World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:16, No:10, 2022

Spatial Clustering Model of Vessel Trajectory to Extract Sailing Routes Based on AIS Data

Lubna Eljabu, Mohammad Etemad, Stan Matwin

Abstract—The automatic extraction of shipping routes is advantageous for intelligent traffic management systems to identify events and support decision-making in maritime surveillance. At present, there is a high demand for the extraction of maritime traffic networks that resemble the real traffic of vessels accurately, which is valuable for further analytical processing tasks for vessels trajectories (e.g., naval routing and voyage planning, anomaly detection, destination prediction, time of arrival estimation). With the help of big data and processing huge amounts of vessels' trajectory data, it is possible to learn these shipping routes from the navigation history of past behaviour of other, similar ships that were travelling in a given area. In this paper, we propose a spatial clustering model of vessels' trajectories *(SPTCLUST)* to extract spatial representations of sailing routes from historical Automatic Identification System (AIS) data. The whole model consists of three main parts: data preprocessing, path finding, and route extraction, which consists of clustering and representative trajectory extraction. The proposed clustering method provides techniques to overcome the problems of: (i) optimal input parameters selection; (ii) the high complexity of processing a huge volume of multidimensional data; (iii) and the spatial representation of complete representative trajectory detection in the context of trajectory clustering algorithms. The experimental evaluation showed the effectiveness of the proposed model by using a real-world AIS dataset from the Port of Halifax. The results contribute to further understanding of shipping route patterns. This could aid surveillance authorities in stable and sustainable vessel traffic management.

Keywords—Vessel trajectory clustering, trajectory mining, Spatial Clustering, marine intelligent navigation, maritime traffic network extraction, sdailing routes extraction.

I. Introduction

MARITIME transportation is at the heart of global trade and the economy. 90% of world trade is carried out by the shipping industry; this results in the generation and evolution of shipping routes [1]. The decision to choose a route over another route is a strategic judgement for navigation since it will affect fuel consumption, estimate time of arrival, and avoid undesired conditions such as weather or traffic.

Knowledge of shipping routes can be extracted by analysing vessels' trajectories. These trajectories can be obtained from the Automatic Identification System (AIS), which is an automatic tracking system for ships equipped with a transponder that sends information (messages) about the ship's identification at predetermined time intervals, such as the Maritime Mobile Service Identity (MMSI), location, course of the ground (cog), speed over the ground (sog), and so on. Terrestrial stations or satellites then gather AIS messages [2], [3]. A typical trajectory analysis task is to find patterns that have behaved in a similar way. For example, trajectories following similar paths, trajectories moving consistently together or sharing other movement properties. This requires mapping the trajectories into distinct groups based on their distance or similarity. The mapping process is defined as *trajectory clustering*, where trajectories in each group tend to be rather similar and distinct from those in other groups.

Trajectory clustering is ubiquitous and has a critical role in trajectory mining in modern intelligent marine navigation systems for surveillance, anomaly detection, traffic control, etc [4]. In the literature, many proposed studies provide automated solutions for exploring ships' movement patterns and extracting shipping routes by using trajectory clustering frameworks (e.g., density-based clustering frameworks [1], [5], hierarchical clustering method [6]). These approaches cluster vessel trajectories from historical Automatic Identification System (AIS) data. Vessel trajectory clustering is a challenging task because of the uneven spatial distribution of actual AIS data. Unfortunately, clustering methodologies necessitate optimal input parameters selection, in which the choice is highly sensitive to the data. In addition, under real-life conditions, it is difficult to find the optimal parameters when the data and the scale are not well understood. Moreover, a real-world data clustering task may be too complex to be entirely automated for many reasons. First, real-world AIS data may contain a variety of plausible clusters, and purely automatic clustering has no way of establishing a group that suits the operator's objectives as this requires external domain knowledge. Second, the quality of the clustering result is heavily dependent on extracting appropriate features and specifying appropriate similarity measures. Hence, operators are usually motivated and willing to interact with the clustering process as it can often significantly improve clustering quality. Another serious drawback of most trajectory clustering methods is the production of an imaginary representative trajectory (i.e., a short straight-line segment), which is useless for further analytical processing tasks for vessels' trajectories (e.g., anomaly detection, destination prediction, time of arrival estimation, and others). Finally, most of the proposed trajectory clustering algorithms suffer from the high complexity of processing a huge volume of multidimensional data. This work has the objective of presenting a technique that can cluster a massive amount of multidimensional data in a minimal amount of time using the minimal possible information (e.g., spatial information of a vessel).

L. Eljabu is with the Institute for Big Data Analytics, Dalhousie University, Halifax, NS, Canada (e-mail: l.eljabu@gmail.com).

M. Etemad is with the Institute for Big Data Analytics, Dalhousie University, Halifax, NS, Canada.

S. Matwin is with the Institute for Computer Science, Polish Academy of Sciences, Warsaw and Postcode, Poland.

To tackle the above-mentioned problems, we propose a spatial clustering model (SPTCLUST) to extract spatial representations of sailing routes from historical Automatic Identification System (AIS) data of the vessels' trajectories. The vessels' trajectories from AIS data are long and complex, and make the process of extracting knowledge challenging. Clustering the whole path of a trajectory ignores the local characteristics and overlooks the sub-patterns that represent the movement patterns of vessel behavior. Therefore, the first step in our model is segmentation, where a segment can start from one port and stop at the next port. This work focuses on the extraction of traffic patterns more efficiently, using richer information on the same shipping lane between two ports. It aims to extract multiple movement patterns for the same shipping lane, which correspond to a fine-grained clustering of the collected AIS data. Using linear interpolation for adding intermediate points to the vessel trajectory assures position continuity, which improves the clustering algorithm and representative trajectory extraction. The algorithm parameters are determined by using the proposed interactive algorithm to select the optimal input parameters. The generated network can be utilized as the foundation to conduct a more in-depth analysis and knowledge exploration of specific types of vessel activity in a given region. The major contributions of this work are:

- It provides a spatial clustering model to cluster vessel trajectories from AIS data. The entire workflow is computationally efficient and capable of processing massive amounts of raw mobility data in a minimal amount of time.
- It modifies the definition of similarity in clustering techniques, by considering the variation of the spatial features of the trajectory points. The generated clusters represent the various ways of sailing from the origin port to the destination port.
- It modifies the definition of similarity based on motion direction by considering the bearing between segments' endpoints only, which significantly increases the efficiency of the clustering process.
- It provides a solution to optimal input parameters selection, which allows more efficient clustering of the trajectories. The resulting clusters represent distinctive movement patterns.
- It extracts multiple sailing routes for the same shipping lane between two ports from the AIS vessel trajectory data. The extracted sailing routes are complete (port-to-port), real, and smooth, thus useful for further analytical processing tasks.

The rest of the paper is organized as follows: We explain the related work in Section II. Then we define some important definitions related to our work in Section III. After that we propose our methodology with three steps in Section IV. We continue our discussion on experiments in Section V. Then we discuss the pros and cons of our work in the discussion Section VI. Finally, we conclude our paper in Section VII by summarizing our finding and potential future work.

II. Related Work

The most important source of information for vessel navigation is the Automatic Identification System (AIS) which was originally designed by U.S. Coast Guard Navigation Center and is publicly utilized for navigation purposes. We categorize related work into three parts. First, we review related work on trajectory segmentation. After that, we review trajectory clustering studies. Then, we review available studies on reference route construction to extract the representative trajectory of a cluster.

A. Trajectory Segmentation

There are many trajectory segmentation algorithms available for segmenting trajectories in different situations. Li et al. [7] proposed Stay Point Detection (SPD) algorithm that detects stay points by observing a moving object stay within a certain spatial region for a period exceeding a certain threshold [7]. Etemad et al. proposed Sliding Window Segmentation (SWS) which produces segments based on the position of the trajectory points where the moving object changes its behavior [8]. In this work, we use the segmentation method proposed by Eljabu et al. [9], [10]. The vessel's trajectory is partitioned into segments by using a semantic layer of ports. For each generated segment, to ensure it has its actual endpoints, the coordinates of the center of the nearest ports are added as the segments start and end points [9], [10]. This step of adding the nearest ports' center points as endpoints is very important to define the movement of similar segments from the origin to the destination port [9], [10].

B. Trajectory Clustering

Han et al. [1] proposed an enhanced density-based spatial clustering of applications with noise (DBSCAN) method to model vessel behaviors based on trajectory point data. They cluster the AIS data points using a two-layer clustering method. In the first-layer-clustering algorithm, they proposed a parameters auto-selection method to select the two input parameters (MinPts and Eps) of DBSCAN. It finds MinPts by selecting 0.1% of the sample size. The Eps values are calculated by the distribution of the k-nearest neighbor (KNN) distances of each data point. The disadvantage of KNN in large datasets is that the cost of calculating the distance between trajectory points is huge, which degrades the performance of the algorithm. In the second-layer cluster, the MinPts, and Eps are manually adjusted, which is an inadequate solution to the problem of optimal input parameters selection.

Sheng and Yin [5] proposed a density-based trajectory clustering model to extract shipping route knowledge based on Automated Identification System (AIS) data. First, they segmented the trajectory; then, proposed a structure similarity distance measurement which consists of (1) a spatial distance measurement; (2) directional distance; (3) speed distance. Then, a linear transformation is used to normalize spatial, directional, and speed similarity distances, which generate synthetic similarity distances. However, the trajectory segments were transformed into line segments. The

revised DBSCAN clustering method uses synthetic similarity distances to automatically recognize clusters of different vessel densities, which requires no input parameters [5]. After clustering the line segments, they extracted a representative trajectory to describe the overall movement. The representative trajectory is a synthetic line sequence perpendicular to the clustered lines. Providing straight lines of trajectory segments and the representative trajectory does not produce reliable, complete navigable routes from end to end (from origin port to destination port). Also, fully automatic clustering has no way of establishing a group that suits the operator's needs as this requires external domain knowledge. Additionally, the quality of the clustering result is heavily dependent on extracting suitable features and specifying proper similarity measures. Hence, users are usually motivated and willing to interact with the clustering process as it can often significantly help achieve better clustering quality.

Lee et al. proposed a trajectory density-based clustering framework called TRACLUS [11]. It is a partition-and-group framework for discovering common sub-trajectories in the trajectory database. The clustering is done on raw trajectory data composed of points, and the distance measure computations have quadratic complexity. Their solution has at least two threshold parameters that are highly sensitive to the dataset, making it difficult to reproduce or use in practice. It provides an imaginary representative line for each cluster of lines and does not provide an actual complete route from origin to destination.

C. Reference Route Construction

Eljabu et al. proposed a method to construct a reference route for segments, called Reference Route of Trajectory (RROT) [9]. The reference route is a representative trajectory constructed by using the average of multiple trajectory segments following the same direction. The advantage of this method is that the constructed representative trajectory is a complete real segment from port to port.

III. Definitions and Preliminaries

In the first subsection, we present the mathematical notations and definitions related to our method, and then we present the problem statement.

A. Definitions

Definition1. Location (l): is a geolocation of object o at time i, and is defined as, $l_i^o = \langle x_i^o, y_i^o \rangle$, where x_i^o represents the longitude of the location, and y_i^o represents the latitude of the location.

Definition2. Trajectory (τ): is a time-ordered sequence of trajectory points (Locations) of a moving object o, $\tau^o = \langle l_0^o, l_1^o, .., l_n^o \rangle$, n is the total number of trajectory points or *Locations* (l) Each trajectory may carry extra pieces f information which is called segment features.

Definition3. Trajectory Segment s_i: is a set of consecutive trajectory points belonging to a trajectory $\tau^o = \langle l_0^o, l_1^o, .., l_n^o \rangle$, $s^o = \langle l_j^o, \cdots, l_k^o \rangle, \quad j \geq 0, \ k \leq n$ and s^o is a subsequence of τ^o. The process of generating segments from a trajectory is called trajectory segmentation [10].

Definition4. Port P: is a polygon defining a circular area of radius r centered on the geographical coordinates of a sea port [12].

Definition5. Partitioning Position: is the last trajectory point of a trajectory segment where the segment movement behaviour changes.

Definition6. Origin and Destination: is the nearest *Port* to the first trajectory point in a segment is the Origin Port $P_O = l_0$ and the nearest *Port* to the last trajectory point of a segment is the Destination Port $P_D = l_k$ [9].

Definition 7. Reference trajectory: between point A and B is a representative trajectory segment (spatial representation) that shows the average behaviour of all trajectory segments starting from point A and ending at point B [9].

Definition 8. Bearing: is the direction between the meridian and the line connecting two endpoints of two trajectory segments (x_0^1, y_0^1), (x_0^2, y_0^2), as

$$\beta = atan2(X, Y) \tag{1}$$

where X and Y can be calculated as, $X = cos(x_0^2) \times sin\Delta$, $\Delta = y_0^1 - y_0^2$, $Y = cos(x_0^1) \times sin(x_0^2) - sin(x_0^1) \times cos(x_0^2) \times cos\Delta$.

B. Problem Statement

Given a set of vessels' trajectory data $T = \{\tau_1, \tau_2, ..., \tau_n\}$ and a list of ports $P = \{P_1, .., P_v\}$, v is the number of ports, we are interested in extracting a set of possible reference trajectories $R_{i,j} = \{r_1, .., r_m\}$ between P_i and P_j so that each reference trajectory $r_l \in R$ is a sail-able unique route for a vessel, $r_z \neq r_t, 1 \leq t \leq m, 1 \leq t \leq z, t \neq z$.

IV. Methodology

In this section, we propose spatial clustering model *(SPTCLUST)* to extract spatial representations of sailing routes from historical Automatic Identification System (AIS) data, as shown in Fig. 1. The model includes three main steps: (1) Data Preprocessing, (2) Path Finding, and (3) Route Extraction. Each step is explained in detail in the following subsections.

The first step of our framework utilizes a semantic layer of ports' information, P, to capture the related information of the vessel's movement from origin to destination ports. Then, it generates segments from the captured movement information by cross-checking the navigation movement with the semantic layer of ports, P. Therefore, this step receives a set of trajectories, T, and generates a set of segments S with the help of an available list of ports P.

The second step intends to cluster the trajectory segments into groups based on segments directions. Thus, the second step receives the set of segments, S provided by the previous step, and clusters the members of S into $S_i \subseteq S$; so that each segment in S_i is in the same cluster based on their bearing.

The third step clusters the trajectory segments in the clusters from the previous step based on the pooled standard deviation between segments. Then for each resulted cluster applies the RROT (Reference Route of Trajectory) algorithm proposed

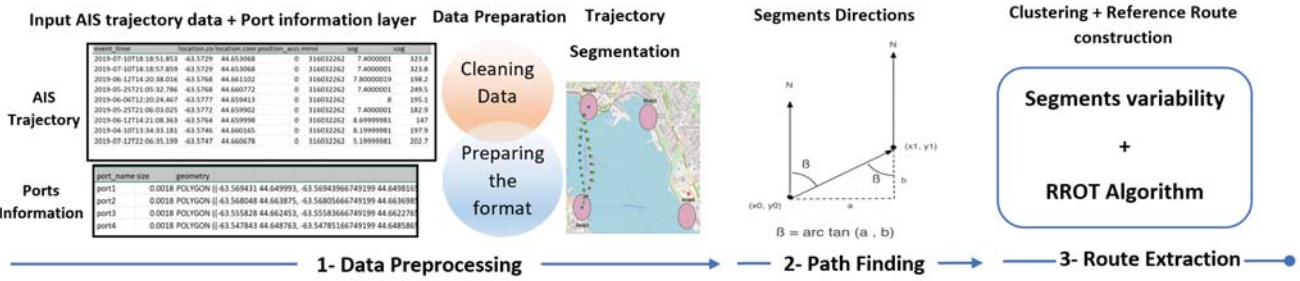


Fig. 1 Framework of Trajectory Clustering and Reference Route construction

in [9] to extract the shipping routes. Hence, the last step returns the $RROT(S_i)$ for all S_i generated in the previous step.

A. Data Preprocessing

The vessels' trajectories of AIS data are long and complex and make the process of extracting knowledge challenging. Clustering the whole path of a trajectory ignores the local characteristics and overlooks the sub-patterns that represent the movement patterns of vessel behavior. Therefore, we use the proposed segmentation method in [9], where the vessel trajectory is partitioned into segments by using a semantic layer of ports. The segmentation process detects the partitioning positions in the trajectory and uses them to divide the trajectory into distinct segments [10].

For each generated segment, to ensure it has its actual endpoints, the coordinates of the center of the nearest port are added as the segment's start and end point [9]. This step of adding the nearest ports' center points as endpoints is very important to define the movement of similar segments from the origin port P_O to the destination port P_D. Therefore, trajectory segments following the same path have matched starting and endpoints.

B. Path Finding

As we have an inflow and outflow of traffic for each port, a preliminary step is to separate these paths. Therefore, to define the sailing routes, our procedure first extracts the directions between trajectory segments. It calculates the *bearing* between the two endpoints of every two trajectory segments. It is like finding the bearing of a line between the two segments' endpoints. Then, we can find this line of direction. The result is the number of paths in different directions between the ports P. According to these paths, the procedure clusters the segments that follow the same direction into one group. The number of the resulted groups equals the number of paths between the related ports of vessel trajectory.

Finally, it visualizes the produced groups of directions to help the operator/domain expert discover interesting patterns and distinguish trivial or wrong patterns from their point of view. Also, this visualization is a technique to discover if a direction group dg_i shows possible multiple routes from port P_O to port P_D.

Let s_i, s_j be two trajectory segments with endpoints (l_i^{start}, l_i^{end}), (l_j^{start}, l_j^{end}), n is the total number of trajectory segments, B is the total number of paths (i.e., directions of segments between ports). DG is the direction groups set; each group contains segments following same direction between two ports (P_O, P_D), $DG = \{dg_1, dg_2, ..., dg_m\}$, where $m = B$. Algorithm 1 presents the grouping algorithm of segments based on the direction:

Algorithm 1 Group Based on Direction Algorithm

```
Input: A set of trajectory segments S = {s_1, s_2, ..., s_n}
Returns: A set of groups DG = {dg_1, dg_2..., dg_m}
DG ← {{}} /*groups of segments*/
B_list ← {} /*list of headings between two segments endpoints*/
for each segment s_i in S do
    if s_1 then /*check if it is the first segment*/
        X = (l_start, l_end)
    else
        Y = (l_start, l_end)
        Bearing = β(X, Y) /*Calculation formula in Equation 1*/
        if Bearing NOT in B_list then
            B_list.add{bearing}
            Pos ← position of bearing from B_list
        else
            Pos ← position of bearing from B_list
        DG(Pos).add{segment}
    return DG
```

Altogether, calculating the *bearing* between the two endpoints of every two trajectory segments reduces the time complexity of clustering segments based on their direction into **constant time O(1)**.

C. Route Extraction

Since trajectory segments are grouped based on direction, the skeleton of the marine route network is identified, but it is hard to observe any discernible trends. It is crucial to discover multiple sailing routes from port A to port B, as it helps the operator/domain expert decide one route over another for navigation according to multiple factors: fuel consumption, avoiding undesired conditions such as weather, traffic, geopolitical tensions, and other external factors. Therefore, the trajectory segments in each group will

be aggregated to define possible routes from one port to another.

1) Identify Threshold: We observed that, for trajectory clustering algorithms, threshold selection is highly sensitive to the data. Threshold selection is the key to effective clustering. To mitigate the manual selection of the threshold, we propose a method that uses pooled standard deviations (SD_{Pooled}) [13], [14] to find the appropriate threshold. It is a method for estimating a single standard deviation to represent two trajectory segments' spatial coordinates in the directed group because they are assumed to come from populations with a common standard deviation. Hence, the pooled standard deviation here can be an indication for finding segments with similar patterns. Then, it produces the histogram of the calculated SD_{Pooled} values to visualize the distributions and how the calculated SD_{Pooled} values cluster. As shown in Fig. 2 the arrow refers to extreme values, while most values cluster on the right of the histogram. If the operator decides that there are two clusters, then we can use a threshold value equal to 54.111.

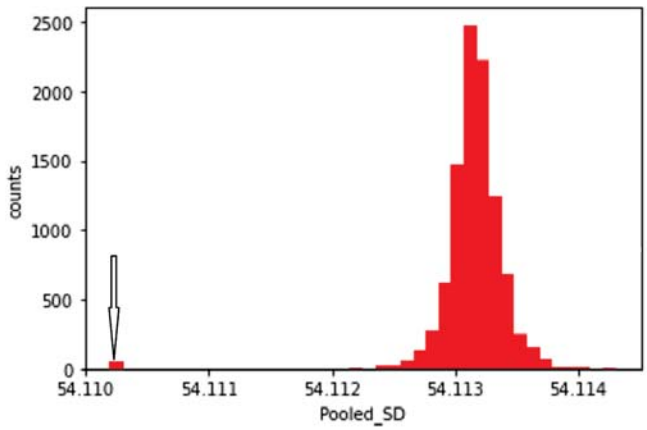


Fig. 2 Histogram of pooled standard deviation values distribution and the arrow points to the outliers

Let dg_i be a group of trajectory segments with same direction; from P_O to P_D. Such that $dg_1 = \{s_1, s_2, .., s_k\}$, where $1 \leq k \leq n$, where n is the total number of trajectory segments. Θ is a set of pooled standard deviations SD_{pooled} between each two trajectory segments spatial coordinates in dg_1, such that: $\Theta = \{SD_{pooled1}, SD_{pooled2}, ..., SD_{pooledk-1}\}$. Since trajectory segments are in different lengths, linear interpolation is used to make all segments have a unified length, Algorithm 2: lines (5,6). The built-in python function is used to interpolate the spatial coordinates of each segment (latitude, longitude). We get the Standard Deviation SD_1 of the first segment's interpolated coordinates, Algorithm 2: lines (8-10). Then, we get the Standard Deviation SD_2 of each other segment's interpolated coordinates. We calculate the pooled standard deviation between the first segment's SD_1 and each other segment's SD_2 in the set dg_1, as:

$$SD_{pooled} = \sqrt{\frac{(n_1 - 1) * SD_1^2 + (n_2 - 1) * SD_2^2}{n_1 + n_2 - 2}} \quad (2)$$

where n_1 is the length of interpolated coordinates of s_1, n_2 is the length of interpolated coordinates of the compared segment s_i. Algorithm 2 presents the identify threshold algorithm. It is used to identify threshold/s of each group of direction:

Algorithm 2 Identify Threshold Algorithm

1: $Input$: (1) A set of trajectory segments following same direction $dg_1 = \{s_1, s_2, ..., s_k\}$.
2: $Returns$: A list of threshold values between the segments $\Theta = \{SD_{pooled1}, SD_{pooled2}, ..., SD_{pooledk-1}\}\}$.
3: $\Theta \leftarrow \{\{\}\}$ /*list of variability values, possible threshold/s*/
4: **for** each segment s_i in dg **do**
5: $interp\_lat = np.interp(s_i.latitude)$
6: $interp\_lon = np.interp(s_i.longitude)$
7: $coord = (interp\_lat, interp\_lon)$
8: **if** s_1 **then** /*check if it is the first segment*/
9: $SD_1 = np.std(coord)$
10: $n_1 = len(coord)$
11: **else**
12: $SD_2 = np.std(coord)$
13: $n_2 = len(coord)$
14: $SD_{pooled}(SD_1, SD_2, n_1, n_2)$ /*The calculation is detailed in Equation 2*/
15: Θ.add(SD_{pooled})
16: **return** Θ

The visualization from the groups of the directions (Section IV-B) is a technique to discover if a group dg_i shows possible multiple routes from port P_O to port P_D. Then, the resulted Θ set's maximum and minimum values, in addition to the histogram of Θ set values will be used to determine the threshold/s to cluster the segments in the group of direction dg_i. Otherwise, segments in a group of directions will be aggregated to create one reference route from port P_O to port P_D. In conclusion, clustering segments based on their spatial information variability by measuring the pooled standard deviation between segments reduces the time complexity of segment clustering to **linear time O(n)**.

2) Cluster segments: At this stage, the clustering algorithm uses the determined threshold by the operator to get the possible routes of segments following the same direction. The clustering algorithm has the same steps of identifying threshold algorithm (Algorithm 2). The only difference here is that the clustering algorithm uses the pooled standard deviation to match the determined threshold and create the clusters.

Let dg_i be a group of trajectory segments with the same direction; from P_O to P_D. Such that $dg_1 = \{s_1, s_2, .., s_k\}$, where $1 \leq k \leq n$, n is the total number of trajectory segments. C is a set of clusters; each cluster contains segments following same route between two ports, $C = \{c_1, c_2, ..., c_u\}$, where $1 \leq u \leq k$. Algorithm 3 shows the clustering algorithm of segments based on the directions:

Algorithm 3 Cluster trajectory segments of same Direction Algorithm

$Input$: (1) A set of trajectory segments following same direction $dg_1 = \{s_1, s_2, ..., s_k\}$.
$Input$: (2) Threshold/s.
$Returns$: A set of clusters $C = \{c_1, ..., c_u\}$.
$C \leftarrow \{\{\}\}\times$ (Number of Threshold/s) /*groups of clusters*/
for each segment s_i in dg **do**
    $interp\_lat = np.interp(s_i.latitude)$
    $interp\_lon = np.interp(s_i.longitude)$
    $coord = (interp\_lat, interp\_lon)$
    if s_1 **then** /*check if it is the first segment*/
        $SD_1 = np.std(coord)$
        $n_1 = len(coord)$
    else
        $SD_2 = np.std(coord)$
        $n_2 = len(coord)$
        $SD_{pooled}(SD_1, SD_2, n_1, n_2)$ /*The calculation is detailed in Equation 2*/
        if $SD_{pooled} \leq Threshold_1$ **then**
            C(0).add$\{s_i\}$
        else if $SD_{pooled} \leq Threshold_2$ **then**
            C(1).add$\{s_i\}$
            :
        else
            C(u).add$\{s_i\}$
        return C

3) Reference Route Construction:: For each resulting cluster, a reference route (i.e., a representative trajectory) is constructed by using the Reference Route of Trajectory *(RRoT)* algorithm proposed in [9]. The reference route is a mean segment that represents the trajectory segments of the clusters between the two ports P_O, P_D. The RROT (C) returns the possible reference routes set $R_{i,j} = r_1, r_2, ..., r_m$. The extracted routes are spatial representations of the representative trajectories for each cluster.

V. Experiments

This section evaluates the performance of our spatial clustering model of vessel trajectory from historical AIS data *(SPTCLUST)* to extract possible multiple sailing routes between each two ports. We tested it with two real vessel trajectories from Halifax Harbor, from March to July 2019. One transit ferry and one cargo vessel AIS data, as shown in Fig. 3. The source data were accessed from [15]. As mentioned in the introduction, AIS transmits ship-related data such as MMSI (Maritime Mobile Service Identity). Therefore, MMSI, the unique 9-digit identification number, is used to represent the ship; an index of each ship. Table I illustrates the statistics of each vessel trajectory data: the length of vessel trajectory and the number of the generated segments.

TABLE I
The Statistics of Vessels Trajectory Data

Transit Ferry	# Points	# Segments	# Paths
Ferry	103162	4263	6
Cargo	38853	24	7

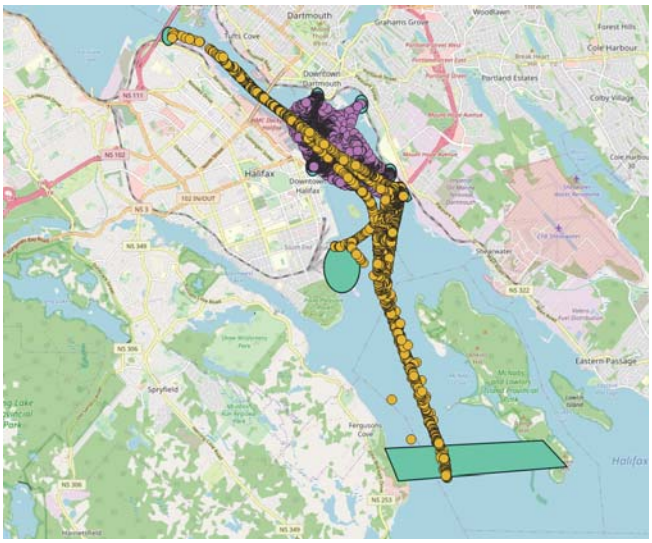


Fig. 3 An overview of two vessels trajectories: The trajectory points of the transit ferry are in purple and the trajectory points of cargo are in yellow with a semantic layer representing the ferry terminals and piers that the cargo vessels are moored to load/unload their goods (the green zones)

First of all, we preprocessed the AIS dataset of ferry and cargo trajectories by cleaning the data and preparing its format. Then, a semantic layer of port information (4 ferry terminals and 4 cargo piers) is used to capture the related information of the vessel's movement from origin to destination ports. Then, segments are generated from the captured movement information by cross-checking the navigation movement with the semantic layer of ports. Then, the center points of the nearest port are added as endpoints to each trajectory segment, which is important to ensure that segments have matched endpoints. As a result, trajectory segments are generated. We evaluated our model accuracy using available ground truth and visualization.

A. Parameters Setting

The *SPTCLUST* takes the trajectory segments of each vessel as an input. Then, the algorithm 1 clusters the segments based on their direction, to distinguish the inflow and outflow of the segments between the set of ports. As we have the ground truth data of segments directions. The accuracy of this step is reported in Table II.

TABLE II
Accuracy and f1 Measure of Clustering the Segments Based on the Direction (Algorithm 1)

Ferry		Cargo	
Acc.	f1	Acc.	f1
100%	100%	100%	100%

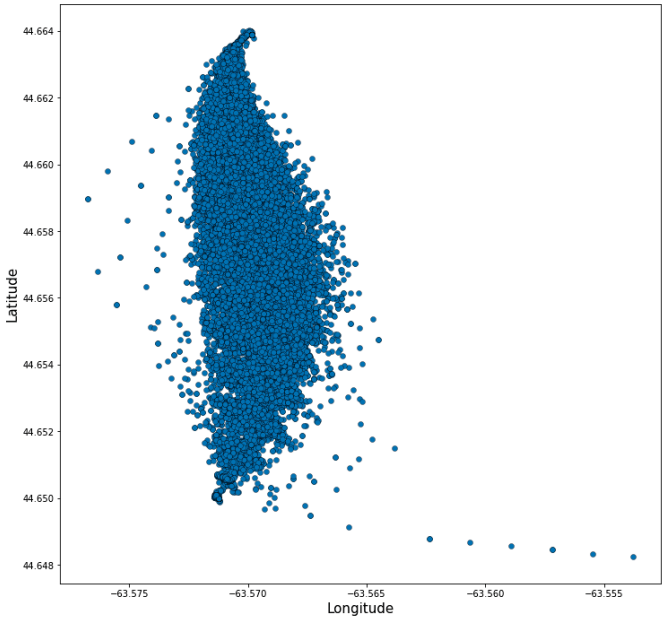
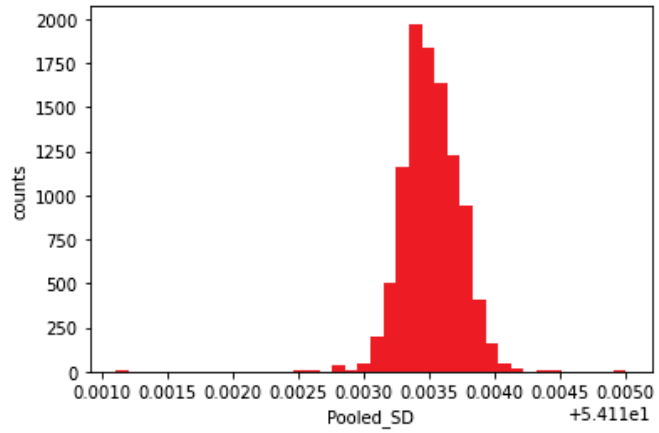


Fig. 4 First direction group from transit ferry trajectory; this group represents path1

As the direction groups are identified, let us take the first direction group (i.e., Path_1) of ferry trajectory, as shown in Fig. 4. Then, we perform the route extraction procedure, with the first step of this procedure being the identification of the threshold/s (Algorithm 2). We identify the threshold/s by using linear interpolation to unify the length of the segments and pooled standard deviations to measure the variability of the spatial information between segments. The results of this algorithm are the variability values and their histogram, from which the operator can determine the threshold/s, as shown in Fig. 5.



(a) Histogram of pooled standard deviation values distribution

```
MIN_Pooled_SD::== 54.1111
MAX_Pooled_SD::== 54.115
The Max  10  Pooled_SD values [54.115, 54.115, 54.1145, 54.1144, 54.1142, 54.1142, 54.1142, 54.1141, 54.1141, 54.1141]
The Min  10  Pooled_SD values [54.1111, 54.1125, 54.1126, 54.1128, 54.1128, 54.1128, 54.1128, 54.1128, 54.1128, 54.1129]
```

(b) Pooled standard deviation values resulted from Algorithm 2

Fig. 5 A visualization of the histogram results from identifying threshold/s algorithm 2 for clustering step

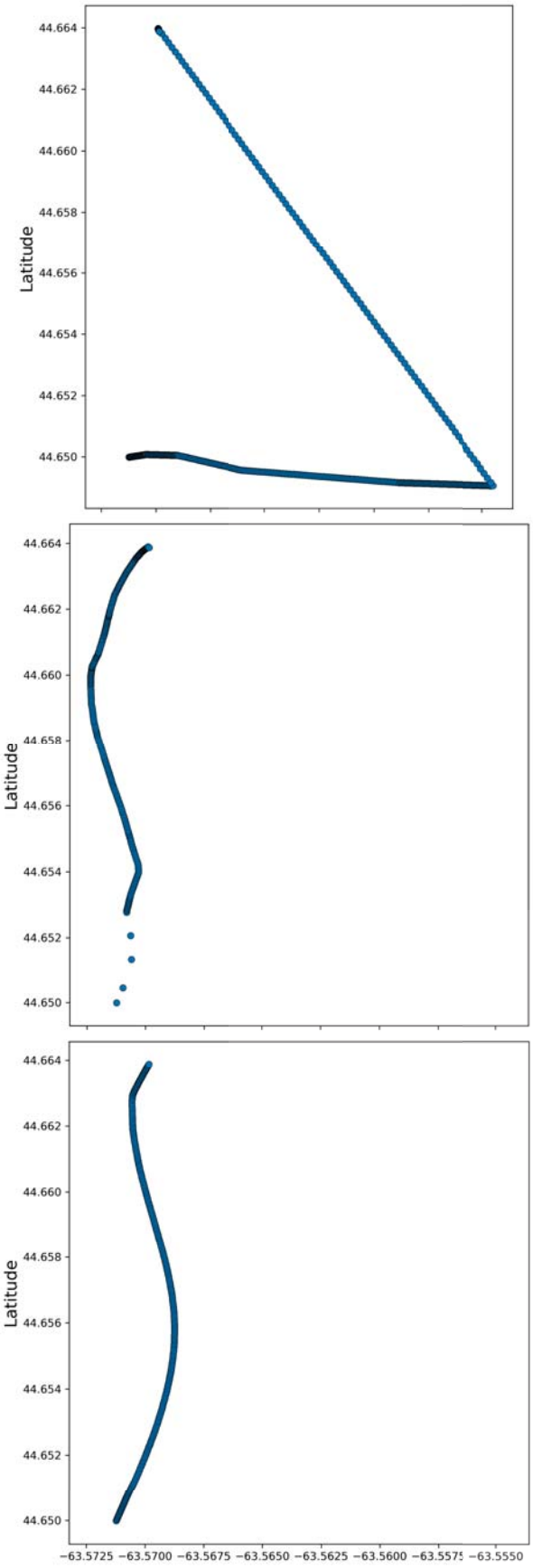


Fig. 6 Clusters of trajectory segments in the first direction group based on spatial information variation

The results in Fig. 5 show the histogram with one peak, which represents the unimodal distribution of the variability values of the spatial information between segments with two outliers. This means that most of the segments in this direction group follow the same movement pattern, but there are some segments that deviate from this distribution; the outliers.

The second step is to cluster segments in the direction group based on the selected threshold by using the Algorithm 3. The segments are clustered based on the variability in their spatial information (the pooled standard deviation measure). If the pooled standard deviation between segments is less than or equal to 54.1111, create cluster1. If the pooled standard deviation between segments is greater than or equal to 54.1142 create cluster2, else create cluster3. Fig. 6 depicts the clustering results of the segments based on variation in their spatial features. Finally, for each resulting cluster in Fig. 6, create a reference route (i.e., a representative trajectory). The reference route describes the overall movement of the trajectory segments that belong to the cluster. For this step, the *RROT* algorthim [9] is used to create the reference route for each cluster. Fig. 7 depicts the results; thus, we have three sailing patterns representing transit ferry movement for the same maritime path (path_1).

The same procedure is followed for each direction group for the ferry trajectory, where each direction group represents a maritime path between two ports or a shipping lane. Path_2 had a threshold of 45.1131, which means two clusters. Path_3 and Path_4 the histograms show there are no outliers, so there is no clustering and a reference route is directly constructed. Path_5 has only one segment, so there is no clustering and a reference route is directly constructed. Path_6 has only two segments with different patterns; there is no clustering and a directly constructed reference route for each segment.

The cargo trajectory data reveal that segments in each group have similar patterns. Hence, the reference route will be constructed from here; there is no need for further clustering. Unlike ferry trajectory direction groups that reveal discernible paths, cargo trajectories do not. This may support the assumption that larger vessels (e.g., cargo vessels) travel between the same ports, following the same route with similar speed and direction.

The constructed reference routes are spatial representations that are displayed on a map using the QGIS application to visualize the final results of the *SPTCLUST* model, which clusters vessel trajectories to extract multiple movement patterns for the same shipping lane. The next subsection provides a visualization of the clustering results of the proposed model for ferry and cargo vessel trajectory. Visualization is key to providing adequate support for human decision-makers, allowing them to picture and explore the movement patterns of vessels' trajectories and to allow clear visual identification of distinct movement patterns. Direction groups are represented with distinctive colors, and then, the constructed reference routes have the colours of their direction groups. As previously stated, this model can be used to conduct a more in-depth analysis and exploration of the maritime traffic of specific types of vessels in a given region.

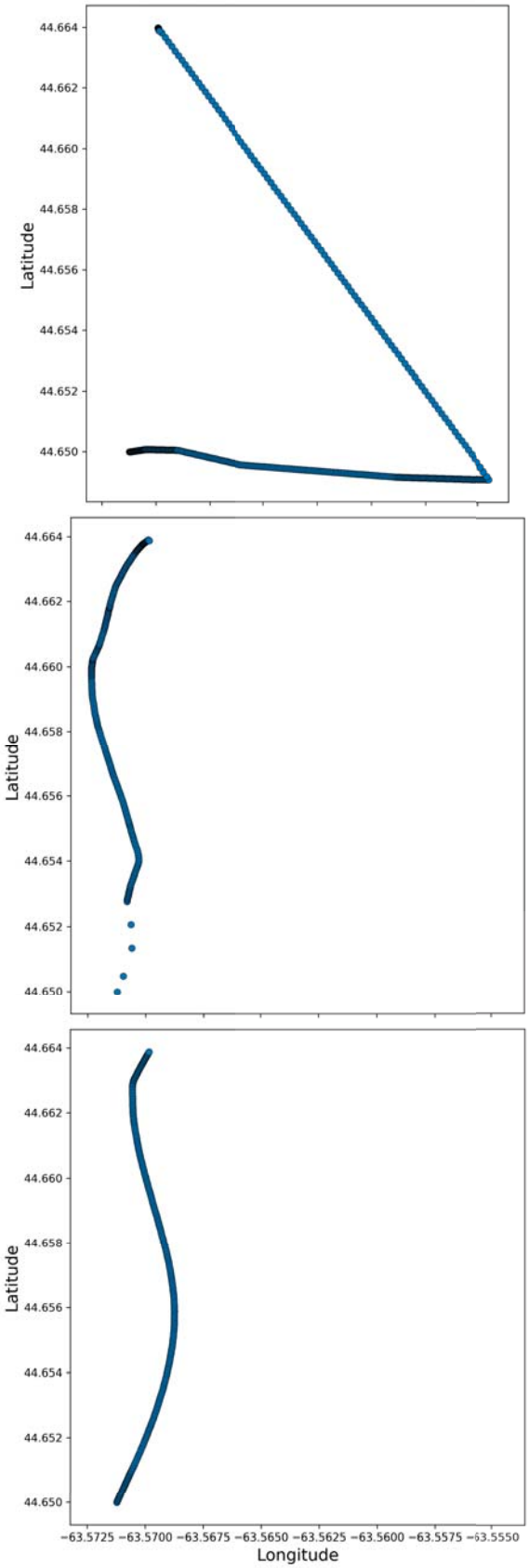
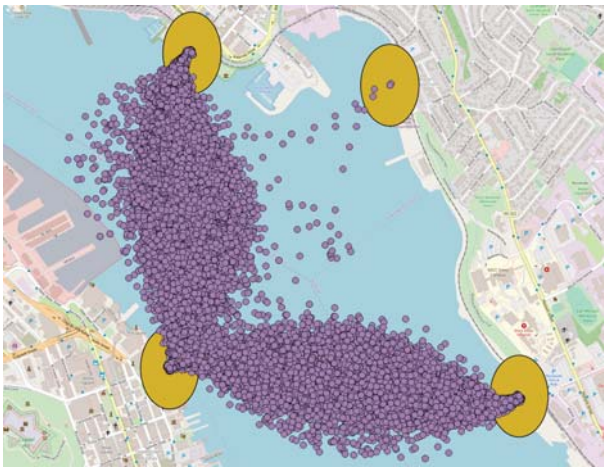


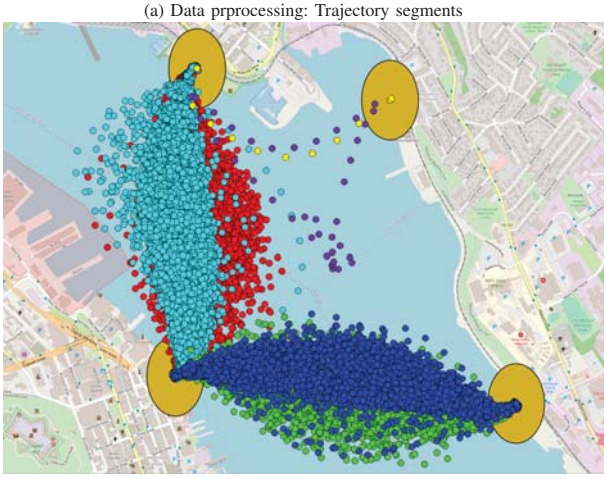
Fig. 7 Reference routes (i.e., representative trajectories) represent the trajectory segments in each cluster

B. Clustering Result

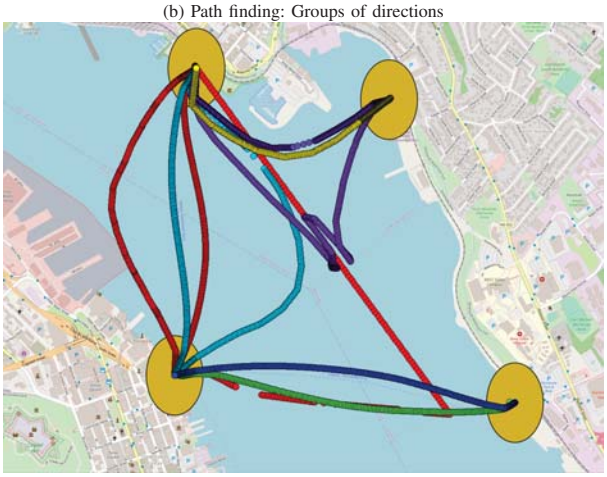
According to our previous parameters setting, we visualize the clustering results with different colours. The output of the *SPTCLUST* model's three main parts is visualised in order to cluster ferry trajectory and extract its traffic network representation.



(a) Data prprocessing: Trajectory segments



(b) Path finding: Groups of directions

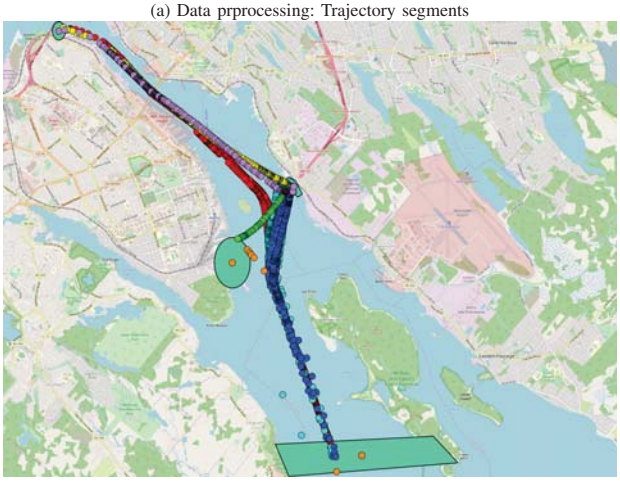


(c) Route extraction: Reference routes

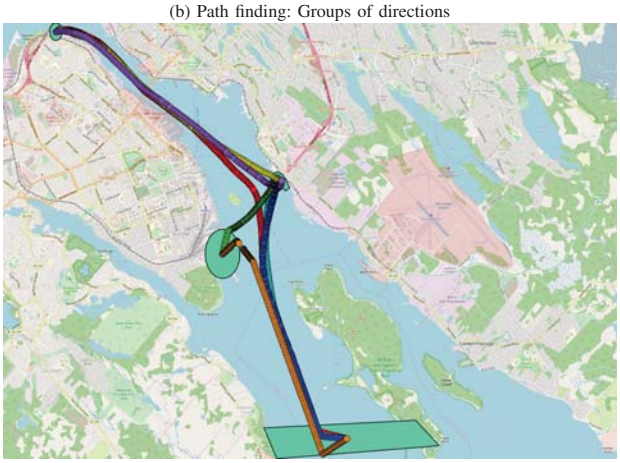
Fig. 8 A depiction of the SPTCLUST framework results of AIS transit ferry vessel trajectory data clustering and the extraction of its navigable routes

The output of the *SPTCLUST* model's three main parts is visualised in order to cluster cargo vessel trajectory and extract its traffic network representation. First, it shows the segmented trajectory of the cargo vessel. Then, it shows the direction groups of different colors. Each colour represents segments following the same direction. Finally, it shows the representative trajectories of each group of directions.

(a) Data prprocessing: Trajectory segments



(b) Path finding: Groups of directions



(c) Route extraction: Reference routes

Fig. 9 A depiction of the SPTCLUST framework results of AIS cargo vessel trajectory data clustering and the extraction of its navigable routes

VI. Discussion

In this paper, a spatial clustering model of AIS vessel trajectory data (SPTCLUST) is proposed to extract the possible movement patterns for the same shipping lane. The movement patterns are extracted as spatial representations of routes between any two ports, which correspond to a fine-grained clustering of the collected AIS data. The SPTCLUST model provides the foundations for modern intelligent marine surveillance and marine traffic control. The proposed SPTCLUST model has the following characteristics:

The SPTCLUST is computationally efficient and capable of clustering multidimensional data in a minimal amount of time. Our model produces a real, complete, smooth representative trajectory from the origin port to the destination port. As a result, the extracted maritime traffic network gives more detailed information on the vessel's sailing behavior, which can be utilized to undertake a more in-depth analysis and comprehension of a specific vessel's maritime traffic in a given area.

An interactive approach includes visualization to select the clustering threshold, making it comparable to other trajectory clustering models to extract shipping route knowledge. In contrast to Han et al. [1], our proposed parameter selection method has a linear $O(n)$ time complexity, whereas their proposed parameter autoselection method has an $O(nlog(n))$ time complexity with the use of indexing to accelerate the computation. In the next step, they went back to manually adjusting parameters. Our approach helps the operator/domain expert to control the clustering process by setting the optimal input parameters, interpreting the results, and directing the algorithm towards the solution that better describes the underlying phenomena.

Our model requires two clustering stages to extract shipping routes from vessel trajectory, making it comparable to other trajectory clustering models to extract shipping route knowledge. Compared to Sheng and Yin [5], our methodology provides a more efficient solution to cluster the segments of vessel trajectory from AIS data. Our model first clusters the trajectory segments based on their direction using the bearing between the endpoints of the segments, which has a constant $O(1)$ time complexity, whereas Sheng and Yin [5] proposed method has a quadratic $O(n^2)$ time complexity. In the second step, our method clusters the segments in the direction groups and extracts spatial representations of the representative trajectories, which are real, smooth, complete segments from port to port, using a linear $O(n)$ time complexity. On the other hand, Sheng and Yin [5] use linear transformation to convert trajectory segments to short straight lines and extract hypothetical representative trajectories with huge time complexity. The extracted short straight-line segments are useless for further analytical processing tasks for vessel trajectories (e.g., anomaly detection, destination prediction, time of arrival estimation, and others). However, the time complexity of the enhanced DBSCAN clustering algorithm is $O(nlogn)$ to the number of the processed points. So, the processing time in clustering would be huge to find the next trajectory point. Also, because of the density of AIS messages, DBSCAN will have a hard time separating the clusters, and usually all the data points are close to each other. Overall, our model can handle the non-linearity of the trajectory segments and improve the computational efficiency of vessels' trajectory clustering, which enhances the performance of the trajectory clustering problem in the maritime domain. Also, our approach outputs not only the curve representation of trajectory segments between ports (that in our case represent distinct mobility patterns) but also a meaningful clustering of the input dataset, which makes our approach suitable for visual data analysis.

Although the proposed method makes it very convenient to obtain multiple sailing routes for marine traffic networks, there are some weaknesses that need to be improved. The algorithm relies on bearings at the endpoints of the trajectory segments. If we have two routes with the same start point and same endpoint but these routes are separated in the middle by an island, it is highly likely our model fails to separate them because of the similarity between routes. This is a weakness that can be resolved in future work by using intermediary points between two ports so that the difference between routes is accentuated. Also, our algorithm does not account for temporal information during clustering. This extension will improve the usability of our algorithm in anomaly detection and traffic management.

VII. Conclusion

In this paper, we propose a model for spatial clustering of vessel trajectories called SPTCLUST, which is a new paradigm in trajectory clustering. To automatically extract navigable routes, we segment vessel trajectory data using a semantic layer of port information and propose clustering the segments based on their directions, which helps to distinguish the directions of the extracted routes. Based on direction clustering, an effective method to choose threshold/s is proposed, which is the key to cluster the segments in each direction based on the segments' spatial information variability. Finally, each cluster's segments are aggregated to create the sailing routes in the marine traffic network. The experimental results show that the proposed method can automatically extract multiple sailing routes between a set of ports, which can be used for modern intelligent marine surveillance and traffic control systems. Also, our proposed method can be used as a visual exploration tool in applications such as marine traffic monitoring, planning, and route selection.

Discovering these routes is essential to many other problems, such as the estimate time of arrival, trip planning, and fuel consumption optimization. In the future, we plan to improve our algorithm so that it can be more vigilant about mistakes such as passing over land. This can be accomplished in a variety of ways. The first thing we would try is to extract critical points in the route by doing trajectory segmentation using the bearing feature. This will give us turning points. This emphasises the difference between segments and aids the algorithm in distinguishing between legitimate routes. Another approach that we would experiment with is to use the length

and time duration of each trajectory as the source for our clustering.

REFERENCES

[1] X. Han, C. Armenakis, and M. Jadidi, "Modeling vessel behaviours by clustering ais data using optimized dbscan," *Sustainability*, vol. 13, p. 8162, 07 2021.

[2] J.-S. Lee and I.-S. Cho, "Extracting the maritime traffic route in korea based on probabilistic approach using automatic identification system big data," *Applied Sciences*, vol. 12, no. 2, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/2/635

[3] W.-J. Son, J.-S. Lee, H.-T. Lee, and I.-S. Cho, "An investigation of the ship safety distance for bridges across waterways based on traffic distribution," *Journal of Marine Science and Engineering*, vol. 8, no. 5, 2020. [Online]. Available: https://www.mdpi.com/2077-1312/8/5/331

[4] J. Bian, D. Tian, Y. Tang, and D. Tao, "A survey on trajectory clustering analysis," *ArXiv*, vol. abs/1802.06971, 2018.

[5] P. Sheng and J. Yin, "Extracting shipping route patterns by trajectory clustering model based on automatic identification system data," *Sustainability*, vol. 10, p. 2327, 07 2018.

[6] S. Wang, S. Gao, and W. Yang, "Ship route extraction and clustering analysis based on automatic identification system data," in *2017 Eighth International Conference on Intelligent Control and Information Processing (ICICIP)*, 2017, pp. 33–38.

[7] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location history," in *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 10. [Online]. Available: https://doi.org/10.1145/1463434.1463477

[8] M. Etemad, Z. Etemad, A. Soares, V. Bogorny, S. Matwin, and L. Torgo, "Wise sliding window segmentation: A classification-aided approach for trajectory segmentation," 2020.

[9] L. Eljabu, M. Etemad, and S. Matwin, "Destination port detection for vessels: An analytic tool for optimizing port authorities resources," *International Journal of Civil and Architectural Engineering*, vol. 15, no. 8, pp. 398 – 406, 2021. [Online]. Available: https://publications.waset.org/vol/176

[10] L. Eljabu, M. Etemad, and S. Matwin, "Anomaly detection in maritime domain based on spatio-temporal analysis of ais data using graph neural networks," in *2021 5th International Conference on Vision, Image and Signal Processing (ICVISP)*, 2021, pp. 142–147.

[11] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," ser. SIGMOD '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 593–604. [Online]. Available: https://doi.org/10.1145/1247480.1247546

[12] E. Carlini, V. Monteiro, A. Soares, M. Etemad, B. Machado, and S. Matwin, "Uncovering vessel movement patterns from ais data with graph evolution analysis," in *EDBT/ICDT Workshops*, 01 2020.

[13] J. Pum, *A practical guide to validation and verification of analytical methods in the clinical laboratory*, 01 2019.

[14] A. Kallner and E. Theodorsson, "Repeatability imprecision from analysis of duplicates of patient samples and control materials," *Scandinavian Journal of Clinical and Laboratory Investigation*, vol. 80, no. 3, pp. 210–214, 2020, pMID: 31899972. [Online]. Available: https://doi.org/10.1080/00365513.2019.1710243

[15] M. Etemad, A. Soares, J. Rose, A. Hoseyni, and S. Matwin, "A trajectory segmentation algorithm based on interpolation-based change detection strategies," 02 2019.