Detecting representative trajectories from global AIS datasets

Nikolas Zygouras[1], Giannis Spiliopoulos[1] and Dimitris Zissis[1,2]

***Abstract*— With real time vessel surveillance data now becoming available at an increasing rate, there is a growing interest in applications that can forecast future vessel positions and routes, especially in congested and busy areas. Since vessels move in "free space", a prerequisite to effectively forecasting vessels' future locations is accurately discovering representative tracks (common paths followed by several vessels). Towards this direction, this work introduces a novel data driven framework that is capable of detecting spatial representations of complete trajectories (from port to port) from massive Automatic Identification System (AIS) datasets. Along these lines, we present a novel approach for forecasting representative tracks from noisy and non-uniform datasets (number of points, sampling rates, coverage gaps etc.) at a global scale. Our technique models the entire space where the vessels traveled in the past, detecting the set of frequently followed locations. This gives our proposed method the ability to forecast the most likely movement from a given query location towards a destination port. Finally, we present extensive experiments with real-world data, so as to demonstrate the effectiveness of our proposed method.**

I. Introduction

Shipping patterns and the related shipping routes are often falsely considered static over time. In reality they are highly dynamic, affected by changes in supply and demand, economic growth, port throughput and specialization, technical advancements, geopolitical tensions and other external factors. The paths connecting these ports are often highly affected on a spatial level (e.g. boundaries, length) or completely disappear. For example, after the blockage of the Suez Canal in March 2021, several ships were rerouted, diverting around the Cape Horn (the southern tip of Africa) adding extra 3,800 miles to their journey and up to 12 days extra sailing time. The majority of route forecasting and time of arrival forecasting algorithms failed to adapt their forecasts, as they relied on traditional and now outdated routing algorithms and cartography. Figure 1 illustrates Conti Cortesia's route deviation from the representative trajectory that connects Mundra and Suez. Essential for effective anomaly detection is building an accurate model of normalcy and updating traditional cartographic maps depicting representative routes between ports with novel data driven commonly travelled routes. The understanding of the complex maritime environment and a vessel behaviour though, cannot be limited to simply connecting vessel positions as they travel across the seas.

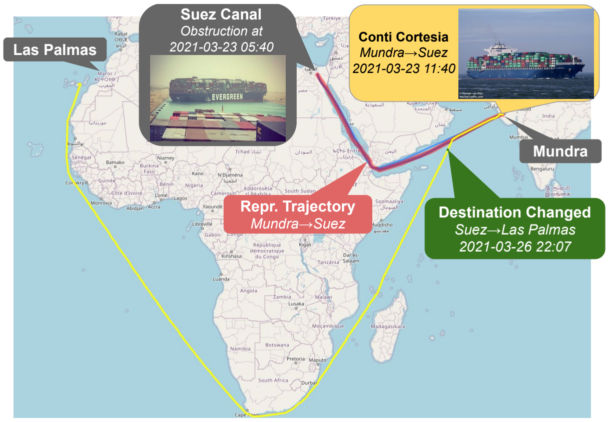


Fig. 1: Illustration of Suez Canal and the Ever Given accident, that caused the Suez Canal obstruction on 2021-03-23 and its impact in the route deviation of Conti Cortesia that operated the route Mundra → Suez.

In 2002 the International Maritime Organisation SOLAS Agreement made it compulsory for vessels over 299 Gross Tonnage(GT) to be fitted with an Automatic Identification System (AIS) transceiver, while in 2006 simpler transceivers made it possible for even smaller ships to join the AIS. The shipborne AIS allows for the efficient exchange of navigational data between ships and between ships and shore stations, with the aim of improving safety of navigation. With the AIS, ships voluntarily broadcast their position and velocity, along with other identification and voyage-related information. This opened up a range of opportunities beyond the original scope of AIS. Nowadays a number of publicly available websites (such as marinetraffic.com) provide an accurate up to date depiction of vessel traffic flows across the globe, reporting the positions of more than 200,000 vessels in real time.

We are now witnessing a growing demand for applications which can make use of the information hidden in huge mobility data repositories (such as AIS), ranging from travel time estimation, to predicting future traffic flow and anomaly detection across the globe. A typical prerequisite data analysis task is that of finding objects that have moved in a similar way. This requires mapping the underlying mobility data or trajectories into descriptive groups which reveal common patterns in the data. The challenge is not novel in the "trajectory data mining" research community and can be defined as that of clustering trajectories. In this context, common paths followed by several moving objects are defined as representative trajectories. For this, throughout

[1] The authors are with MarineTraffic, Athens, 11525, Greece {nikolas.zygouras,giannis.spiliopoulos, dimitris.zissis}@marinetraffic.com

[2] Dimitris Zissis is also with the Department of Product and Systems Design Engineering, University of the Aegean, Ermoupoli, Syros 84100, Greece dzissis@aegean.gr

the related literature numerous clustering approaches have been presented (*i.e.* OPTICS[1], DBSCAN [2], BIRCH [3], TRACLUS[4]), which as we shall see in the following section can be categorised based on their capacity to work on whole trajectories or only portions of these, and their ability to forecast representative trajectories in "free space" or over contained networks (such as roads).

However designing a complete framework for forecasting representative trajectories at a global scale is not a trivial task and can be rather challenging. A serious drawback of the majority of these approaches is their capacity to scale to large datasets and coverage areas. Unfortunately, the majority of these works rely on input parameters which need to be defined by domain experts or selected in a visual way. Since these approaches are highly sensitive to the selected input parameters, the usefulness and practicality of the result is undermined. Also, simple density based or hierarchical clustering techniques that are commonly applied in point clouds are inadequate to model complex objects movements that usually follow multiple paths towards a destination port. Digital maps (*i.e.* Open Street Map[1]) that contain paths connecting different ports may face outdated issues. At the same time digital maps usually provide a single path connecting an origin and a destination port, while in practice it is likely that vessels follow multiple paths which deviate significantly from each other.

Thus the focus of our work is on defining a real world solution with the desired properties of i) practicality ii) accuracy and iii) execution efficiency. In this paper, we introduce a novel algorithmic approach for the purpose of defining smooth representative trajectories in free-space at a global scale. Our focus is on forecasting complete representative trajectories in the maritime domain from a given query location to a destination port. We demonstrate our approaches effectiveness and accuracy on a large highly skewed and non uniform dataset. Along these lines, we present a computationally efficient approach for forecasting representative tracks from noisy and non-uniform datasets (number of points, sampling rates, coverage gaps etc.) in free space.

In short, the core contributions of this article are as follows:

- The approach is data driven and non supervised, thus does not rely on any additional context or map information. The algorithms do not rely on expert selected parameters, thus exhibiting good accuracy and performance over highly skewed and non uniform datasets.
- The entire workflow is computationally efficient and distributable, thus capable of processing massive amounts of raw mobility data in minimal time.
- The approach is capable of producing representative trajectories that are complete (end to end) and smooth thus useful for further analytical processing tasks (e.g. anomaly detection, time of arrival estimation and others).

We demonstrate all the above properties on a real world dataset and compare the results with other state of the art approaches.

II. Related Work

Several seminal works have been proposed in order to reveal the hidden structure of a set of unlabeled data points, grouping in clusters similar subsets of points. Among different clustering approaches density-based clustering had been widely adopted. DBSCAN [2] groups together closely located data points, while points belonging to low density regions are annotated as outliers. OPTICS [1] extended DBSCAN by automatically adapting to various densities. This is achieved by ordering the points and considering the closest neighbors first. Another commonly used clustering approach is hierarchical clustering. For instance, BIRCH [3] builds a tree for the given data points. The density-based and the hierarchical clustering approaches, that were mentioned above, could be used in the trajectories domain considering the distances between the entire trajectories. In [5], [6] similar trajectories are grouped together into clusters. These techniques consider the overall distance between the entire trajectories. Such techniques are inadequate to handle real trajectories where moving objects follow different paths in order to reach the destination.

Several methods, that aim to discover moving patterns, assume that the objects are moving in *free space*, examples of such trajectories are animals' movements (e.g. birds), people hiking and vessels or planes trajectories. Lee et al. [4] proposed a trajectory clustering algorithm, named *TRACLUS* that discovers common subtrajectories from trajectories. In *TRACLUS* the trajectories are partitioned firstly into a set of subtrajectories, using the minimum description length (MDL) principle. Then the different subtrajectories are grouped into clusters introducing a clustering algorithm similar to DBSCAN. Also, a pipelined algorithm for clustering movement data was proposed by Gudmundsson et al. [7]. The algorithm splits trajectories in subtrajectories and provides labels for each subtrajectory according to its geometric property. Then, the trajectories are transformed in sequences of these labels used to detect frequently occurring strings (motifs). Finally, similar subrajectories are detected using the DBSCAN clustering algorithm. Cao et al. proposed in [8], an approach that transforms a trajectory in a sequence of segments and then a heuristic method searches for frequent patterns in the data, using a substring tree. The authors in [9] proved that the problem of finding subtrajectories' clusters is NP-Complete.

Additionally the problem of summarizing trajectories in corridors has been investigated in [10]. In order to extract the corridors they segemented trajectories into subtrajectories using a mesh grid, then they grouped subtrajectories into clusters using an agglomerative clustering algorithm that considers their discrete Fréchet distance, creating clusters of similar movement. Finally the corridors were the sequences of the detected clusters with similar starting/ending locations. Another technique that detects corridors that the moving

[1] `https://www.openstreetmap.org`

objects frequently traverse together was proposed in [11], partitioning the trajectories in subtrajectories taking into account spatial areas that are frequently traversed together. In [12] the trajectories are transformed into sequences of regions of interest and they found frequent patterns in these sequences considering the travel times.

The authors in [13] process AIS data in order to predict the vessel's behavior in the next 30 minutes proposing a clustering algorithm that uses the Karhunen-Loeve transform and Gaussian Mixture Models. A deep learning architecture that forecasts the future locations of the vessel considering its recent locations has been presented in [14]. A bidirectional LSTM model is used in combination with an attention mechanism to aggregate the past vessel's locations. A deep learning approach that forecasts the inflow and outflow of vessels at a particular area has been presented in [15] employing a bidirectional LSTM network in combination with a CNN network. In [16] a recurrent neural network is used that consists of an encoder network aiming to summarize the past movement of the vessel and a decoder network that forecasts the next position of the vessel. A model that predicts the ship's position at a given time along with the association probability between an existing track and a new message has been proposed in [17].

The complex nature of trajectories (i.e. sequences of coordinates) makes it difficult to estimate the distance or the similarity between two trajectories. The problem of measuring the similarity between two trajectories has attracted considerable research effort over the last years. Simple techniques like the sum-of-pairs distance [18] that assume that the two trajectories have the same number of points and the same sampling frequencies could not be applied in real settings where the moving objects move with different speeds and report their locations with different frequencies. In order to treat this problem Dynamic Time Warping (DTW) [19] technique was proposed aligning the positions of the trajectories and allowing multiple matches to the same point. A longest common subsequece (LCSS)-based model was proposed in [20] for efficient spatiotemporal queries in trajectories databases. Edit Distance on Real sequence (EDR) distance function between two trajectories was introduced in [21]. This function aims to reduce the effects of trajectories noisiness quantizing the distance between a pair of elements to two values, 0 and 1. Edit distance with Real Penalty (ERP) metric was proposed in [22] and can be viewed as a combination of EDR and L1-norm that assign penalties to the gaps between two matched trajectories.

III. Problem Description

In this work we develop an efficient technique that receives as input the current location of a particular vessel along with its destination port and forecasts the path that the vessel will follow till its arrival at the destination port. Our model is trained using a dataset of N historical vessel trajectories $\mathcal{D} = T_1, T_2, \dots, T_N$ of a particular route (i.e. a pair of origin and destination ports) and extracts a mobility graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that abstracts the vessels' movements. The vertices correspond to locations that are frequently traversed by multiple vessels and the edges favour the most likely movement among these vertices. Each trajectory $T_i : p_1\, p_2 \dots p_{M_i}$ is defined as a time ordered sequence of M_i consecutive coordinates $p_i \in \mathbb{R}^2$. The total travel time of the T_i is noted as $T_{i \cdot tt}$ Our formal definition is presented bellow:

> *Given the vessel's origin and destination ports along with the vessel's current query location p_q our task is to forecast the representative trajectory $\widehat{T_{p_q}}$ that the vessel will follow towards the destination port aiming to minimize the distance between $\widehat{T_{p_q}}$ and the actual trajectory T_{p_q} from the query location p_q towards the destination port.*

IV. Methodology

In this section, we describe the components of the proposed framework. Section IV-A describes the data preprocessing that has been followed. Section IV-B describes a technique for partitioning the locations considering sliding envelopes along the vessels' course. Section IV-C presents the grouping of locations in different clusters considering the vessels' heading and location inside each envelope. Section IV-D presents a technique for building a directed graph that summarizes vessels' transitions and models the connectivity among different clusters. Finally, Section IV-E describes the approach that is followed in order to extract the representative trajectory from a given query location towards a destination port.

A. Preprocess AIS Data

The data used in this study includes the following sources: *(i)* AIS data of passenger vessels moving in the Aegean sea for 1 year (January 2019 till December 2019) and *(ii)* port geometries as provided by the World Port Index from the National Geospatial Intelligence Agency [23]. Regarding the AIS data, there is an upper limit of 64 possible types of messages that AIS transceivers can exchange [24]. These message types may be related to vessel position tracking, vessel's identification or voyage information. In our study we focus on types 1-3, 18, 19 that are linked to tracking vessel positions and type 5 which comprises vessel identification and voyage information. Our AIS dataset contains approximately 5.5 million positions, broadcasted from more than 450 passenger ships. Although the collected data include all the required information to identify spatiotemporally the operation of each ship, significant processing is needed to extract additional value.

A ship journey begins and ends at a sea port (or an anchorage within or close to the port's operational area). An essential preprocessing task is assigning to all positional data collected through AIS, origin-destination information. Although AIS messages often include a destination port, this field is ignored in our study, as it is manually entered by each vessel's crew, without following a specific standard, making it thus prone to errors. For this purpose we recalculate destination and departure ports by making use of the World

Message (t-1)	Message (t)	Travelling Status	Port Move
In Port(A)	Not in Port	Departure from Port(A)	True
Not in Port	In Port(A)	Arrival at Port(A)	True
In Port(A)	In Port(A)	In Port(A)	False
Not in Port	Not in Port	Travelling at open sea	False

TABLE I: Port call events.

Port Index dataset, which contains the location and physical characteristics of major ports and terminals worldwide [23]. We calculate the operational boundaries of each port in a data driven method as described in [25], [26]. We execute a spatial query to assess intersections of port geometries (or operational areas) with vessel positions.

All the positions that intersect with a port geometry are assigned the corresponding geometry unique identifier (i.e., port id). Then, data are sorted per ship id and timestamp and for each consecutive pair of messages with the same ship id, changes in port id are detected, to determine port call events (i.e., departures / arrivals). As depicted in Table I, four different cases may occur for each pair of consecutive messages received. All vessel positions that are between departure and arrival time are considered as part of the same voyage. Following this, we follow several data cleaning steps such as identifying whether kinematic equations explain the dynamic positional reports for each vessel or evaluating for each data field whether it is complete and determining its integrity. These cleaning steps are beyond of the scope of the current paper, but are adequately documented in related papers such as [27] and [28].

B. Building Envelopes

In this section, we propose a sequence of steps that creates a set of envelopes sliding along the vessels' course. These envelopes will be used in order to partition vessels' positions.

1) Detect Baseline Trajectory: In order to detect the trajectories' course we select a baseline trajectory applying several filters in the set of trajectories $\mathcal{D}$ that followed a particular route. The candidate baseline trajectories are firstly selected by filtering out the trajectories with large travel times, according to equation 1 (i.e. consider only the trips with travel time less than the average travel time that is required to traverse the route). Our assumption is that trips with large travel times may be outliers containing parts where the vessel was stopped waiting to enter the anchorage or cases where the vessel faced severe weather conditions deviating from the normal movement. From the set of the remaining filtered trajectories $\mathcal{D}_f$ we select the baseline trajectory as the one that contains the minimum maximum haversine distance among consecutive points, according to equation 2. In this way we avoid the baseline trajectory to have large gaps, generated due to the sparsity of the samples and the non-uniform sampling rates. Finally, we smooth the baseline trajectory T_r, computing its spline curve $\tilde{T_r}$ using the B-Spline [29] approach and we set $\tilde{T_r}$ as the baseline trajectory.

Example. Consider the four trajectories illustrated at the left part of Figure 2, firstly we filter out the red trajectory since its duration exceeds the trips' average travel time. Then from the three remaining trajectories (central part of Figure) we select the one with the minimum maximum distance among consecutive samples. The baseline trajectory is finally the smooth curved line of the previously selected trajectory (red trajectory in the right of Figure 2).

$$\mathcal{D}_f = \{T_i : T_{i \cdot tt} \leq \frac{\sum_{j=1}^{N} T_{j \cdot tt}}{N}, T_i \in \mathcal{D}\} \tag{1}$$

$$T_r = \min_{\forall T_i \in \mathcal{D}_f} \max_{\forall k \in \{1 \ldots M_i - 1\}} haversine(p_k, p_{k+1}) \tag{2}$$

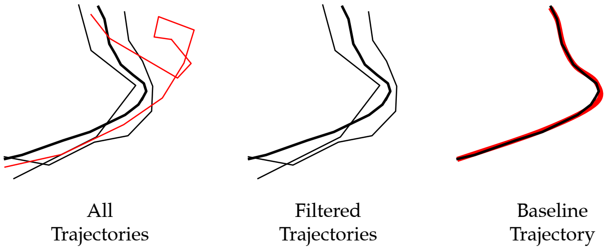


Fig. 2: Example of baseline trajectory extraction process.

2) Building the envelopes: Here we generate a set of envelopes, traversing the coordinates of the smoothed baseline trajectory $\tilde{T_r}$. More specifically, we traverse the coordinates of the candidate trajectories and we generate a rectangle with width w rotated in the direction of the vector that joins two consecutive points p_k and p_{k+1}, as it is illustrated in Figure 3.

In order to detect the coordinates of the envel e_1, e_2, e_3 and e_4 we first compute the angle of movement $\theta = arctan2(p_{k+1}.lat - p_k.lat, p_{k+1}.lon - p_k.lon)$. Then, we compute the vertical and horizontal distances dy and dx respectively from p_k and p_{k+1}, that will be used in order to compute the coordinates of the envel.

$$dy = \frac{w}{2} cos(\theta) \tag{3}$$

$$dx = \frac{w}{2} sin(\theta) \tag{4}$$

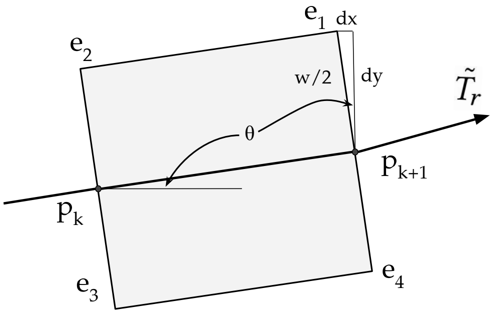


Fig. 3: An example of two consecutive points p_k and p_{k+1} and the generated envelope.

C. *Clustering Locations*

In this section we describe how the frequently followed locations inside each envelope are detected, grouping together the locations of the vessels that are spatially close.

Initially we create an envelope considering two consecutive points p_k and p_{k+1} of the baseline trajectory $\tilde{T}_r$, as it was described in section IV-B.2 and the we detect the vessels' reported positions that lie inside the envelope. If the number of points inside the envelope does not exceed a pre-settled threshold then the envelope is extended considering the next point of the baseline trajectory (i.e. build an envelope considering p_k and p_{k+2}). This process is iterated till the number of points inside the envelope exceeds the $maxEnvPoints$ threshold. In this way we avoid generating envelopes with a limited number of points in areas with limited sampling coverage. We selected $maxEnvPoints$ to be equal to the number of training trips of each route (i.e. each envelope will have approximately one point per trip).

In order to detect the frequently followed locations we project the vessels' positions in a line perpendicular to the direction of the vector that joins the points that form the envelope, as it is illustrated in Figure 4. Then we group together the closely packed together projected points using DBSCAN. This procedure detects a set of dense locations inside each envelope.

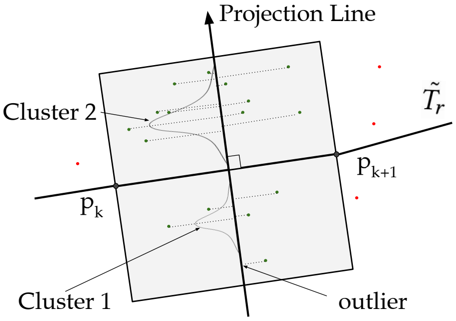


Fig. 4: *(i)* Projecting the points that lie inside the envelope on a line perpendicular to the direction of the baseline trajectory and *(ii)* Detecting clusters inside the envelope.

D. *Building a Directed Graph*

In order to capture the main mobility patterns we generate a mobility graph that depicts the connectivity among different envelopes' clusters. A directed edge-weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed for *each* envelope's cluster. The set of vertices $\mathcal{V}$ corresponds to the spatial area covered by the points inside each cluster and the set of edges $\mathcal{E}$ connects the clusters. Two different weights w_1 and w_2 are assigned to each edge $e = (v_1, v_2)$, depicting the number of transitions from v_1 to v_2 and favoring the most frequent path from v_1 to v_2 respectively.

Initially, the cluster and the corresponding envelope for each point of the trajectories $T_i \in \mathcal{D}$ is detected transforming T_i into T_i'. T_i' is defined as a sequence of envelope clusters $T_i' : C_1\, C_2\, \ldots\, C_{M_i'}$, where M_i' is the number of clusters of T_i'. If two or more consecutive coordinates of T_i are mapped into the same envelope cluster then we keep only the first instance, not allowing T_i' to have consecutive points of the same envelope cluster, meaning that $M_i' \leq M_i$ and that $C_k \neq C_{k+1} \forall k \in \{1, \ldots, M_i' - 1\}$. Finally, a new dataset $\mathcal{D}'$ of sequences of trajectories' clusters is generated, after iterating this process for each trajectory $T_i \in \mathcal{D}$.

Then, in order to generate the graph $\mathcal{G}$ for each $T_i' \in \mathcal{D}'$ each consecutive pair of envelope clusters C_k and C_{k+1} $\forall k \in \{1, \ldots, M_{i-1}'\}$ is parsed updating at each step the graph according to the following procedure:

- if the clusters C_k or C_{k+1} are not in the set of vertices $\mathcal{V}$ of the graph $\mathcal{G}$ then the missing clusters are added in the set of vertices $\mathcal{V}$.
- if there is not an edge connecting the vertices C_k and C_{k+1} (*i.e.* not in the set of edges $\mathcal{E}$ of $\mathcal{G}$) then a new edge connecting C_k and C_{k+1} is added setting the corresponding weight $w_1(C_k, C_{k+1}) \leftarrow 0$, which measures the number of connections between the two consecutive clusters.
- update the weight $w_1(C_k, C_{k+1}) \leftarrow w_1(C_k, C_{k+1}) + 1$.

Following that, a second weight w_2 is introduced for each edge of $\mathcal{G}$ that favors the most likely movement among clusters, considering the number of transitions from one cluster to another. More specifically, we iterate over each cluster C_k and we compute the total number of output transitions $C_k.out$ from C_k cluster towards any other cluster, according to equation 5. The weight w_2 is computed using equation 6 favoring the connections with more transitions in the historical data. In order to penalize the connections with clusters in remote envelopes d_{env} is introduced, d_{env} is defined as the distance between the envelopes of two consecutive clusters. For instance, if one cluster C_k is in the 5^{th} envelope and its successor C_l belongs in the 8^{th} then $d_{env}(C_k, C_l) = 3$. An example of how the weights w_2 are estimated from the transition weights is illustrated in Figure 5.

$$C_k.out = \sum_{\{C_l : (C_k, C_l) \in \mathcal{E}\}} w_1(C_k, C_l) \tag{5}$$

$$w_2(C_k, C_l) = d_{env}(C_k, C_l)\, \frac{C_k.out - w_1(C_k, C_l)}{C_k.out}, \quad \forall (C_k, C_l) \in \mathcal{E}, C_k \in Env_j \tag{6}$$

Finally, several edges are inserted connecting all the clusters C_k that belong to the last envelope Env_{last} with a $sink$ node with 0 weight, according to equation 7.

$$w_2(C_k, sink) = 0, \forall C_k \in Env_{last} \tag{7}$$

E. *Representative Trajectories*

In this section, we describe how the representative trajectories are forecasted considering a given query location and a destination port. Our algorithm is presented in algorithm 1. Firstly, the envelope cluster q_{cl} that is closest to the given query location q_{loc} is identified. Following that,

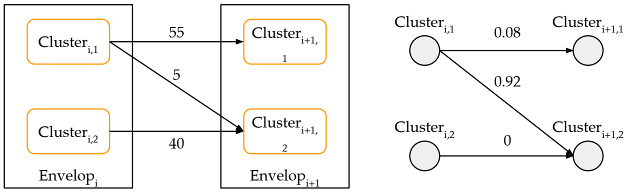


Fig. 5: An example of building the directed graph $\mathcal{G}$ (right), considering the transitions between the clusters of the different envelopes (left).

the shortest path from cluster q_{cl} towards the sink node considering the weight w_2 of $\mathcal{G}$ is computed. The shortest path contains the most likely sequence of envelopes' clusters from the given query location towards the destination port. Then, a $trajectory$ is constructed considering the centroids of all the envelopes clusters of the shortest path. Finally, the representative is created by smoothing the previously detected trajectory, using the B-Spline [29] approach, since the consecutive centroids of the forecasted trajectory are not always aligned.

Algorithm 1: Forecasting the representative trajectory

```
input : A query location $q_{loc}$
output: A representative trajectory $repr\_traj$
$q_{cl} \leftarrow$ Find the closest envelopes' cluster of $q_{loc}$;
$envelopes\_clusters \leftarrow$
  $\mathcal{G}.shortest\_path(q_{loc}, sink, weight = w_2)$;
$trajectory \leftarrow [\,]$;
for $envelope\_cluster \in envelopes\_clusters$ do
    $trajectory.append(envelope\_cluster.get\_centroid())$;
end
$repr\_traj \leftarrow b\_spline(trajectory)$;
```

V. Evaluation

In this section we present our experimental results that evaluate the effectiveness of the proposed technique. We first describe the experimental data. We then discuss the results for passenger vessels that moved in the Aegean sea for the entire 2019.

A. Experimental Setting

The dataset that was used was provided by MarineTraffic[2] and contains AIS messages during a year (*i.e.* entire 2019). We experiment with an AIS dataset retrieved from passenger vessels moving in the Aegean sea. The overview of the investigated routes is presented in the first columns of Table II. For each route we present the number of trips that are available in the dataset, along with the average and the standard deviation of trips' duration. In general, we observed that routes with larger standard deviation of duration tend to be more complex containing trips that follow different paths. We preprocessed the AIS dataset extracting the vessel trajectories between the origin and destination ports following the steps described in Section IV-A. For each trajectory we removed the part of trajectories that is inside the port and the anchorage, modeling the vessels' movement outside the ports boundaries. Finally, we used the 70% of the trips of each route for training and the rest 30% of the trips for the testing.

B. Comparison Techniques

In order to study the proposed technique's effectiveness we describe the performance of the following techniques:

- *E*NVCLUS: contains the representative trajectory that is forecasted by our technique, considering the shortest path from the most frequently visited cluster of the first envelope towards the $sink$ node.
- *T*RACLUS: contains the clusters of trajectories that were detected from the method that was introduced by Lee et al. in [4]. We evaluate the *T*RACLUS algorithm using all the trajectories that share the same origin and destination ports (*i.e.* same route). Since, *T*RACLUS could detect multiple clusters we are reporting the performance of the trajectory cluster with the lowest distance from the actual trajectory.
- *O*pen Street Map (OSM): contains the detailed path connecting two ports as it is displayed in Open Street Map[3]. For each OSM route we removed the part of the trip that is inside the port boundaries following a similar approach with the preprocessing of vessel trajectories.

Since the extracted trips of OSM and TRACLUS are usually sparsely sampled we interpolated the intermediate points in case that the distance between two consecutive points is very large.

C. Evaluation Metrics

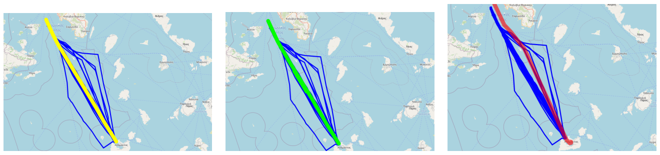
For evaluating the effectiveness of the proposed approach we used the DTW [19] in order to evaluate the distance between the actual and the predicted trajectories. More specifically, DTW is used to align the two trajectories (the actual and the forecasted). Then the distances between the matched points are computed in km, employing the haversine distance. Finally, the reported value is the average distance in km of all the matched points between the actual and the forecasted trajectory.

D. Results for Passenger Vessels

The overall performance of the proposed technique (ENVCLUS) in comparison to the baseline techniques is illustrated in Table II. Along with the average DTW distance, measured in km, between the actual and the predicted trajectories we present the percentage improvement (i.e. % impr.) that refers to the reduction of DTW distance that our technique achieves. Greater percentage improvement means that the representative trajectory that is produced by our

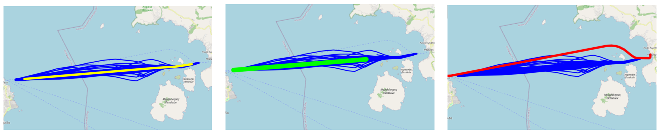
[2] https://www.marinetraffic.com

[3] https://www.openstreetmap.org



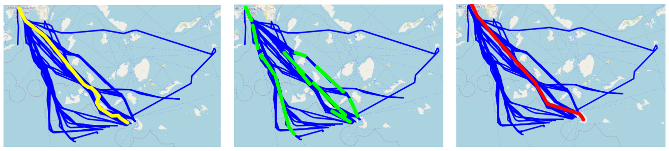
(a) ENVCLUS (b) TRACLUS (c) OSM

Fig. 6: Results for the route Piraeus → Milos



(a) ENVCLUS (b) TRACLUS (c) OSM

Fig. 7: Results for the route Rafina → Marmari



(a) ENVCLUS (b) TRACLUS (c) OSM

Fig. 8: Results for the route Piraeus → Santorini

technique is closer to the actual trajectory considering the baseline techniques. At the same time, Figures 6, 7 and 8 visualize the performance of the different techniques for 3 different routes.

Our technique in general outperforms the comparison techniques and avoids large DTW distances from the actual trajectories. More specifically, the total average percentage improvement of our technique, considering all the routes is 28% against TRACLUS and 27% against OSM.

TRACLUS in several cases detects multiple clusters of trajectories, as it described in Figure 8. In this case each cluster models only part of the entire trip. At the same time there are some parts where vessels moved that are not modeled by any cluster, as it is shown in Figure 6.Thus, we conclude techniques that detect clusters of trajectories are not able to model in detail the objects movement.

Also, we observed that the paths of OSM in several cases deviate considerably from the actual path that the vessels followed. For instance, as it is described in Figure 7 the vessels that moved towards Marmari port follow a different path from the OSM path. Also, OSM path contains a single path while the vessels may travel from a particular origin port towards a destination port through various paths.

Finally, in Table III we present the performance of our technique making queries at different parts of the test trajectories. As we mentioned earlier a benefit of our approach is that it is able to model the entire space where vessels moved. This differentiates us from OSM that provides a single path. For each test trip we generated 20 query points at different parts of the trips. In this way, PartA contains queries at the first $1/3$ of the trip, PartB contains queries at the second $1/3$ and finally PartC contains the queries at the last $1/3$. As we can see our approach is able to adapt to deviations from the main path reducing the distance from the actual trajectory as more information is provided regarding the path followed by the vessel. This observation is more obvious for complex routes where vessels tend to follow different paths from the origin port towards the destination port (i.e. Piraeus → Santorini and Chios → Mytilini). For these cases the comparison techniques might outperform our technique for route forecast queries from the origin towards the destination port (Table II), but we can see how the error is reduced in our technique as the vessel moves and reports its locations (Table III).

VI. Conclusion

In this paper we proposed a novel data driven framework capable of revealing representative trajectories from massive AIS datasets. To show the effectiveness of our approach we performed extensive experiments using real world datasets from passenger vessels moved in the Aegean sea. Overall we observed that our technique outperforms the comparison techniques in most of the testing routes. At the same time, our technique is able to model the entire space where vessels moved. Finally, we observed that our technique reduces the average distances with the actual trajectories in comparison to the baseline techniques more than 27% and that the distance between the forecasted representative trajectory and the actual trajectory is decreased as the vessel starts its trip.

Acknowledgement

This research is supported by European Union's Horizon 2020 research and innovation programme under grant agreement No 957237, project ENABLING MARITIME DIGITALIZATION BY EXTREME-SCALE ANALYTICS, AI AND DIGITAL TWINS (VesselAI).

References

[1] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

[2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[3] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

[4] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.

[5] Scott Gaffney and Padhraic Smyth. Trajectory clustering with mixtures of regression models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 63–72, 1999.

[6] Igor V Cadez, Scott Gaffney, and Padhraic Smyth. A general probabilistic framework for clustering individuals and objects. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 140–149, 2000.

[7] Joachim Gudmundsson, Andreas Thom, and Jan Vahrenhold. Of motifs and goals: mining trajectory data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 129–138. ACM, 2012.

Origin	Destination	#Trips	Avg Dur.	Std Dur.	ENVCLUS	TRACLUS	% impr.	OSM	% impr.
PIRAEUS	SANTORINI	158	9:27:29	5:20:45	15.25	22.30	31.62	13.02	-17.14
RAFINA	MARMARI	1031	1:03:51	0:03:30	0.69	1.26	45.56	1.71	59.89
SYROS	TINOS	388	0:40:15	0:07:03	0.72	2.26	68.21	1.11	35.32
CHIOS	MYTILINI	379	2:47:23	0:13:54	3.73	2.76	-35.12	2.20	-69.89
SYROS	PAROS	320	1:28:59	0:18:24	0.70	1.03	31.57	1.16	39.42
SYROS	MYKONOS	203	1:03:31	1:05:12	1.18	1.15	-2.51	1.83	35.70
SERIFOS	SIFNOS	174	0:51:02	0:05:13	0.60	1.43	58.15	0.89	32.88
SANTORINI	ANAFI	158	1:36:11	0:12:37	0.91	1.66	45.33	1.31	30.83
SIFNOS	KIMOLOS	139	0:59:21	0:03:06	0.53	2.25	76.71	0.66	20.86
PIRAEUS	MILOS	110	4:54:59	1:33:48	1.18	0.98	-20.69	5.62	78.99
KYTHNOS	SYROS	104	2:39:34	0:10:38	1.34	2.88	53.61	1.35	0.88
PIRAEUS	KYTHNOS	88	3:24:04	0:06:51	1.03	2.37	56.63	2.19	52.95
PIRAEUS	SOUDA	662	8:07:27	1:49:57	3.41	3.31	-3.00	3.88	12.27
PIRAEUS	HERACLIO	627	9:13:50	1:32:27	1.74	1.55	-12.29	6.37	72.63

TABLE II: Average DTW error of the evaluation methods for different vessel routes in the Aegean sea along with the percentage error improvement against TRACLUS and OSM

Origin	Destination	PartA	PartB	PartC
PIRAEUS	SANTORINI	15.8	12.8	3.9
RAFINA	MARMARI	0.7	0.7	0.6
SYROS	TINOS	0.7	0.8	0.9
CHIOS	MYTILINI	2.0	1.1	0.9
SYROS	PAROS	0.7	0.7	0.6
SYROS	MYKONOS	1.2	1.2	1.0
SERIFOS	SIFNOS	0.6	0.5	0.4
SANTORINI	ANAFI	0.9	0.8	12.5
SIFNOS	KIMOLOS	0.5	0.5	0.4
PIRAEUS	MILOS	1.2	1.1	0.8
KYTHNOS	SYROS	1.2	0.9	0.5
PIRAEUS	KYTHNOS	1.0	1.0	0.8
PIRAEUS	SOUDA	4.3	4.6	2.2
PIRAEUS	HERACLIO	2.1	1.9	0.7

TABLE III: Measuring the average DTW of *E*NVCLUS for queries at different parts of the trips.

[8] Huiping Cao, Nikos Mamoulis, and David W Cheung. Mining frequent spatio-temporal sequential patterns. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

[9] Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Maarten Löffler, and Jun Luo. Detecting commuting patterns by clustering subtrajectories. In *International Symposium on Algorithms and Computation*, pages 644–655. Springer, 2008.

[10] Haohan Zhu, Jun Luo, Hang Yin, Xiaotao Zhou, Joshua Zhexue Huang, and F. Benjamin Zhan. Mining trajectory corridors using fréchet distance and meshing grids. In *Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part I*, pages 228–237, 2010.

[11] Nikolaos Zygouras and Dimitrios Gunopulos. Corridor learning using individual trajectories. In *2018 19th IEEE International Conference on Mobile Data Management (MDM)*, pages 155–160. IEEE, 2018.

[12] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 330–339. ACM, 2007.

[13] Brian Murray and Lokukaluge Prasad Perera. Ship behavior prediction via trajectory extraction-based clustering for maritime situation awareness. *Journal of Ocean Engineering and Science*, 2021.

[14] Samuele Capobianco, Leonardo M Millefiori, Nicola Forti, Paolo Braca, and Peter Willett. Deep learning methods for vessel trajectory prediction based on recurrent neural networks. *arXiv preprint arXiv:2101.02486*, 2021.

[15] Xiangyu Zhou, Zhengjiang Liu, Fengwu Wang, Yajuan Xie, and Xuexi Zhang. Using deep learning to forecast maritime vessel flows. *Sensors*, 20(6):1761, 2020.

[16] Duc-Duy Nguyen, Chan Le Van, and Muhammad Intizar Ali. Vessel trajectory prediction using sequence-to-sequence models over spatial grid. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, pages 258–261, 2018.

[17] Jun Ye Yu, Moslem Ouled Sghaier, and Zofia Grabowiecka. Deep learning approaches for ais data association in the context of maritime domain awareness. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2020.

[18] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):1–41, 2015.

[19] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:, 1994.

[20] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. Indexing multidimensional time-series. *The VLDB Journal—The International Journal on Very Large Data Bases*, 15(1):1–20, 2006.

[21] Lei Chen, M Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD*, pages 491–502. ACM, 2005.

[22] Lei Chen and Raymond Ng. On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803. VLDB Endowment, 2004.

[23] World Port Index. `https://msi.nga.mil/NGAPortal/MSI.portal?_nfpb=true&_pageLabel=msi%20_portal_page_6&pubCode=0015/`, 2008. [Online; accessed 14-January-2021].

[24] M Series. Technical characteristics for an automatic identification system using time-division multiple access in the vhf maritime mobile band, 2010.

[25] Leonardo M Millefiori, Dimitris Zissis, Luca Cazzanti, and Gianfranco Arcieri. Scalable and distributed sea port operational areas estimation from ais data. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 374–381. IEEE, 2016.

[26] Leonardo M Millefiori, Dimitris Zissis, Luca Cazzanti, and Gianfranco Arcieri. A distributed approach to estimating sea port operational regions from lots of ais data. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 1627–1632. IEEE, 2016.

[27] Philipp Last, Christian Bahlke, Martin Hering-Bertram, and Lars Linsen. Comprehensive analysis of automatic identification system (ais) data in regard to vessel movement prediction. *The Journal of Navigation*, 67(5):791–809, 2014.

[28] Dimitris Zissis, Konstantinos Chatzikokolakis, Giannis Spiliopoulos, and Marios Vodas. A distributed spatial method for modeling maritime routes. *IEEE Access*, 8:47556–47568, 2020.

[29] Paul Dierckx. *Curve and surface fitting with splines*. Oxford University Press, 1995.