received.
        // the ascending order of each column in  $S$ ;
Step 3.  $Q(\bar{i}, k') = T(\bar{i}, k' + 1) - T(\bar{i}, k')$ ,  $\bar{i} = 1, \dots, m$ ,
         $k' = 1, \dots, n - 1$ .
        // the first derivative of the sorted  $k^{th}$  distance values;
Step 4.  $Q(\bar{i}, k'')$ ,  $\bar{i} = 1, \dots, m$ ,  $k'' = 1, \dots, n - 1$ .
        // the ascending order of the first derivative;
Step 5. The comparison curves of  $T(\bar{i}, \bar{k})$ ,  $\bar{k} = 3, \dots, n$  and
         $Q(\bar{i}, k'')$ ,  $k'' = 3, \dots, n - 1$  are displayed and compared.
        // visualize the parameter value images;
Step 6. Select the optimal  $Eps$  and  $MinPts$ .
        // Mark all points as the core points, boundary points
        and noise points. //
Step 7. Choose an arbitrary point  $p \in X$ ,
        IF  $N_{Eps}(p) \geq MinPts$ 
             $N_{Eps}(p) = \{q \in O, dist(p, q) \leq Eps, p \neq q\}$ 
            THEN find all the points that are density-reachable
            from  $p$  to form a cluster;
        ELSE
            THEN exit this cycle, and select the next point;
        Repeat
        UNTIL all points are processed.
Step 8. Delete the noise points.
Step 9. Connect the core points of each group to form a
        cluster.
Step 10. Assign each boundary point to the associated cluster
        of core points.
    
```

trajectory clustering and visualization in bridge areas are areas of focus and are performed to discover customary routes and detect abnormal behavior to prevent accidents and reduce the accident rate.

Yangtze River presents the world busiest inland waterway in the world. The Mississippi River is the largest river in the United States, which has convenient and cheap navigational resources, abundant mineral resources and the unique agricultural resources. Therefore, the complex traffic flow characteristics and changeable environment conditions make the Yangtze River and Mississippi River the focus of this research.

To guarantee the safety of ships sailing in a bridge area waterway or in other complex waters, research on the Wuhan section of the Yangtze River and the Mississippi River were analyzed in this paper. The visualization of original AIS trajectories was first realized in the experiments; then, trajectory clustering analysis of three clustering algorithms was



**FIGURE 4.** Visualization of different types of AIS information: (a) Some original data samples; and (b) GIS-based and Satellite-based AIS information.

conducted for the different datasets to verify the effectiveness of the proposed method. Thus, the obtained clustering results facilitated the identification of vessel traffic flow patterns and customary routes.

**A. EXPERIMENTAL SETUP**

All numerical experiments were performed on a computer running 64-bit Windows 10 with a 2.60 GHz Intel Core i7-5600 CPU and 8 GB memory. We implemented the data mapping and clustering methods using MATLAB R2016a.

Extensive experimental analyses on AIS trajectories were carried out to verify the effectiveness of our proposed method. The AIS trajectory data in a bridge area waterway were collected from the AIS base station in the Wuhan section of the Yangtze River. The bridge area waterway datasets include 187 vessel trajectories with 29,015 points. The data set in Mississippi River includes 106 AIS trajectories, which is composed of 2,442 points. The visualization of different types of AIS information is shown in Fig.4.

$OP$  represents the original data set, which consists of different trajectories, and  $MP$  represents the mapping data set. The experimental procedure is as follows.

*Step 1.* Perform original data cleaning and trajectory pre-processing in  $OP$ .

*Step 2.* Calculate the similarity measurement of AIS trajectories. The distance matrix between trajectories is calculated by MD using Microsoft Visual Studio 2010.

*Step 3.* Determine the spatial point representation of trajectories based on the distance matrix is assessed via MDS.

*Step 4.* Perform an optimal parameter selection and clustering analysis. The points are clustered by the improved DBSCAN, and the optimal  $Eps$  and  $MinPts$  parameters are chosen. The clustering performance is based on the distance of trajectories, and the points are compared with each other to further verify the effectiveness of the proposed method.

*Step 5.* Compare the different clustering algorithms in different data sets. The proposed clustering method is compared with spectral clustering and affinity propagation clustering methods using the specified clusters to further verify the effectiveness of the proposed algorithm.

*Step 6.* Identify the customary routes and abnormal trajectories based on the clustering analysis.

### B. COMPARISONS WITH OTHER CLUSTERING METHODS

Spectral clustering is based on spectral graph partition theory, which is based on the concept of transforming a data clustering problem into an optimal partition problem. Spectral clustering can divide a graph into several subgraphs that do not intersect. The similarity is the highest in the same subgraph and the lowest between different subgraphs [48]. Spectral clustering can identify a sample space with an arbitrary shape and converge to the global optimal solution; however, it is not suitable for datasets with many clusters. The basic idea of spectral clustering is to cluster the received feature vectors according to the similarity matrix of the sample data. However, spectral clustering is sensitive to the scale parameters. The eigenvalue decomposition problem of a large-scale matrix requires enormous computational costs and an immense storage capacity.

Affinity propagation clustering [49] is a semi-supervised clustering algorithm, and the number of clusters  $k$  in affinity propagation clustering does not need to be determined in advance. The algorithm can automatically generate the appropriate number of clusters during the iterative process. The algorithm does not have a special requirement for the similarity matrix, which can be either symmetric or asymmetric [50]. The results of affinity propagation clustering are all equivalent when the iteration is repeated, although the algorithm has a high computational complexity. However, affinity propagation clustering with a specific number of clusters can automatically generate an appropriate clustering result.

### C. OPTIMAL PARAMETER SELECTION

DBSCAN is sensitive to the radius  $Eps$  and the number of points  $MinPts$ , and thus, parameter optimization is the research focus. The literature indicates that two strategies are primarily used to determine the parameters: selecting the optimal parameters automatically and testing certain values manually in a predefined parameter range. Then, the appropriate parameters and satisfactory performance will be selected and fixed. The performance of different values of  $Eps$  and  $MinPts$  is compared based on the sorted  $k^{th}$  distance of each point and the sorted distance difference map; then, the optimal parameters  $Eps$  and  $MinPts$  are selected.

The  $Eps$  and  $MinPts$  are important parameters that influence the clustering performance. Experiments were first carried out to determine the optimal  $MinPts$  value, and then the optimal  $Eps$  was received from the  $k^{th}$  neighbor distance graph. The two parameters have a significant impact on the experimental results, and the proper parameters with

satisfactory performance will be selected and fixed throughout all experiments. The optimal parameter selection results based on trajectories and mapping points were compared in the experiments. To obtain more accurate results, the distance matrix between points was chosen to find the optimal parameter in this paper.

The optimal parameter selection process is designed as follows.

*Step 1.* Visualize the sorted  $k^{th}$  distance graph.

*Step 2.* Visualize the sorted distance difference graph.

*Step 3.* Compare the results of the different sorted  $k^{th}$  distance graphs and distance difference graphs and select the best  $Eps$  and  $MinPts$ .

*Step 4.* Select the optimal parameters.

The best  $k$  value is  $MinPts$ , and the corresponding inflection point is the best value of  $Eps$ .

### D. VISUALIZATION OF CLUSTERING RESULTS IN A BRIDGE AREA WATERWAY

The clustering performance of different algorithms is visualized and analyzed in detail. The experimental process conducted in the bridge area waterway is described in detail as follows. Data cleansing is the basic step of trajectory visualization, and it can delete erroneous data and repair incomplete data. The original trajectories are first judged by the trajectory acquisition time and time interval, and then the incomplete and invalid trajectory data are deleted.

After data cleansing, 161 vessel trajectories with 25678 coordinate points are preserved. Four trajectories among the 161 trajectories have an opposite course, and the points in the four trajectories have a reverse order compared with those in other trajectories. Therefore, the distances between the four trajectories and the other 157 trajectories are perhaps larger. The experiment does not indicate the directions of the trajectories, and thus, the proposed method can be tested to determine whether it can identify the reverse trajectories.

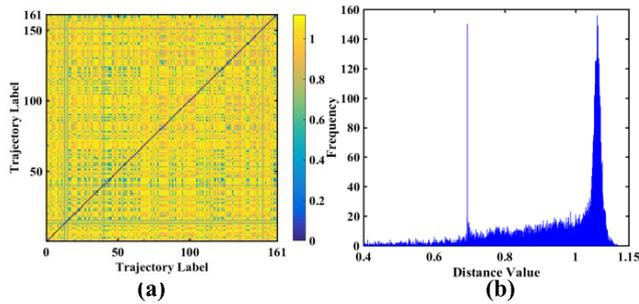
#### 1) VISUALIZATION OF THE SIMILARITY MATRIX

MD was implemented to calculate the distances between the 161 trajectories, and then the distance matrix was calculated. The 2D visual display of the distance matrix and the statistical histogram of all distances are shown in Fig. 5.

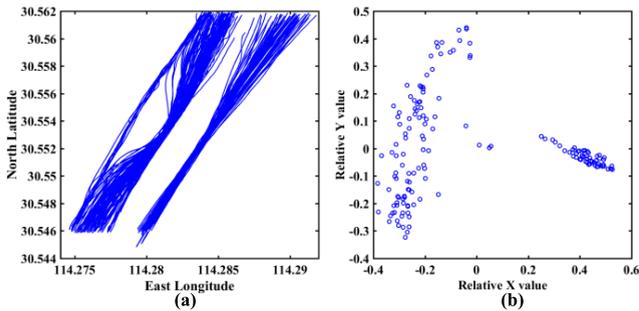
As shown in Fig. 5 (a), the X-axis and Y-axis represent trajectory labels, and the different colors express different values. The 2D visualization of the distance matrix clearly shows the symmetry of the distance matrix and the distribution characteristics of the distances. As shown in Fig. 5 (b), the X-axis represents the distance value and the Y-axis represents the frequency of the distances, and the distribution of the distances can be further displayed.

#### 2) VISUALIZATION OF ORIGINAL TRAJECTORIES AND AFTER MDS

The original AIS trajectories and the spatial point representation of the trajectories based on MDS are shown in Fig. 6.



**FIGURE 5.** Image display of the distance matrix with MD: (a) 2D display of the distance matrix; and (b) statistical histogram of all distances between trajectories.

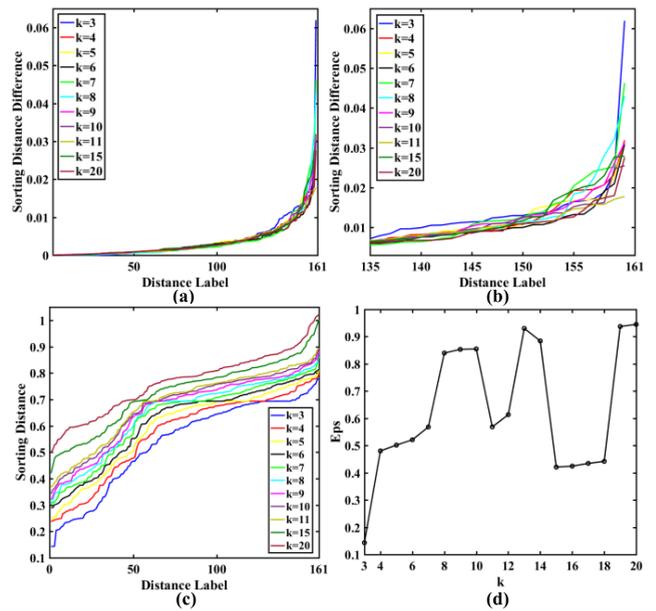


**FIGURE 6.** Original trajectory and data mapping results: (a) original AIS trajectories; and (b) spatial point representation with MDS.

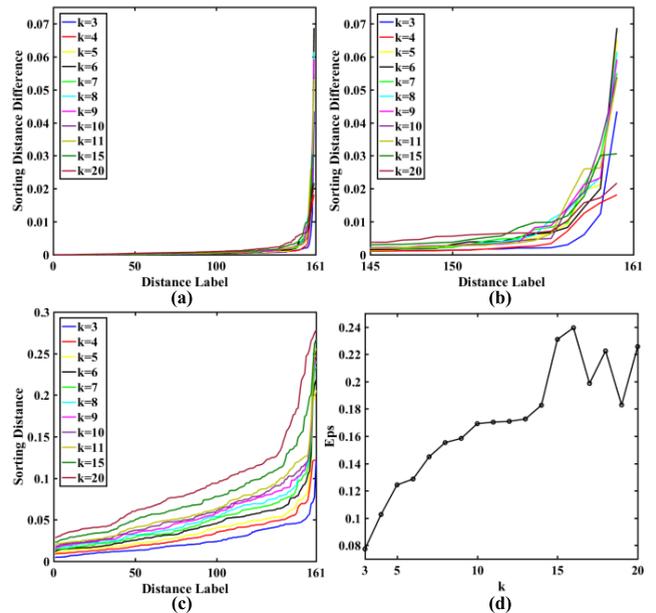
The visual display of the 161 trajectories in the bridge area is shown in Fig. 6 (a), which displays three bridge openings and two bridge piers. The vessel trajectories converge together before entering the bridge opening and after leaving the bridge opening in the left trajectories. The spatial point representation based on MDS is shown in Fig. 6 (b), and the relationship between the points is easily observed. As shown in Fig. 6 (b), the trajectories are mapped into the points in the 2D plane. Then, the trajectory clustering problem is transformed into a point clustering problem. The distance between the points is expressed by the Euclidean distance, which can greatly reduce the time cost and computation complexity. The original trajectories consist of 25678 coordinate points, and MDS mapping reduces this number to only 161 points. Therefore, MDS represents a useful data mapping and dimensional reduction method.

### 3) VISUALIZATION OF THE OPTIMAL PARAMETER SELECTIONS

The two *Eps* and *MinPts* parameters in the improved DBSCAN are chosen from the sorted  $k^{th}$  distance graph of each point and the distance difference map, and the sorted  $k^{th}$  distance can clearly show the growth trend. The maximum value at the inflection point is the optimal value, and  $MinPts = the\ optimal\ k$ . To verify the effectiveness and superiority of the proposed method, the different comparison results of the optimal parameter selections between the original distance matrix based on trajectories and the distance matrix based on mapping points are shown in Figs. 7 and 8.



**FIGURE 7.** Optimal parameter selection of the original distance matrix based on trajectories: (a) ascending order graph of the distance difference for different column  $k$  values; (b) locally enlarged image of (a); (c) ascending order graph of different column  $k$  values; and (d) *Eps* value for different  $k$  values.



**FIGURE 8.** Optimal parameter selection of the distance matrix based on mapping points: (a) ascending order graph of the distance difference for different column  $k$  values; (b) locally enlarged image of (a); (c) ascending order graph of different column  $k$  values; and (d) *Eps* for different  $k$  values.

The ascending order graph of the distance differences for different column  $k$  values ( $k = 3, 4, 5, 6, 7, 8, 9, 10, 11, 15,$  and  $20$ ) is shown in Fig. 7 (a), in which the overall trend is basically the same and that all curves have an inflection point. To visualize the differences more clearly, a locally enlarged image of Fig. 7 (a) is displayed in Fig. 7 (b). The ascending order graph of the different column  $k$  values is

shown in Fig. 7 (c), and the overall trend is clear. The column  $k$  values in the distance matrix of the trajectory represent the sorted  $k^{\text{th}}$  distance of every point. The  $Eps$  values at different  $k$  values are shown in Fig. 7 (d), which shows that the  $Eps$  values vary greatly.

The comparative results of  $k$  are clearly shown in Figs. 7 (b) and (c), and the performance of different  $k$  values is displayed in detail. From the overall trend and the curves of  $k = 3, 4, 5, 6, 7, 8, 9, 10, 11, 15,$  and  $20$  in Figs. 7 (b) and (c), we can see that the biggest change occurs when  $k = 4$  and that the inflection point appears at the earliest when  $k = 4$ . Then, the optimal value of  $MinPts$  is 4. The corresponding value of the inflection point when  $MinPts = 4$  is the optimal value of  $Eps$ ; thus,  $Eps = 0.4803$ . Thereafter, the improved DBSCAN can cluster the points with  $Eps = 0.4803$  and  $MinPts = 4$ . However, the trend of the sorted  $k^{\text{th}}$  distance of every point is not very clear, and two inflection points are shown in Fig. 7 (c). The same optimal parameter selection process of the distance matrix based on mapping points is shown in Fig. 8, which can be used to further verify the effectiveness of the proposed method. A comparison of Figs. 7 and 8 shows that the graph of different  $k$  values in Fig. 8 has a more obvious trend and that the different  $Eps$  have more stable values.

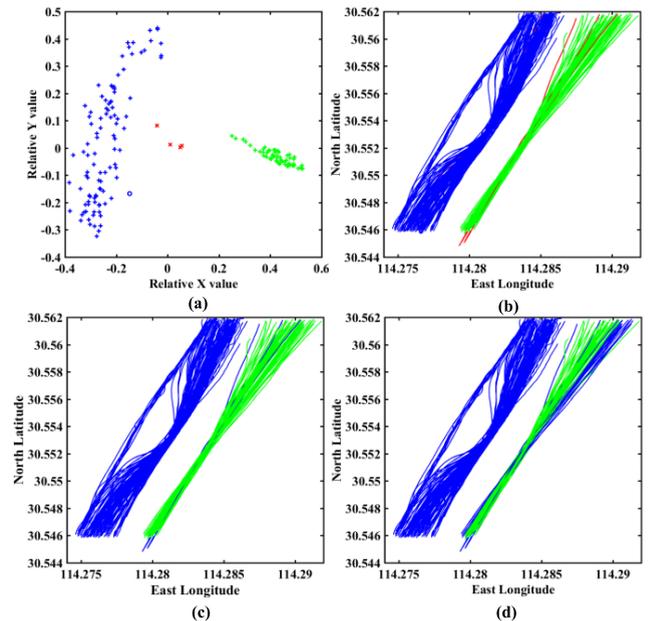
The sorted  $k^{\text{th}}$  distance difference is shown in Fig. 8 (a), and the overall trend is basically the same; all of the curves have an inflection point. To visualize the difference between each more clearly, a locally enlarged image is displayed in Fig. 8 (b). The ascending order graph of the  $k^{\text{th}}$  distance is shown in Fig. 8 (c), and the overall trend is clear. Only one inflection point is shown in Fig. 8 (c). The comparative results of different  $k$  values are clearly shown in Figs. 8 (b) and (c). The  $Eps$  values at different  $k$  are shown in Fig. 8 (d), which shows that the change in  $Eps$  is relatively stable.

The overall trend and the curves of  $k = 3, 4, 5, 6, 7, 8, 9, 10, 11, 15,$  and  $20$  in Figs. 8 (b) and (c) show that the biggest change occurs and the inflection point appears at the earliest when  $k = 4$ . The variation trend and the overall performance in Figs. 8 (b) and (c) are better than those in Figs. 7 (b) and (c). Thus, the optimal value of  $MinPts$  is 4. The corresponding value of the inflection point when  $MinPts = 4$  is the optimal value of  $Eps$ ; thus,  $Eps = 0.1026$ . Thereafter, the improved DBSCAN can cluster the points with  $Eps = 0.1026$  and  $MinPts = 4$ .

The comparison results of the optimal parameter selection are shown in Figs. 7 and 8, which indicate that the distance matrix based on mapping points presents a superior performance. Therefore, the proposed parameter selection method has improved performance.

#### 4) VISUALIZATION OF CLUSTERING RESULTS AND COMPARISON OF DIFFERENT ALGORITHMS

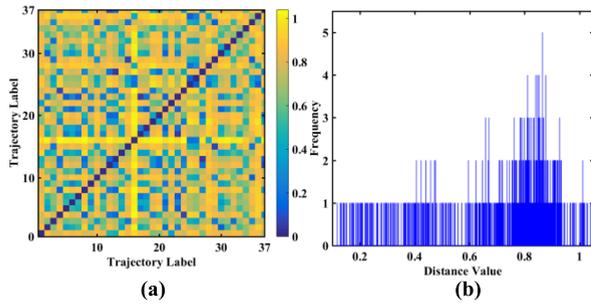
The proposed algorithm is compared with spectral clustering and affinity propagation clustering using a specific cluster number, and the experimental comparison results of the three algorithms are shown in Fig. 9.



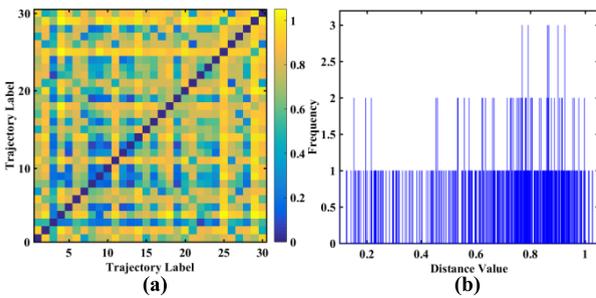
**FIGURE 9.** Clustering results of the different algorithms: (a) point clustering results of the improved DBSCAN when  $Eps = 0.1026$  and  $MinPts = 4$  in the mapping data set  $MP$ ; (b) corresponding trajectory clustering results of (a), where the red trajectories represent the opposite course; (c) clustering results of spectral clustering when the cluster center is 2; and (d) clustering results of affinity propagation clustering when the cluster center is 2.

The point clustering result and the corresponding trajectory clustering results of the improved DBSCAN with  $Eps = 0.1026$  and  $MinPts = 4$  are shown in Figs. 9 (a) and (b). The point clustering results of the improved DBSCAN are shown in Fig. 9 (a), in which the blue and green plus signs represent different clusters and the red points denote noise. The corresponding trajectory clustering results of the point clustering results are shown in Fig. 9 (b), in which the blue and green trajectories denote different clusters, and the red trajectories represent abnormal trajectories. Further analysis of the four red trajectories reveals that they are the reverse trajectories. Compared with the other trajectories, the red trajectories have an opposite direction. Although the experiment does initially not indicate the course of the trajectories, the proposed method could still discern the trajectories with an opposite course.

The clustering results of the spectral clustering method are shown in Fig. 9 (c), which indicates that the three reverse trajectories are misclassified. The spectral clustering method cannot identify the reverse trajectories. The clustering results of the affinity propagation clustering method are shown in Fig. 9 (d), in which blue and green also represent different clusters. However, the green trajectories among blue trajectories are incorrect clustering results and do not represent reverse trajectories. Therefore, the proposed method can identify abnormal trajectories automatically and detect the course of trajectories. The experiments also show that the performance of the proposed method is better than those of the affinity propagation clustering and spectral clustering algorithms.



**FIGURE 10.** Image display of distance matrix with MD of 37 up-bound trajectories: (a) 2D display of the distance matrix; and (b) statistical histogram of all distances between trajectories.



**FIGURE 11.** Image display of distance matrix with MD of 30 down-bound trajectories: (a) 2D display of the distance matrix; and (b) statistical histogram of all distances between trajectories.

**E. VISUALIZATION OF CLUSTERING RESULTS IN MISSISSIPPI RIVER**

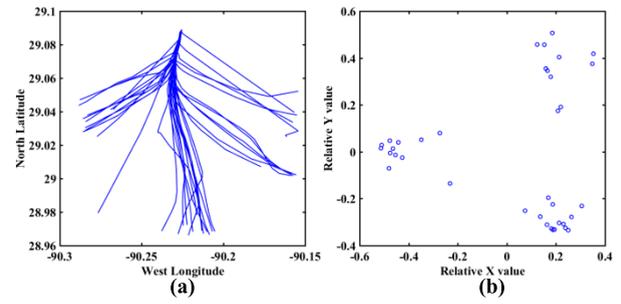
The second experiment data set was associated with the Mississippi River. The 106 trajectories contain 2,442 points. Data cleansing is conducted to remove the error data and repair incomplete data. Then 67 trajectories with 1,532 coordinate points are preserved after data cleansing. There are 37 trajectories of up-bound vessels and 30 trajectories of down-bound vessels, and they have the opposite navigation directions.

**1) VISUALIZATION OF THE SIMILARITY MATRIX**

Fig.10 shows the 2D visual display of the  $37 \times 37$  dimensional distance matrix and the statistical histogram of all distances.

As shown in Fig. 10(a), the X-axis and Y-axis represent the trajectory labels, and the different color expresses different values. The 2D display can clearly show the symmetry of the distance matrix and the distribution characteristics of the distances. In Fig. 10(b), the X-axis represents the distance value and the Y-axis represents the frequency of all the distances. The statistical histogram of all the distances is conducive to observing the distribution of the distances.

The 2D visual display of the  $30 \times 30$  distance matrix and the statistical histogram of all distances are shown in Fig. 11. In Fig. 11(a), the X-axis and Y-axis represent the trajectory labels, and 2D display can clearly show the symmetry of the distance matrix. The X-axis represents the distance value and the Y-axis represents the frequency of all the distances



**FIGURE 12.** Original trajectory and data mapping results: (a) Original AIS trajectory display of 37 up-bound vessels; and (b) spatial point representation with MDS.

in Fig. 11(b). The statistical histogram can clearly shows the distribution of the distances.

**2) VISUALIZATION OF ORIGINAL TRAJECTORIES AND AFTER MDS**

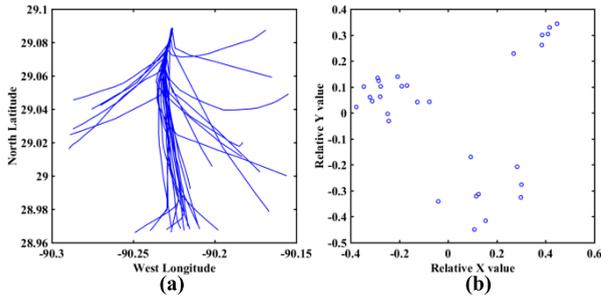
The original trajectory and the spatial point representation of 37 up-bound vessels based on MDS are shown in Fig. 12. The trajectories are mapping into the points in 2D plane in Fig. 12(b). Then the trajectory clustering problem is transformed into a point clustering problem. The original trajectories consist of 744 coordinate points, and these points belong to the data set  $OP_{37}$ . While the spatial points after mapping only have 37 points, which belong to the data set  $MP_{37}$ . The distance between the points in  $MP_{37}$  is expressed by Euclidean distance, which greatly reduces the time and computation complexity.

The original trajectories and the spatial point representation of 30 down-bound vessels based on MDS are shown in Fig. 13. The trajectories of 30 down-bound vessels is displayed in Fig. 13 (a), which clearly shows the trend of trajectories. The spatial point representation with MDS is shown in Fig. 13 (b), and the relationship between the points can be found easily. From Fig. 13 (b), the trajectories are mapped into the points in 2D plane. The original 30 trajectories consist of 788 coordinate points, while the spatial points after mapping only have 30 points.  $OP_{30}$  represents the original 788 coordinate points, while  $MP_{30}$  represents the 30 points after mapping.

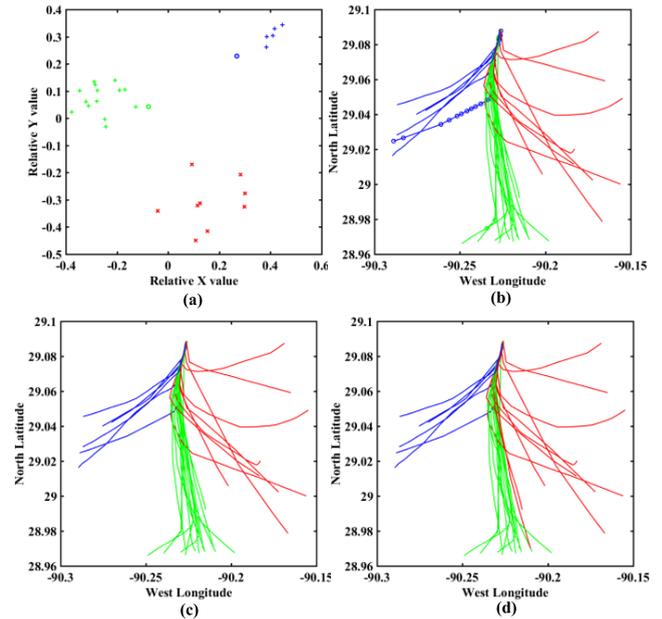
**3) VISUALIZATION OF CLUSTERING RESULTS AND COMPARISON OF DIFFERENT ALGORITHMS**

The parameter  $Eps$  and  $MinPts$  are found by the proposed optimal parameter selection algorithm. The parameters  $Eps = 0.1739$  and  $MinPts = 3$  are the optimal in the trajectory of the 37 up-bound vessels, and the parameters  $Eps = 0.1373$  and  $MinPts = 4$  are the optimal in the trajectories of the 30 down-bound vessels.

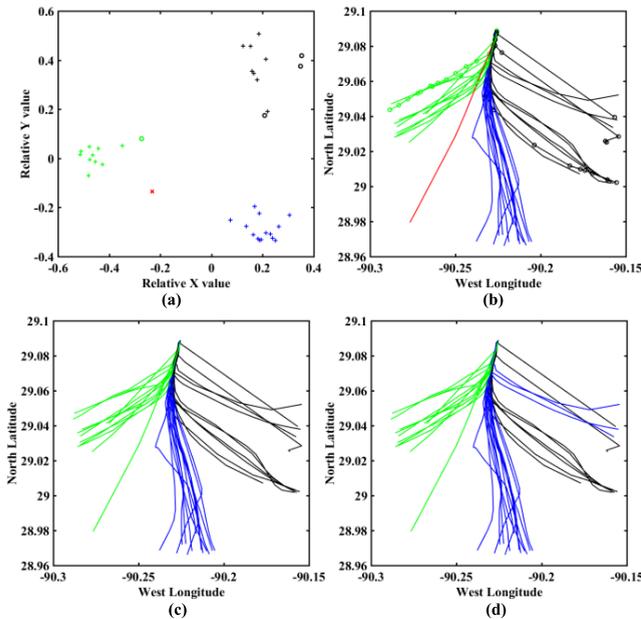
The point clustering result and the corresponding trajectory clustering result of different algorithms about the 37 up-bound vessels are shown in Fig 14. The point clustering results of the improved DBSCAN with  $Eps = 0.1739$  and  $MinPts = 3$  are shown in Fig. 14 (a). As shown in Fig. 14 (a),



**FIGURE 13.** Original trajectory and data mapping results: (a) Original AIS trajectory display of 30 down-bound vessels; and (b) spatial point representation with MDS.



**FIGURE 15.** Clustering results of different algorithms about the 30 down-bound trajectories: (a) spatial point clustering results of the improved DBSCAN when  $Eps = 0.1373$  and  $MinPts = 4$ ; (b) corresponding trajectory clustering results of (a); (c) clustering results of spectral clustering when the cluster center is 3; and (d) The clustering results of the affinity propagation clustering when the cluster center is 3.



**FIGURE 14.** Clustering results of different algorithms about the 37 up-bound trajectories: (a) point clustering results of the improved DBSCAN when  $Eps = 0.1739$  and  $MinPts = 3$ ; (b) corresponding trajectory clustering results of (a), where the red trajectory represents the abnormal trajectory; (c) clustering results of spectral clustering when the cluster center is 3; and (d) The clustering results of the affinity propagation clustering when the cluster center is 3.

the green, blue, and black colors represent different clusters and the red represents the noise points. The green, blue, and black plus signs represent the core points of the different clusters. The black and green circles denote the border points. The corresponding trajectory clustering result is shown in Fig. 14 (b), where the blue, red and green denote different clusters and the red trajectories represent abnormal trajectories. The result of spectral clustering is shown in Fig. 14 (c), and the clustering performance is better than affinity propagation clustering. However, the spectral clustering cannot identify the abnormal trajectories. The clustering result of affinity propagation clustering is shown in Fig. 14 (d), the green, blue, and black also represent different clusters. However, the blue trajectories among the black trajectories are the wrong clustering results. By comparing the Fig. 14 (b), Fig. 14 (c) and Fig. 14 (d), it can be clearly seen that the proposed method

not only recognizes the abnormal trajectories, but also has the better clustering performance.

The point clustering result of the proposed method and the corresponding trajectory clustering results of different algorithms about the 30 down-bound vessels are shown in Fig. 15. The point clustering results of the improved DBSCAN when  $Eps = 0.1373$  and  $MinPts = 4$  are shown in Fig. 15 (a). In Fig. 15 (a), the blue and green represent different clusters and the red expresses noise points. The blue and green plus signs represent the core points. The corresponding trajectory clustering result is shown in Fig. 15 (b), the blue and green trajectories denote different clusters and the red expresses abnormal trajectories. The red trajectories in the Fig. 15 (b) represent the possible abnormal trajectories, and the distances of the red trajectories are relatively larger than those of the green and blue ones. Then, the abnormal trajectories are further analyzed by the COG and SOG. The further verification indicated that these potentially abnormal trajectories are actually at a normal navigational state. These potential anomalies could be the result of a small data sample. The result of spectral clustering is shown in Fig. 15 (c), where the clustering performance is better than that of affinity propagation clustering. Spectral clustering can cluster the trajectories based on the distance matrix, but cannot identify the abnormal trajectories. The result of affinity propagation clustering is shown in Fig. 15 (d), where the blue, red and green trajectories represent different clusters. However, the red trajectories among green trajectories are the wrong clustering results. Spectral clustering and affinity propagation clustering can only cluster the points, and cannot identify other information.

**TABLE 1. Comparison results of different algorithms.**

Symbol	Proposed Algorithm	Spectral Clustering	Affinity Propagation Clustering	Method in [38]
$T_{TC}$	$O(n^2)$	$O(n^2)$	$O(n^2 \cdot \log n)$	$O(n^2)$
$T_{MD-B}$	15.14 min	15.14 min	15.14 min	—
$T_{DTW-B}$	—	—	—	8.25 min
$T_{clus-B}$	2.841 s	2.056 s	3.054 s	1.834 s
$R_{accu-B}$	100%	98.1%	88.8%	100% (*)
$T_{MD-M37}$	4.28s	4.28s	4.28s	—
$T_{DTW-M37}$	—	—	—	5.89 s
$T_{clus-M37}$	0.404s+0.393s =0.796s	1.043s	1.212s	0.836s
$R_{accu-M37}$	100%	97.30%	89.19%	100%
$T_{MD-M30}$	6.29s	6.29s	6.29s	—
$T_{DTW-M30}$	—	—	—	2.075s
$T_{clus-M30}$	0.424s+0.393s =0.817s	1.239s	1.321s	0.737s
$R_{accu-M30}$	100%	100%	93.33%	96.67%
IN	Yes	No	No	No
IC	Yes	No	No	No

$T_{TC}$  represents the time complexity of the involved different algorithms.  $T_{MD-B}$  indicates the running time of MD.  $T_{DTW-B}$  indicates the running time of DTW.  $T_{clus-B}$  represents the clustering running time of the trajectory in the bridge area waterway.  $R_{accu-B}$  represents the clustering accuracy of the trajectory in the bridge area waterway.  $T_{MD-M37}$  indicates the running time of MD about 37 up-bound trajectories in the Mississippi River.  $T_{DTW-M37}$  indicates the running time of DTW about the 37 up-bound trajectories.  $T_{clus-M37}$  represents the clustering running time of the 37 up-bound trajectories in the Mississippi River.  $R_{accu-M37}$  represents the clustering accuracy of the 37 up-bound trajectories.  $T_{MD-M30}$  indicates the running time of MD about the 30 down-bound trajectories in the Mississippi River.  $T_{DTW-M30}$  indicates the running time of DTW about the 30 down-bound trajectories.  $T_{clus-M30}$  represents the clustering running time of the 30 down-bound trajectories.  $R_{accu-M30}$  represents the clustering accuracy of the 30 down-bound trajectories. IN represents whether the method can identify noise. IC indicates whether the method can identify the trajectories with the opposite course.

100% (\*) indicates that the accuracy of clustering is 100% without considering the course. However, four trajectories had an opposite direction relative to the other trajectories.

Our proposed method not only identifies the different clusters, but also detects the anomalous trajectories.

#### F. TIME COMPLEXITY ANALYSIS COMPARISONS WITH OTHER CLUSTERING METHODS

The time complexity of the proposed method involves the calculations by the MD, MDS and the improved DBSCAN algorithm. The time complexity of the methods are as follows: MD is  $O(\prod_{i=1}^n m_i)$ , MDS is  $O(n^2)$ , and the improved DBSCAN algorithm is  $O(n \cdot \log n)$ . The affinity propagation clustering has a complexity of  $O(n^2 \cdot \log n)$ , and spectral clustering is  $O(n^2)$ . In the above time complexity expressions,  $m_i$  expresses the number of the points in the  $i$ -th trajectory, and  $n$  represents the number of AIS trajectories. The time complexity and the running time of the proposed algorithm, spectral clustering and affinity propagation clustering in the bridge area waterway are listed in Table 1.

All the trajectories in the bridge area waterway have 25678 points in  $OP$ . The time complexity of MD is the product of the sum of the points in every trajectory; therefore, the running time of MD is 15.14 min. The time complexity of MD is very large, although the stability of the algorithm is higher. The running time of the data mapping is 1.569 s, and the clustering running time is only 1.272 s after applying MDS. The running time of affinity propagation clustering is 3.054 s, and the spectral clustering is 2.056 s. The clustering accuracy of the proposed method is higher than those of the other two algorithms. Compared with the method in [38], the proposed algorithm in this paper can identify the course and achieve a better clustering result.

The trajectories of 37 up-bound vessels in the Mississippi River data set consist of 744 points, for which the running time of MD is 4.28s, that of data mapping is 0.404s, and that of the clustering is only 0.392s. The total running time is only 0.796s, which is less than the other two algorithms. The running time of affinity propagation clustering is 1.043s, and that of spectral clustering is 1.212s. The clustering accuracy of the proposed method is higher than the other two algorithms. All the trajectories of 30 down-bound vessels have 788 points. The time complexity of MD is the product of the sum of points in every trajectory; thus, the running time is 6.29s. For the proposed method, the data mapping running time is 0.424s, while that of the clustering is 0.393s. The total running time is only 0.817s, which is less than the running time of the other two algorithms. The running time of affinity propagation clustering is 1.239s, and the one of spectral clustering is 1.321s. The clustering accuracy of the proposed method is better than that of the other two algorithms. Compared with the method in [38], the proposed algorithm identifies the potential anomalies and achieves a better clustering result.

As a result, the proposed clustering algorithm has higher clustering accuracy and relatively lower time complexity. The experiments show that the proposed method can not only detect the reverse trajectories but also identify abnormal trajectories automatically.

#### V. DISCUSSION

The experimental results show that the proposed method can find trajectories with an opposite course and discern noisy data. Spectral clustering cannot identify reverse trajectories or abnormal trajectories, and affinity propagation clustering produces incorrect clustering results. The experiments verify that the performance and accuracy of the proposed method are better than the performance and accuracy of spectral clustering and affinity propagation clustering. The data mapping process from trajectories to points can not only greatly reduce the time complexity of clustering but also improve the clustering performance.

The experiments on different data sets confirm the validity and feasibility of the proposed method in this paper. As a trajectory similarity measurement algorithm with strong robustness under subsampling and supersam-

pling, MD can effectively calculate the distance between trajectories, and MDS is then used to transform the distance between trajectories into the distance between points and construct a spatial point representation of the trajectories. These data mapping and dimensional reduction methods greatly reduce the time complexity of trajectory clustering. The experimental results demonstrate the enormous potential of the proposed method for trajectory clustering. Moreover, traffic patterns and customary routes can be found from the clustering results. The fusion among the MD, MDS and improved DBSCAN algorithm represents the innovation of this paper, and it can automatically determine the number of clusters based on the distance between mapping points and find trajectories that present an opposite course.

## VI. CONCLUSIONS AND FUTURE WORK

Trajectory clustering is different from traditional point clustering, and traditional point data cannot be directly clustered via clustering algorithms. Instead, we propose a novel conversion method that can transform the trajectory clustering problem into a point clustering problem. The main contribution of the novel conversion method is that it takes full advantage of the distance information between trajectories and greatly reduces the time complexity while achieving high-quality clustering results. In the experiments, the clustering time does not increase with the amount of data because of the MDS data mapping algorithm. Numerous experiments were carried out on both a bridge area waterway in the Yangtze River and the Mississippi River to verify the effectiveness of the proposed method. The results of comparisons show that the proposed algorithm performs better than those of spectral clustering and affinity propagation clustering. The proposed method can not only detect reverse trajectories but also identify abnormal trajectories automatically.

To generalize the proposed method in future, we need to increase the data set sample to realize the big data analysis. In addition, the parameters in the improved DBSCAN must be determined according to the distances and differences in the distances between points. Thus, future relevant studies should investigate large-scale database research and perform a more detailed study of the automatic method for parameters selection based on navigation directions.

## REFERENCES

- [1] 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, Jan. 2018.
- [2] D. Nikolic, N. Stojkovic, and N. Lekic, "Maritime over the horizon sensor integration: High frequency surface-wave-radar and automatic identification system data integration algorithm," *Sensors*, vol. 18, no. 4, p. 1147, 2018.
- [3] L. Elkins, D. Sellers, and W. R. Monach, "The autonomous maritime navigation (AMN) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles," *J. Field Robot.*, vol. 27, no. 6, pp. 790–818, 2010.
- [4] Y. Yu, Q. Wang, X. Wang, H. Wang, and J. He, "Online clustering for trajectory data stream of moving objects," *Comput. Sci. Inf. Syst.*, vol. 10, no. 3, pp. 1293–1317, 2013.
- [5] L. Sang, A. Wall, Z. Mao, X.-P. Yan, and J. Wang, "A novel method for restoring the trajectory of the inland waterway ship by using AIS data," *Ocean Eng.*, vol. 110, pp. 183–194, Dec. 2015.
- [6] F. Zhu, "Mining ship spatial trajectory patterns from AIS database for maritime surveillance," in *Proc. IEEE Int. Conf. Emerg. Manage. Manage. Sci. (ICEMMS)*, Aug. 2011, pp. 772–775.
- [7] 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.
- [8] B. Ristic, B. F. La Scala, M. R. Morelande, and N. Gordon, "Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction," in *Proc. Int. Conf. Inform. Fusion*, Jun. 2008, pp. 1–7.
- [9] J. Wang, C. Zhu, Y. Zhou, and W. Zhang, "Vessel spatio-temporal knowledge discovery with AIS trajectories using co-clustering," *J. Navigat.*, vol. 70, no. 6, pp. 1383–1400, 2017, doi: [10.1017/S0373463317000406](https://doi.org/10.1017/S0373463317000406).
- [10] A. Harati-Mokhtari, A. Wall, P. Brooks, and J. Wang, "Automatic identification system (AIS): Data reliability and human error implications," *J. Navigat.*, vol. 60, no. 3, pp. 373–389, 2007.
- [11] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Jose, CA, USA, Aug. 2007, pp. 330–339.
- [12] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.
- [13] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 1996, pp. 226–231.
- [14] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Philadelphia, PA, USA, 1999, pp. 49–60.
- [15] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Montreal, QC, Canada, 1996, pp. 103–114.
- [16] Z. Ou and J. Zhu, "AIS database powered by GIS technology for maritime safety and security," *J. Navigat.*, vol. 61, no. 4, pp. 655–665, 2008.
- [17] K. Patroumpas, E. Alevizos, A. Artikis, M. Voudas, N. Pelekis, and Y. Theodoridis, "Online event recognition from moving vessel trajectories," *Geoinformatica*, vol. 21, no. 2, pp. 389–427, 2017.
- [18] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
- [19] A. Ismail and A. Vigneron, "A new trajectory similarity measure for GPS data," in *Proc. 6th ACM SIGSPATIAL Int. Workshop GeoStreaming (IWGS)*, Bellevue, WA, USA, 2015, pp. 19–22.
- [20] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.
- [21] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Beijing, China, 2007, pp. 593–604.
- [22] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 169–194, Jun. 1998.
- [23] W. Wang, J. Yang, and R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *Proc. 23rd Int. Conf. Very Large Data Bases (VLDB)*, 1997, pp. 186–195.
- [24] M. Meilä and D. Heckerman, "An experimental comparison of model-based clustering methods," *Mach. Learn.*, vol. 42, nos. 1–2, pp. 9–29, 2001.
- [25] E. H. Ruspini, "Numerical methods for fuzzy clustering," *Inf. Sci.*, vol. 2, no. 3, pp. 319–350, 1970.
- [26] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A min-max cut algorithm for graph partitioning and data clustering," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2001, pp. 107–114.
- [27] Z. Dehghan and E. G. Mansoori, "A new feature subset selection using bottom-up clustering," *Pattern Anal. Appl.*, vol. 21, no. 1, pp. 57–66, Jun. 2016.
- [28] J. Cong and M. L. Smith, "A parallel bottom-up clustering algorithm with applications to circuit partitioning in VLSI design," in *Proc. 30th Int. Design Automat. Conf. (DAC)*, Dallas, TX, USA, Jul. 1993, pp. 755–760.
- [29] C. Böhringer and T. F. Rutherford, "Integrated assessment of energy policies: Decomposing top-down and bottom-up," *J. Econ. Dyn. Control*, vol. 33, no. 9, pp. 1648–1661, 2009.
- [30] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," *Inf. Syst.*, vol. 26, no. 1, pp. 35–58, 2001.
- [31] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial-temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, 2007.

- [32] Y. He, H. Tan, W. Luo, S. Feng, and J. Fan, "MR-DBSCAN: A scalable MapReduce-based DBSCAN algorithm for heavily skewed data," *Frontiers Comput. Sci.*, vol. 8, no. 1, pp. 83–99, 2014.
- [33] L. Duan, L. Xu, F. Guo, J. Lee, and B. Yan, "A local-density based spatial clustering algorithm with noise," *Inf. Syst.*, vol. 32, no. 7, pp. 978–986, 2007.
- [34] M. Parikh and T. Varma, "Survey on different grid based clustering algorithms," *Int. J. Adv. Res. Comput. Sci. Manage. Stud.*, vol. 2, no. 2, pp. 427–430, 2014.
- [35] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: A multi-resolution clustering approach for very large spatial databases," in *Proc. 24th VLDB Conf.*, New York, NY, USA, 1998, pp. 428–439.
- [36] C. Bouveyron and C. Brunet-Saumard, "Model-based clustering of high-dimensional data: A review," *Comput. Stat. Data Anal.*, vol. 71, pp. 52–78, Mar. 2014.
- [37] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 841–847, Aug. 1991.
- [38] 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, Aug. 2017.
- [39] M. Nanni and D. Pedreschi, "Time-focused clustering of trajectories of moving objects," *J. Intell. Inf. Syst.*, vol. 27, no. 3, pp. 267–289, 2006.
- [40] S. J. Gaffney, W. R. Andrew, P. Smyth, S. J. Padhraic, and M. Ghil, "Probabilistic clustering of extratropical cyclones using regression mixture models," *Climate Dyn.*, vol. 29, no. 4, pp. 423–440, 2007.
- [41] J. Pan, Q. Jiang, and Z. Shao, "Trajectory clustering by sampling and density," *Mar. Technol. Soc. J.*, vol. 48, no. 6, pp. 74–85, 2014.
- [42] Y. Mao, H. Zhong, X. Xiao, and X. Li, "A segment-based trajectory similarity measure in the urban transportation systems," *Sensors*, vol. 17, no. 3, pp. 524–540, 2017.
- [43] E. Cambria, Y. Song, H. Wang, and N. Howard, "Semantic multidimensional scaling for open-domain sentiment analysis," *IEEE Intell. Syst.*, vol. 29, no. 2, pp. 44–51, Mar. 2014.
- [44] Y. Aflalo and R. Kimmel, "Spectral multidimensional scaling," *Proc. Nat. Acad. Sci. USA*, vol. 110, no. 45, pp. 18052–18057, 2013.
- [45] M. C. Hout, M. H. Papesh, and S. D. Goldinger, "Multidimensional scaling," *Wiley Interdiscipl. Rev. Cogn. Sci.*, vol. 4, no. 1, pp. 93–103, 2013.
- [46] K. M. Kumar and A. R. M. Reddy, "A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method," *Pattern Recognit.*, vol. 58, pp. 39–48, Oct. 2016.
- [47] T. N. Tran, K. Drab, and M. Daszykowski, "Revised DBSCAN algorithm to cluster data with dense adjacent clusters," *Chemometrics Intell. Lab. Syst.*, vol. 120, pp. 92–96, Jan. 2013.
- [48] E. Arias-Castro, G. Lerman, and T. Zhang, "Spectral clustering based on local PCA," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 253–309, 2017.
- [49] L. Sun, C. Guo, C. Liu, and H. Xiong, "Fast affinity propagation clustering based on incomplete similarity matrix," *Knowl. Inf. Syst.*, vol. 51, no. 3, pp. 941–963, Jun. 2017.
- [50] P. Li, H. Ji, B. Wang, Z. Huang, and H. Li, "Adjustable preference affinity propagation clustering," *Pattern Recognit. Lett.*, vol. 85, pp. 72–78, Jan. 2017.



**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 research interests include intelligent transportation systems, vessel traffic flow, vessel navigation safety, risk assessment, and intelligent traffic organization.



**KEFENG WU** received the B.Sc. degree from the School of Science, Wuhan University of Technology, China, in 2014, and the M.Sc. degree from the School of Automation, Huazhong University of Science and Technology, Wuhan, China, in 2017. His research direction includes path optimization and machine learning.



**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 are analysis and modeling of safety, resilience, and sustainability of transport networks, particularly

maritime and logistics systems.



from 2018 to 2019. Her research interests include trajectory visualization, trajectory data mining, maritime transport, computational transportation science, and artificial intelligence.

**HUANHUAN LI** received the M.Sc. degree from the School of Science, Wuhan University of Technology, Wuhan, China, in 2015, where she is currently pursuing the Ph.D. degree with the School of Navigation. She was a Visiting Ph.D. Student at the School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, China, from 2016 to 2018. She is currently a Joint Ph.D. Student in maritime and mechanical engineering, Liverpool John Moores University, U.K.,



Session Chair of the IEEE International Conference on Systems, Man, and Cybernetics in 2015.

**RYAN WEN LIU** received the B.Sc. degree from the Department of Mathematics, Wuhan University of Technology, Wuhan, China, in 2009, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2015. He is currently an Associate Professor with the School of Navigation, Wuhan University of Technology. His research interests mainly include computational transportation science, image processing, computer vision, and machine learning. He was the



**NAIXUE XIONG** (M'07–SM'12) received the Ph.D. degree in sensor system engineering from Wuhan University, and the Ph.D. degree in dependable sensor networks from the Japan Advanced Institute of Science and Technology. He was with Georgia State University, Wentworth Technology Institution, and Colorado Technical University for about ten years. He is currently an Associate Professor with the School of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK, USA. He has authored or co-authored over 200 international journal papers and over 100 international conference papers. His research interests include optimization theory, cloud computing, and parallel and distributed computing. Dr. Xiong received the Best Paper Award from the 10th IEEE International Conference on High Performance Computing and Communications and the Best Student Paper Award from the 28th North American Fuzzy Information Processing Society Annual Conference. He has been the general chair, the program chair, the publicity chair, a PC member,

and an OC member of over 100 international conferences. He has been a Reviewer of about 100 international journals, including the IEEE JOURNAL ON SPECIAL AREAS IN COMMUNICATIONS (JSAC), the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS (Parts: A/B/C), the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON MOBILE COMPUTING, and the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. Some of his works were published in the IEEE JSAC, the IEEE or ACM TRANSACTIONS, the ACM Sigcomm Workshop, the IEEE INFOCOM, the ICDCS, and the IPDPS. He is the Chair of the Trusted Cloud Computing Task Force, the IEEE Computational Intelligence Society, and the Industry System Applications Technical Committee. He is the Editor-in-Chief, an Associate Editor, or an Editor Member for over 10 international journals, including an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS: SYSTEMS, an Associate Editor of the *Information Science*, the Editor-in-Chief of the *Journal of Internet Technology*, and the Editor-in-Chief of the *Journal of Parallel and Cloud Computing*, and a Guest Editor of over 10 international journals, including the *Sensor Journal*, *Wireless Networks*, and the *Mobile Networks and Applications*.

• • •