IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 December 2023, accepted 28 December 2023, date of publication 1 January 2024,
date of current version 10 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3349081

RESEARCH ARTICLE

EnvClus*: Extracting Common Pathways for Effective Vessel Trajectory Forecasting

NIKOLAS ZYGOURAS[1], ALEXANDROS TROUPIOTIS-KAPELIARIS[2], (Graduate Student Member, IEEE), AND DIMITRIS ZISSIS[2], (Senior Member, IEEE)

[1]MarineTraffic, 166 74 Athens, Greece
[2]Department of Product and Systems Design Engineering, University of the Aegean, Syros, Greece

Corresponding author: Alexandros Troupiotis-Kapeliaris (alextroupi@aegean.gr)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programs; in part by the Project "Enabling Maritime Digitalization by Extreme-Scale Analytics, AI and Digital Twins (VesselAI)" under Grant 957237; in part by the Project "Critical Action Planning over Extreme-Scale Data (CREXDATA)" under Grant 101092749; in part by the Program of Industrial Scholarships of Stavros Niarchos Foundation; and in part by Hellenic Academic Libraries Link (HEAL).

ABSTRACT The task of accurately forecasting the trajectory of a vessel, and in general a moving object operating in free space until its destination remains an open challenge. This paper addresses this problem by describing an unsupervised data-driven framework for short and extended horizon forecasts, from the perspectives of data mining and machine learning. We propose a data-driven algorithmic approach named "EnvClus*" that models efficiently historical vessel trajectories at a global scale, forming a mobility graph that depicts the most likely movements among two ports. EnvClus* is able to make tailored route forecasts considering the characteristics of the vessels (i.e. length, draught) along with information regarding the executed trip. The proposed method is able to forecast the most likely realistic and smooth trajectory from a given query position of a vessel (entire route or underway forecasting) towards its destination port. We illustrate the accuracy and effectiveness of our method through a series of scenarios for long and short term forecasting using real world data from around the globe. These experiments indicate an overall improvement of 33% over state-of-the-art and baseline methods; with the benefits of our approach being more apparent when dealing with longer trips from container vessels.

INDEX TERMS Trajectory prediction, mobility analytics, vessel route forecasting, trajectory clustering.

I. INTRODUCTION

Smart technologies and sensors are gradually being adopted throughout the transportation chain and especially sea transport. Monitoring vessel movement has been made possible by the Automatic Identification System (AIS) [1], with positional messages transmitted from hundreds of thousands of vessels worldwide each day. While knowing a vessel's current position is important, the ability to accurately forecast its trajectory into the future, is vital for a wide range of stakeholders across the industry. Applications of having an accurate estimation of the vessels' future paths range from handling port traffic congestion to avoiding accidents at sea. Nowadays the fields of "mobility analytics" and "computing with spatial trajectories" have emerged, focusing on computational methods capable of deciphering the mobility patterns of various objects (such as people, vehicles, airplanes etc.). Consequently, the research community has dedicated significant efforts over the past few years to solving the issue of moving object trajectory forecasting [2], [3], [4], [5].

Accurate route forecasting leads to better route planning, a critical component of safe and efficient maritime operations. Having a reliable route forecast allows for ports to accurately estimate the time of arrival of vessels, thus improving port operations efficiency. Additionally, through combining the predictions for different vessels, an estimation for the traffic of an area in a specific future time can be extracted. Such information allows for better

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Liu.

voyage planning, resulting in turn in more efficient and environmentally friendly movement from the vessels' part. Finally, Vessel Traffic Services and other ships may use these forecasts to improve the overall situational awareness and detect current and future dangers (e.g. collisions).

In contrast though with other areas, the maritime domain exhibits several unique challenges that limit the applicability of general-purpose methods. The first challenge in respect to general purpose approaches, is related to the nature of the AIS data in real world conditions. The non-uniform transmission rate, noise and network coverage issues induce both spatial and temporal uncertainties [6], [7], [8] differentiating vessel movement monitoring from other GPS usecases. Secondly, unlike the automotive domain, marine vessels operate in ''free space'' unconstrained by road networks, while heavily affected by external conditions (such as bad weather). Vessel trajectories can be understood as constrained only in specific areas where sea separation schemes are enforced (e.g port entry areas, heavy traffic areas etc.). Shipping routes form a dense network of port to port connections but are far from static over time. 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) and can completely disappear on various occasions.

As a result, nautical charts used for marine navigation, that amongst others contain paths connecting different ports, often face outdated issues. Additionally, such maps (i.e. Open Street Map[1]) 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. 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 an extra 3,800 miles to their journey and up to 12 days extra sailing time. On this occasion, the majority of route forecasting and time of arrival forecasting algorithms used in of the shelf systems, failed to adapt their forecasts, as they relied on traditional routing algorithms and cartography, not including alternative routes to a given destination. The current weather conditions and vessel characteristics, also highly affect a captains decision on which route to follow as depicted in n Figure 1, where the path taken may differ significantly depending on several factors.

While several works regarding route forecasting in the maritime domain have been proposed over the years [3], [9], most approaches focus on short-term predictions. Although some of these approaches are very accurate for the first 20 or 30 minutes, they suffer in predicting the vessels' route after 1 hour. Hence tasks that focus on long-term events (e.g., Estimation of Time of Arrival, Total amount of fuel consumption) can not be solved using such mechanisms

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

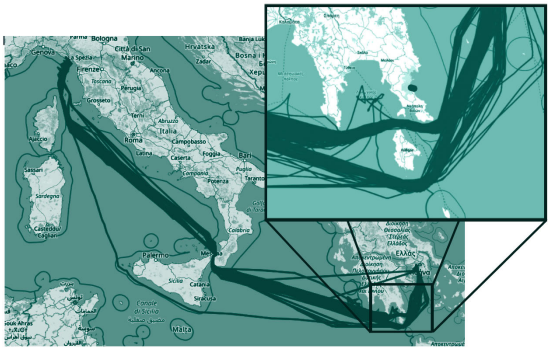

FIGURE 1. Illustration of the *Piraeus-La Spezia* trip, from the year of 2019, where the path taken by the vessels near the island of Kithira may differ significantly depending on the vessel or the journey in question.

and require a separate approach. Moreover, we can see that the majority of current approaches are based on machine-learning, and especially neural networks, thus requiring large amounts of data for training. In order not to limit the size of training datasets, it is common practice that generic models are generated, while more often than not additional features included in the AIS messages are being ignored. This often results in models that do not consider the vessel's destination or characteristics, providing predictions in line with common behavior, without focusing on the journey in question. For the purpose of overcoming such issues, some recent works incorporate the past movement of the vessel during the prediction [10], [11], thus requiring multiple positions for each query, without though taking advantage of any static information within the AIS.

Overall, this work provides a framework capable of forecasting the full path of a vessel from a given query location until its destination port. To accomplish this, we propose a data driven algorithmic approach named ''*Envclus**'' that efficiently models historical vessel trajectories at a global scale, forming a mobility graph that depicts the most likely vessels' movements among two ports (origin/destination). Since the proposed algorithm does not rely on expert selected parameters, it can be applied on highly skewed and non-uniform datasets, exhibiting good accuracy and performance. The focus of our work is on designing a real world solution with the desired properties of practicality and accuracy.

The contributions of this work are as follows:

- A data driven and non-supervised method for extracting common pathways of movement of vessels between two ports. As a result, for each origin-destination ports pair, a mobility graph is created encapsulating patterns of vessel movement in the area.
- A method that can forecast the full route of a vessel - from the origin port - while at the same time is capable to answer to queries submitted while the vessel is underway, returning the predicted future positions until the end of its trip. Considering the vessels' and the trips' characteristics, the proposed method is able to perform tailored route forecasts for each particular vessel.

We improve our previous work (*EnvClus* [12]) with *Envclus**, in the following directions:

- *EnvClus** returns vessel-specific forecasts through the use of classification models, while *EnvClus* provides the same route forecast for every query vessel.
- *EnvClus** is able to capture in detail the entire space where vessels moved considering multiple baseline trajectories, in contrast to *EnvClus* that only considered one.

An empirical evaluation of our method is performed, using real-world data for both passenger and container vessels. The evaluation focuses on entire trajectory forecasts, underway forecasts and short-term trajectory forecasting, using state-of-the-art implementations as baselines.

The remainder of this paper is organized as follows: first we present related work for route forecasting techniques (Section II). Afterwards, the components of our proposed method are described in detail (Section III) and the experimental evaluation is presented (Section IV). Finally, we conclude our work with a brief summary where we also provide possible future steps for further improving our approach (Section V).

II. RELATED WORK

A. TRAJECTORY DATA MINING

Data mining from trajectory data has been the focus of several works in the last few years [13], [14], especially in the big data era where rich spatio-temporal data allows for more extensive analysis [15], [16], [17]. One of the main purposes of such studies is to better understand and model the behaviour of moving objects. This can be interpreted as extracting moving patterns and may be achieved through different techniques, such as clustering [18], [19] or classification [20]. As an example, the technique proposed in [21] constructs a traversal graph by using the road segments that have been traversed by some reference trajectories. Another technique that extracts a digital map with rich knowledge from an unstructured GPS point cloud of moving vehicles was proposed in [22]. The road segments are detected using novel graph-based clustering techniques.

B. FREE SPACE MOVEMENT PROCESSING

Furthermore, several methods aimed at discovering moving patterns can be found in literature for objects moving in *free space*. Examples of such movement are hiking activities or the trips performed by vessels and planes. Lee et al. [23] proposed a trajectory clustering algorithm, named *TraClus* that discovers common parts from multiple trajectories. In *TraClus* the trajectories are partitioned firstly into a set of *subtrajectories*, using the minimum description length (MDL) principle. Then the different parts are grouped into clusters introducing an algorithm similar to DBSCAN [24]. Also, a pipelined algorithm for clustering movement data was proposed by Gudmundsson et al. [25], where the trajectories are split and a label for each subtrajectory is annotated, according to its geometric property. Then, the trajectories are transformed into label sequences and a method for detecting frequently occurring strings (motifs) is applied. In the final step, similar subrajectories are detected using the DBSCAN algorithm. In [26] the authors partition the space using a grid and introducing a graph that summarizes the trajectories movements among the cells. Additionally, the problem of summarizing trajectories into *corridors* (*i.e.* passages where common movement has been noticed) has been investigated in [27]. In order to extract these corridors they segmented the trajectories using a mesh grid and group the parts into clusters, using an agglomerative algorithm that considers their discrete Fréchet distance. In the end, the corridors returned were the sequences of the detected clusters with similar starting/ending locations. Another technique that detects corridors where the moving objects frequently traverse together was proposed in [28], partitioning the trajectories in subtrajectories taking into account spatial areas that are frequently traversed together.

C. MARITIME DATA MINING

Trajectory mining techniques have been widely applied in the maritime domain too [9], [29]. For instance, an unsupervised technique that detects roads of sea has been proposed in [30], clustering the vessels positions as they are reported through AIS messages. Other attempts in extracting common pathways for vessel movement can be found in [31], where roads are extracted by grid merging, and in [32] through the use of trajectory clustering and statistical analysis. The framework described in [33] focuses mainly on container and tanker vessel activity, to discover commercial routes and determining shipping trends, using loads of historical AIS messages, as well as vessel-specific information on gross tonnage. A pipeline for extracting shipping lanes from large AIS datasets is proposed in the work by Kontopoulos et al. [34]. This approach is able to model vessel movement, as common pathways between waypoints, through a novel clustering technique based on DBSCAN that takes the speed and the course attributes of the moving vessels into account. The inclusion of the Lagrange method for interpolating the trajectories allows for the approach to handle even sparse AIS data. The end results include a representation of common vessel traffic in an area of interest, regardless of the vessels' destinations or characteristics.

In this work, the proposed framework is able to extract common pathways between specific origin and destination ports, providing a more accurate representation of the vessel movement for such trips. Moreover, other ship or voyage characteristics are incorporated into the extracted mobility graphs in the form of classification models. Finally, our framework uses these pathways in order to provide accurate forecasts of the path the vessel will follow. The latter, i.e. forecasting future movement of vessels, is probably one of the most popular problems in trajectory maritime analytics in recent years [3]. As an example, the authors in [35] 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.

1) DEEP LEARNING FOR VESSEL TRAJECTORY ANALYSIS

With the rise of neural networks in the past few years, several works utilize deep learning techniques to predict the behavior of vessels. These works may focus on different aspects of movement at sea from predicting a vessel's heave motion [36] to providing an estimate about future vessel traffic flow over an area [37], [38], [39]. Similarly, a number of frameworks that take advantage of their capabilities for effective trajectory forecasting have been proposed in recent years [40]. While there are different types of neural networks, the most prominent works for route forecasting are based on Recurrent Neural Network architectures (RNNs). These works take advantage of the memory capabilities of the RNNs, dealing with the trajectory as a data series, and effectively consider the past movement of a vessel for their predictions.

A specific form of RNNs, the Long Short-Term Memory (LSTM) architecture, is able to keep an internal state after processing a data row. This way, the LSTM is able to encode the vessel's state of movement and provide informed predictions, making it a viable solution for the route prediction problem. Chondrodima et al. [41] showcased the efficiency of LSTM networks compared to other techniques, through an experimental comparison over AIS data from the areas of Brest and the Aegean. For a more complex approach, the network proposed in [42] consists of an encoder network aiming to summarize the past movement of the vessel and a decoder network that forecasts the next positions of the vessel. A variation of the LSTM network, called bidirectional LSTM (Bi-LSTM), has been used recently in some state-of-the-art approaches to achieve even better accuracy. First, in [10] an the encoder-decoder architecture allows for forecasting the future locations of the vessel considering its recent movement. During the encoding phase, the Bi-LSTM model is used to extract appropriate information from the vessel's recent movement. The results of this network appropriately enrich the input sequence of positions through an attention mechanism, and allows for a final LSTM structure to make a prediction of future movement for an horizon of up to three hours. On the other hand, Yang et al. rely largely on a denoising method to prepare the data for their Bi-LSTM network in order to predict future movement [43]. Finally, in an attempt create a less demanding architecture, a new model based on Gated recurrent units (GRUs) was proposed in [44]. An experimental evaluation conducted on the Chongqing and Wuhan sections of the Yangzi River indicated that the proposed technique is more beneficial in terms of training time and accuracy compared to a similar approach based on LSTMs.

Although these approaches are highly accurate in predicting the movement of a vessel, they do not provide full forecasts of each trip until its destination. Additionally, none of the works mentioned consider the characteristics of the vessels in their algorithms and provide a common forecast for all types of ships. Moreover, even if the LSTM-based approaches have proven to be effective, they all require a number of past positional messages from the vessels in order to extract an accurate prediction. The proposed approach in this work, overcomes these issues by providing full path forecasts according to the vessel and voyage characteristics and requiring only its current position.

III. METHODOLOGY

In this section the components of our framework, as seen in Figure 2, are presented.

Firstly, a preprocessing step selects and transforms the raw AIS messages to create a training dataset with all trajectories for a specific port pair (origin-destination) (Section III-A). Then, using a selected baseline trajectory, we extract a corridor geometry, by clustering the positional messages along its path (Section III-B). This last process is repeated for all remaining trajectories until the full dataset is covered (Section III-C). Afterwards, we model the vessels' movement by building a directed graph that depicts the likelihood of moving between these corridors (Section III-D1). We enrich this graph by training classification models upon its edges (i.e. the transitions between corridors) to adjusts appropriately the weights of the graph for each given query vessel (Section III-D2). After creating the model, a mechanism for providing a route forecast based on the vessel's current position, its selected features and the journey's characteristics (Section III-E).

A. AIS DATA PREPROCESSING

With multiple gigabytes of AIS data produced worldwide everyday, a preprocessing mechanism is required to filter and select the appropriate data for our model training. This process (described also in [30]) includes the removal of erroneous or duplicate messages, noise filtering and trip extraction. More precisely, the first step simply removes messages with erroneous or empty positional and movement fields. Then, a filtering mechanism removes messages that indicate improbable transitions. In order to do so, for all consecutive messages of the same vessel, the mean speed required for the subsequent transition is calculated. Messages indicating a speed beyond normal movement (i.e. over 50 knots) are discarded. Finally, the AIS messages are partitioned based on the vessel trip they belong. This was accomplished considering the geometries of the different ports around the globe, resulting in an origin-destination pair for each trip.

B. DISCOVERING A SINGLE CORRIDOR

In this section we describe an algorithm that detects a corridor of movement along a selected baseline trajectory. Initially, a baseline trajectory is selected from a set of given trajectories. Then, a set of envelopes are generated along the baseline trajectory and a set of clusters are detected for

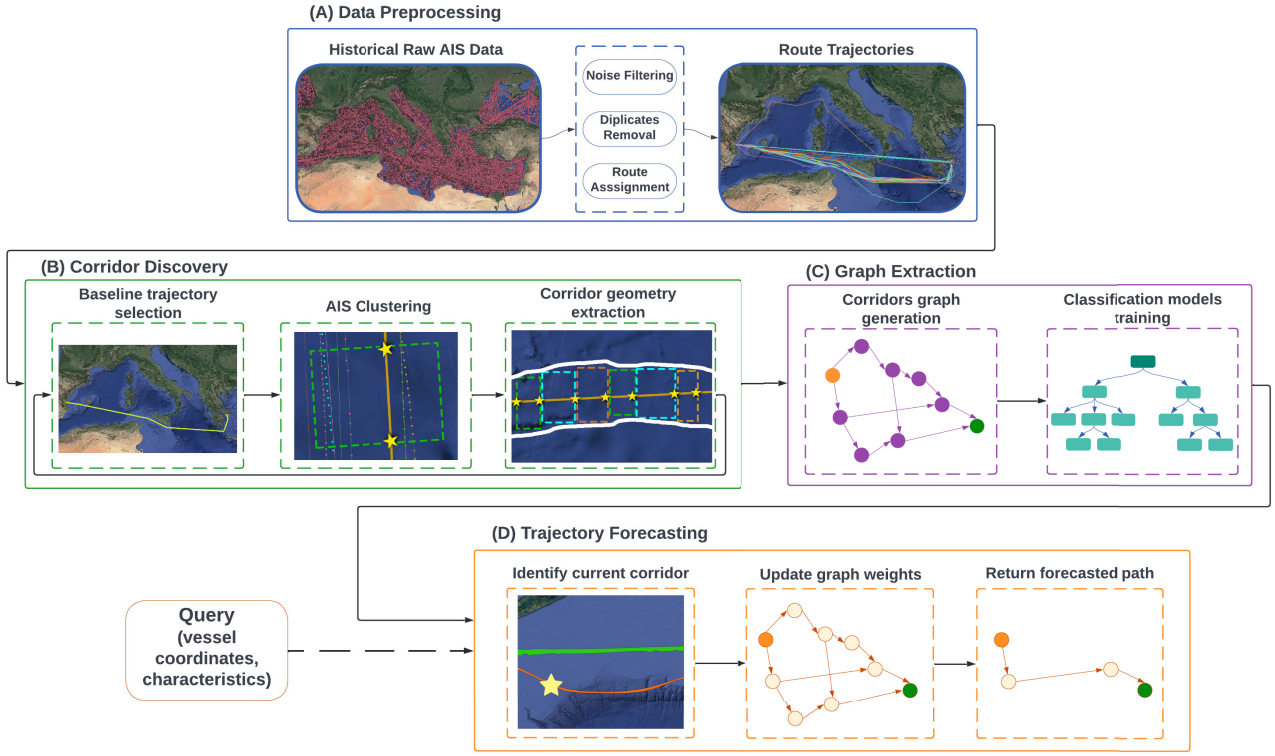

FIGURE 2. EnvClus* components; beginning with the preprocessing of the data (A) to the extraction of the corridors (B) and their enrichment with classification models regarding the vessels behavior (C). Finally, a query point is used to provide a route forecast until the destination port (D).

each envelope. Finally, a corridor is constructed from the bounding points of the clusters that surround the positions of the baseline trajectory. More precisely our technique consists of the following steps.

1) BUILDING THE ENVELOPES

First, we select one of the historical trajectories as a baseline, guiding us through this process. The shortest, in terms of duration, trip is selected in order to minimize noise coming from outlier trajectories. Then, a set of envelopes is created, traversing the coordinates of this baseline trajectory $\tilde{T}_r$. More specifically, we traverse two consecutive coordinates p_k and p_{k+1} of the baseline trajectory and we create a rectangle with width w rotated in the direction of the vector that joins p_k and p_{k+1} (Figure 3).

In order to detect the coordinates of the envelope 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 $dy = \frac{w}{2} cos(\theta)$ and $dx = \frac{w}{2} sin(\theta)$ respectively from p_k and p_{k+1}, that will be used in order to compute the coordinates of the envelope.

2) CLUSTERING LOCATIONS

Density based clustering is employed in order to group together the vessels' positions that are spatially close. This way we detect the spatial areas that are frequently traversed by multiple vessels inside each envelope.

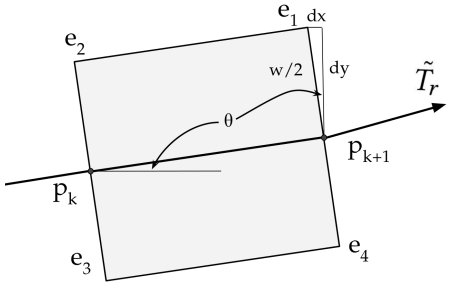

FIGURE 3. An example of two consecutive points p_k and p_{k+1} and the generated envelope.

Initially, an envelope is created considering two consecutive points p_k and p_{k+1} of the baseline trajectory $\tilde{T}_r$, as it was described in section III-B1. Then we detect the vessels' reported positions that lie inside the envelope and follow the same direction as the direction of the baseline trajectory. In order to detect these points, we first single out the vessels' positions that intersect spatially with the geometry of the envelope, considering only the positions whose moving direction is not greater than θ_{max} degrees from the direction of the baseline trajectory. For this threshold we could consider angles between 0 and 90 degrees on each side,

since any object with a larger difference could be considered going the opposite way from the baseline trajectory. For smaller thresholds, more corridors will be included in the end result, since all trajectories should be covered in the end. In our experiments we use a medium threshold of 45 degrees (on each side). 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 spatially close projected points using DBSCAN. This procedure detects a set of dense locations inside each envelope, considering the radius ϵ of a neighborhood around each point.

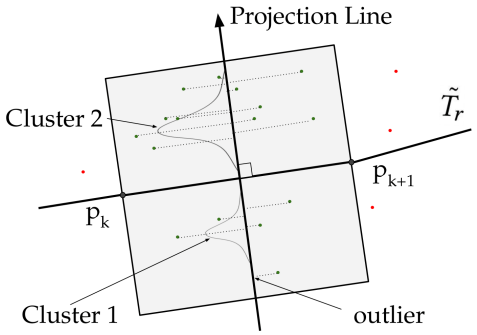

FIGURE 4. Example of Projecting the points that lie inside the envelope on a line perpendicular to the direction of the baseline trajectory and Detecting clusters inside the envelope.

3) DISCOVERING THE CORRIDOR

For each envelope we detect the bounding points (*i.e.* a minimum and a maximum point along the envelope's projection line) of the cluster that surrounds the baseline trajectory, ignoring the other clusters of the envelope. For instance, Figure 5 presents the envelopes and the corresponding clusters that are detected considering a baseline trajectory. We will only consider the clusters that surround the baseline trajectory, colored green in Figure 5.

Our next step, is to generate the corridor that summarizes the vessels movement across the baseline trajectory. The overview of the proposed technique is presented in Algorithm 1. In order to export the corridor we generate two linestrings: the first is generated by concatenating the minimum points of the envelopes' clusters that are associated with the baseline trajectory, while the second is generated similarly by the maximum points of the same clusters.

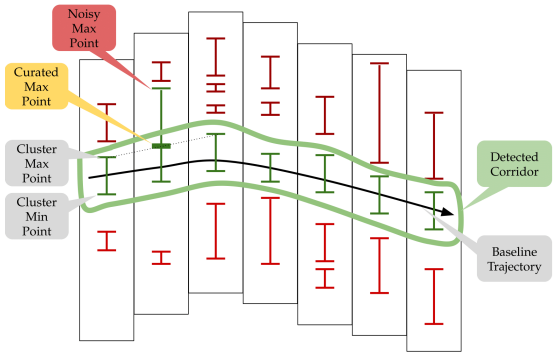

FIGURE 5. An example of detecting a corridor considering the clusters, of the different envelopes, that surround the baseline trajectory.

The clusters' bounding points were noisy in several cases, due to unusual vessel movement. These unusual movements resulted to the undesirable fluctuation of the clusters' bounding points and the union of two different clusters inside the envelope. Therefore, we followed a cleaning step that considers the sequence of bounding points of all the envelopes. Our technique first detects the clusters with large fluctuations of the bounding points. For each envelope we compute the distance between the clusters' bounding points and the baseline trajectory $dist_i$, $i \in [1, \ldots, |Envelopes|]$. Then, we estimate how the distances differentiate among consecutive envelopes, $dist'_i = dist_i - dist_{i-1}$, $i \in [2, \ldots, |Envelopes|]$. Following that, we annotate the envelopes that contain sudden rises or drops in their bounding points, detecting those envelopes that have absolute $dist'$ value greater than the standard deviation of all absolute $dist'$ values. Finally, we set as noisy all the bounding points of the envelopes that have a sharp rise (drop) which is followed by a sharp drop (rise) in at most the next 4 envelopes. Finally, we curate these noisy points, by replacing them with the intersecting point between *(i)* the projection line of the envelope and *(ii)* the previous and the next not-noisy bounding points. This is illustrated in Figure 5 where the maximum bound of the second envelope's cluster is annotated as noisy and is replaced by a new curated point.

The next step towards the detection of the corridors is to extend by ϵ the boundaries of the detected clusters, for each envelope. This gives the flexibility to our technique to capture the points near the edges of the corridor. Finally, a corridor is generated by concatenating and smoothing (using the B-Spline [45] technique) the detected *(i)* minimum and *(ii)* maximum points of the clusters. Figure 5 depicts a corridor that connects the extended minimum and maximum points of envelopes' clusters.

Finally, the area covered by the detected corridor is partitioned again into a *new set of envelopes*. As it is mentioned above, the envelopes do not have a fixed length and they are generated considering the number of points they cover. Therefore, we decided to partition each corridor in multiple *new envelopes* of fixed length considering the course of the baseline trajectory. More specifically, we traverse the

baseline trajectory and at fixed length intervals we compute the intersections between the corridors' boundaries and the line that is perpendicular to the baseline trajectory.

Algorithm 1 Detecting a Corridor That Covers the Baseline Trajectory

input : A baseline trajectory $\tilde{T}_r$, a trajectories dataset $\mathcal{D}$, a scalar ϵ
output: A *corridor* covering the baseline trajectory
envelopes_clusters ← Perform envelopes clustering considering $\tilde{T}_r$ & $\mathcal{D}$;
envelopes_clusters$'$ ← Select the *envelopes_clusters* that surround $\tilde{T}_r$;
for *bound* ∈ *envelopes_clusters*$'$.*bounding_points* **do**
  noisy_clusters ← Find the clusters of *envelopes_clusters*$'$ that contain a noisy *bound*;
  envelopes_clusters$'$ ← Curate the *noisy_clusters*;
  envelopes_clusters$'$ ← Extend the *bound* of *envelopes_clusters*$'$ by ϵ;
end
corridor ← *envelopes_clusters*$'$;

C. DETECTING MULTIPLE CORRIDORS

In this section we describe an algorithm for detecting multiple corridors of movement that cover the entire given dataset, employing iteratively the one presented in Section III-B. The overview of this algorithm is presented in Algorithm 2. Here we start with a baseline trajectory and we detect the corridor that covers it. Then, we remove all the points that lie inside each envelope and the trajectories whose points have been covered by any detected corridor. We then search for another baseline trajectory from the set of remaining trajectories. From the selected baseline trajectory we only consider its parts that do not overlap with any of the existing corridors and for each part of the baseline trajectory we compute its corridor. In turn, we only consider the area from the detected corridor that does not intersect with any other corridor, not allowing overlapping corridors. This process is repeated till all the trajectories have been covered, resulting in multiple corridors (as depicted in Figure 6).

Algorithm 2 Detecting Multiple Corridors From a Collection of GPS Trajectories

input : A set of GPS trajectories $\mathcal{D}$
output: A set of *corridors*
$\mathcal{D}' \leftarrow \mathcal{D}$;
corridors ← [ ];
while $\mathcal{D}' \neq \emptyset$ **do**
  $\tilde{T}_r$ ← Select a baseline trajectory from $\mathcal{D}'$;
  $\tilde{T}_{no\_intersect}$ ← Find parts not intersecting with *corridors*;
  for $\tilde{t}_r \in \tilde{T}_{no\_intersect}$ **do**
    corridor ← find corridor of $\tilde{t}_r$ considering $\mathcal{D}'$;
    $\mathcal{D}'$ ← Remove from $\mathcal{D}'$ the positions that lie inside the *corridor*;
    corridor$'$ ← find the part of the *corridor* that do not intersect with any of the *corridors*;
    corridors.*add*(*corridor*$'$);
  end
  $T_{covered}$ ← Find the trajectories in $\mathcal{D}'$ that are fully covered by the *corridors*;
  $\mathcal{D}'$.*remove*($T_{covered}$);
  $\mathcal{D}'$.*remove*($\tilde{T}_r$);
end

D. MODELING VESSELS MOVEMENT

1) BUILDING A DIRECTED GRAPH

In order to capture the vessels' mobility patterns, we generate a mobility graph that depicts the connectivity among the *new envelopes* of the corridors. A directed edge-weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed. The set of nodes $\mathcal{V}$ corresponds to the spatial area of the corridors with transitions to other corridors and the set of edges $\mathcal{E}$ connects the envelopes. In order to detect the nodes of $\mathcal{G}$ we identify the envelopes with transitions to other corridors. With this approach, we avoid adding nodes and edges in consecutive envelopes were there are no transitions to other corridors. In case that consecutive envelopes of one corridor are connected to consecutive nodes of another corridor then we only maintain one connection between the two corridors considering the first envelopes of the two corridors. Three different weights *dist*, *transitions* and w are assigned to each edge $e = (v_1, v_2)$. *dist* depicts the distance between the two envelopes, *transitions* indicates the number of transitions from v_1 to v_2 and the weight w favors the most likely path from v_1 to v_2.

Initially, the envelope and the corresponding corridor 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 corridors' envelopes $T_i' : C_1\, C_2 \ldots C_{M_i'}$, where M_i' is the number of envelopes of T_i'. If two or more consecutive coordinates of T_i are mapped into the same envelope then we keep only the first instance, not allowing T_i' to have consecutive points of the same envelope, meaning that $M_i' \leq M_i$ and that $C_k \neq C_{k+1} \forall k \in \{1, \ldots, M_i' - 1\}$. Also, in several cases T_i' connects non consecutive envelopes, due to data sparsity. Therefore, we enrich T_i' with all the envelopes that are between two consecutive points. Finally, a new dataset $\mathcal{D}'$ that contains the sequences of trajectories' envelopes is generated, after iterating this process for each trajectory $T_i \in \mathcal{D}$.

The weight w is introduced for each edge of $\mathcal{G}$. It favors the most likely movement among envelopes, considering the number of transitions from one envelope to another and the distance covered by the edges. More specifically, we iterate over each detected envelope that connects different corridors C_k and we compute the total number of output transitions $C_k.out$ from C_k envelope towards any other envelope. The weight w is computed using equation 1 favoring the

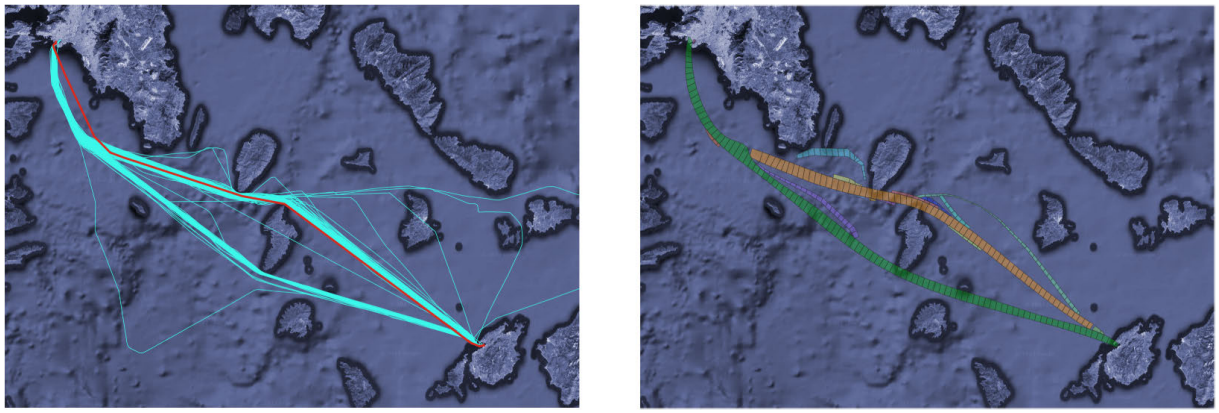

FIGURE 6. The raw vessel movement for the trips from Piraeus to Paros in light blue and the corresponding trajectory provided by Open Street Map in red (left). Our approach is capable of extracting multiple pathways of movement, modelling vessel movement more accurately (right).

connections with smaller distance and more transitions in the historical data.

$$w(C_k, C_l) = \frac{dist(C_k, C_l)}{transitions(C_k, C_l)}, \forall (C_k, C_l) \in \mathcal{E} \quad (1)$$

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 2.

$$w_2(C_k, sink) = 0, \forall C_k \in Env_{last} \quad (2)$$

2) ENRICHING THE GRAPH

Here, we generate classification models at branch nodes of $\mathcal{G}$, as vessels tend to follow specific corridors for their trips, according to their characteristics. There are several factors that could determine the vessel's path, as for instance, the vessel's dimensions and draught could pose restrictions regarding the path it will follow towards the destination port. These models consider the vessel's characteristics along with the temporal characteristics of each trip.

For each node a dataset is generated considering the vessels that departed from the corridor and those that remained. More precisely, using the historical data, the vessel transitions concerning each node are grouped based on whether the vessel remained at or left the corridor. Using these groups, as labels, and including vessel-related features (i.e. vessel ID, vessel type, vessel dimensions and draught), along with temporal features (i.e. hour of day, day of week and month), we generate the training dataset. Finally, using these data we are training a classification model that would decide whether a vessel will remain at the corridor or not.

In this work, two classification models were chosen for experimentation: Decisions Trees [46] and Support Vector Machines (SVM) [47]. The first creates a tree structure based on the different features selected in order to classify an input instance; in our case to decide the most probable corridor the vessels is going to follow. The same goal is achieved by the SVM, by creating a set of hyperplanes for the input feature vectors and generating classification results accordingly. For each selected branch node both types of models are trained, and taking into account a validation set, the most accurate is included within the final model.

E. ROUTE FORECAST

In this section we describe how a route forecast is generated for a given query location of a specific vessel. Our algorithm is presented in algorithm 3. Firstly, the corridor's envelope q_{env} that is closest to the given query location q_{loc} is detected. Then, a temporary graph $\mathcal{G}_{temp}$ is initialized, considering the predictions of the classification models that were trained earlier and penalizes the movement at the not selected corridors. Following that, the shortest path from the detected envelope q_{env} towards the sink node considering the weight w of $\mathcal{G}_{temp}$ is computed. Finally, a *trajectory* is generated considering the centroids of all the envelopes clusters of the shortest path.

Algorithm 3 Detecting the Representative Trajectory

input : A query location q_{loc}
output: A representative trajectory *repr_traj*
$q_{env} \leftarrow$ Find the closest corridors' envelope of q_{loc};
$\mathcal{G}_{temp} \leftarrow$ Copy of $\mathcal{G}$ with penalties at the not selected branches according to the classification models;
envelopes_clusters $\leftarrow$ $\mathcal{G}_{temp}.shortest\_path(q_{loc}, sink, weight = w_2)$;
trajectory $\leftarrow$ [ ];
for *envelope_cluster* $\in$ *envelopes_clusters* **do**
    trajectory.append(*envelope_cluster.get_centroid*());
end
repr_traj $\leftarrow$ *b_spline*(*trajectory*);

IV. EVALUATION

In this section we present an experimental evaluation of our proposed technique; incorporating three evaluation scenarios

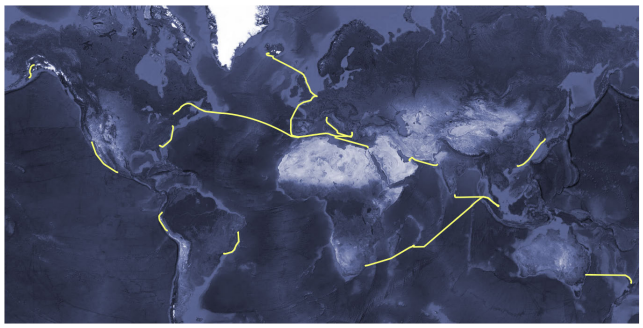

FIGURE 7. Indicative trajectories from the different Container trips used in our evaluation. These trips include voyages from and towards ports of all continents (except of Antarctica), with their duration ranging from a few hours to several days.

for both long and short-term route forecasting. First, we provide details regarding the dataset used and introduce the three evaluation scenarios. The baseline methods, as well as the evaluation metrics, are then described. Finally, we conclude this section by presenting and discussing the performance of the proposed method.

A. DATASETS

For the purpose of our experiments, we use AIS data from two types of vessels from around the globe, provided by MarineTraffic. In order to study the behavior of our approach in different types of movement, the trips selected include voyages with little maneuvering required (e.g. Pireaus-Heraclion) to highly complex trajectories (e.g. Pireaus-Santorini). Additionally, we included trips whose duration range from a few hours to days at a time. Regions with both low and high traffic were considered while selecting the trips, in order to highlight the effectiveness of the proposed approach in a variety of circumstances. Focusing on both Passenger and Container vessels, a separate dataset was extracted for each type. The first dataset consisted of 9 passenger vessels trips across the Aegean sea for 1 year (entire 2019). Due to the Aegean sea's many islands, the travelling vessels need to follow complex routes to reach their destination. The second dataset contained four-years of data (2016-2019) for 16 container trips from and towards some of the busiest ports around the globe. These trips allowed us to study the vessel behavior in different parts of the world, as seen in Figure 7, thus resulting in a more complete evaluation of our method.

Table 1 provides a more detailed overview of the 25 trips that we are examining in this study; describing the number of vessels, the number of distinct trajectories and the number of AIS messages of the different trips. During training and evaluation of our models, the trajectories of each route were split according to their time of occurrence. After sorting them based on the timestamp of their last AIS message, the first 70 % of the trajectories (full trips between the ports) were used for training while the rest were considered for testing.

TABLE 1. The number of vessels, trajectories (*i.e.* full paths) and messages, for each trip of the datasets used for evaluation. An abbreviation of each port's country name is also included.

origin		destination		vessels	traj.	messages
Passenger vessels						
Piraeus	GR	Santorini	GR	37	158	35143
Rafina	GR	Marmari	GR	2	1031	27288
Chios	GR	Mytilini	GR	4	379	23971
Sifnos	GR	Kimolos	GR	3	139	3279
Piraeus	GR	Milos	GR	7	110	13011
Piraeus	GR	Kythnos	GR	1	88	7087
Piraeus	GR	Heraclio	GR	20	627	138945
Piraeus	GR	Paros	GR	10	307	35552
Syros	GR	Patmos	GR	4	134	13115
Container vessels						
Long Beach	US	Manzanillo	MX	52	209	208521
New York	US	Savannah	US	252	1170	761196
Santos	BR	Salvador	BR	57	191	143916
Tanger Med	MA	Rotterdam Maasvlakte	NL	58	190	286993
Piraeus	GR	La Spezia	IT	80	229	253324
Colombo	LK	Tanjung Pelepas	MY	91	229	113955
Busan	KR	Hong Kong	CN-HK	111	692	270899
Montreal	CA	Algeciras	ES	17	101	210321
Durban	ZA	Singapore	SG	62	136	279176
Callao	PE	Guayaquil	EC	124	613	302379
Anchorage	US	Kodiak	US	4	308	53989
Rotterdam Waalhaven	NL	Reykjavik	IS	20	137	122385
Algeciras	ES	Piraeus	GR	32	65	79719
Tauranga	NZ	Botany	AU	34	298	314292
Marsaxlokk	MT	Port Said	EG	218	531	323679
Jebel Ali	AE	Kandla	IN	19	72	38436
Total (25 trips)				**1312**	**8144**	**4,060,571**

B. EVALUATION SCENARIOS

In order to provide a more detailed analysis of our method we propose three evaluation route forecasting scenarios:

1) *Entire route forecasting*: evaluating the forecasted trip regarding the full route, *i.e.* from departure port to the reported destination port.
2) *Underway long-term forecasting*: evaluating the forecasted path to the reported destination port from query points along the actual trajectory. For the purposes of these experiments we split each validation trajectory in three parts and performed multiple queries at each part.
3) *Underway short-term forecasting*: evaluating the short-term path forecasted by our method, compared to a state-of-the-art technique. The focus here is placed on the accuracy of the first part of the forecasted path, performing multiple queries along each trip.

C. BASELINE METHODS

For evaluating our method, we consider a set of state-of-the-art works for route forecasting in the maritime domain. These selected methods are based on different algorithmic categories: from a static set of predetermined paths, a classic trajectory analysis method based on historical data clustering and an advanced encoder-decoder mechanism that uses two types of recurrent neural networks in its core. Additionally, we consider the forecasts of our previously presented work,

resulting in a complete performance study of the proposed framework. More precisely we compare the effectiveness of *EnvClus** with:

- *EnvClus*: the previous version of our approach [12], which considered the shortest path from the most frequently visited cluster of the first envelope towards the destination (*sink*) node.
- *TraClus*: a widely-used trajectory clustering technique introduced by Lee et al. [23]. In this case, we evaluate *TraClus* algorithm using all trajectories that share the same origin and destination ports (*i.e.* same trip). Since multiple clusters could be detected for the same trip, we are reporting the performance of the trajectory cluster with the lowest distance from the actual trajectory.
- *Open Street Map (OSM)*: the open-source nautical charts indicating the detailed paths that connect two ports. The *OSM* routes are only available for the passenger trips and not for the container trips.
- *Capo*: a recently presented deep learning technique proposed by Capobianco et al. [10] for forecasting the vessel's movement for a short period of time.

D. EVALUATION METRICS

For each evaluation scenario we appropriately post-process the predicted paths and use the corresponding evaluation metric for comparing the forecasted route to the actual. For the first two scenarios, concerning the long-term forecasting of the vessels path, we employ a metric capable of comparing series of different lengths, since each method provides a forecast of non-constant size. For the short term forecasting scenario the baseline method returns forecasts for specific times in the future. Hence, in this scenario we are able to use a metric purposed for similarly constructed trajectories, as mentioned in [10].

Eval. Scenarios 1 & 2 For the first two scenarios we use the Dynamic Time Warping (DTW) [48], an algorithm used to align the two trajectories (the actual and the forecasted), allowing multiple matches to the same point. Then the distances between the matched points are computed in *km*, employing the haversine distance. The reported value is the average distance (in *km*) of all the matched points between the actual and the forecasted trajectory. The forecasts by *OSM* and *TraClus* are usually sparsely sampled, containing large gaps among two consecutive points. Thus, we interpolate both the actual and all forecasted trajectories (*EnvClus**, *EnvClus*, *TraClus* and *OSM*) maintaining the same distance (1*km*) among consecutive points.

Eval. Scenario 3 For the third scenario we evaluate the short-term performance of our technique following the same approach as [10]. More precisely, we compare the distances among the positions of the actual and the forecasted trajectories for the upcoming three hours, using three hours data as input. We then use the Mean Absolute Error (MAE) metric, measured in Nautical Miles, for our results. For each query the *Capo* implementation receives data from the past three hours as input, and forecasts the vessel's next 12 positions sampled at a rate of 15 minutes. Accordingly, we isolate the part of *EnvClus**'s forecasted trip considering the length of Capo's predicted path and resample it in 12 equal intervals.

E. PERFORMANCE RESULTS AND DISCUSSION

1) ENTIRE ROUTE FORECASTING

The performance of *EnvClus** and the competing techniques for forecasting the entire route are presented in Table 2. We also display the average distance for the different trips in *km* along with number of journeys that were used in order to evaluate our technique. Besides the average DTW distance, we present the percentage of improvement (i.e. %impr.) that refers to the reduction of DTW distance that our technique achieves. Greater percentage improvement means that the trajectory that is forecasted by *EnvClus** is closer to the actual trajectory, compared to the baseline method in question.

Overall, the proposed approach provides an improvement of 33.35% in full path forecasts (43.35% and 20.4%, against state-of-the-art methods and the previous version respectively). The improvement for passenger and container vessels was 33.34% and 33.39% respectively. More precisely, in the passenger dataset *EnvClus** outperforms the other techniques for the majority of the trips, avoiding large DTW distances. In more detail, the ability of *EnvClus** to make vessel-specific forecasts results to better predictions in comparison to *EnvClus*. The only exception occurs for the *Piraeus* → *Heraclio* trip, where the vessels tend to follow a single corridor and *EnvClus* achieves a slightly better performance. *TraClus* detects multiple clusters from the given set of trajectories, with each cluster modeling only a part of the entire route. There are some parts where vessels move, that are not modeled by any cluster. Thus, techniques that simply discover trajectories clusters are not suitable for accurate route forecasts, especially when vessels follow multiple pathways. Furthermore, the paths available at *OSM* in several cases are close to the actual paths that the vessels followed (e.g. *Piraeus* → *Santorini*), but for the majority of the trips *EnvClus** outperforms *OSM* significantly. The vessels' trajectories for *Piraeus* → *Santorini* are illustrated at the upper part of Figure 8. As we can see *EnvClus** captures the entire space where vessels moved; different colors are used for the resulting corridors, while the envelopes are depicted in yellow.

In the containers dataset, we are comparing *EnvClus** only with *EnvClus*, since the *TraClus* available implementation did not terminate in a reasonable amount of time and *OSM* paths are mainly available for passenger vessels. The trips of container vessels are much longer than those of the passenger vessels and the DTW distance errors are in general larger. This is happening since the corridors that container vessels follow, at the selected trips, are usually much wider; this can also be seen at the lower part of Figure 8 for the trip *New York* → *Savannah*. *EnvClus** in several cases (i.e. *Long Beach* → *Manzanillo*) is able to provide a remarkable improvement

TABLE 2. Performance results of entire route forecasting for EnvClus* (EC*), EnvClus (EC), TraClus (TC) and OpenSeaMaps (OSM), as calculated by the DTW (in *km*), along with the respective improvement of EnvClus*. For the Container vessels dataset we include the results of the EC approach as a baseline, since OSM routes are mainly available for Passenger vessels and TC could not terminate in a reasonable amount of time.

origin	destination	dist	traj.(#)	EC*	EC	impr(%)	TC	impr(%)	OSM	impr(%)
Passenger vessels										
Piraeus	Santorini	262	46	16.8	16.9	0.8	31.6	46.9	**16.4**	-2.1
Rafina	Marmari	25	308	**0.3**	0.5	33.6	0.6	47.6	1.3	74.9
Chios	Mytilini	96	112	**1.8**	1.9	3.3	2.8	35.1	1.9	5.9
Sifnos	Kimolos	23	40	**0.6**	0.6	4.4	3.6	84.2	0.6	2.9
Piraeus	Milos	144	31	**1.3**	1.3	0.5	1.5	14.4	5.0	74.5
Piraeus	Kythnos	86	25	**0.7**	0.8	9.9	2.5	71.4	1.9	63.3
Piraeus	Heraclio	310	187	1.8	1.8	-1.8	**1.7**	-4.3	6.5	72.5
Piraeus	Paros	163	73	**3.0**	3.2	5.3	7.2	58.1	3.9	22.0
Syros	Patmos	160	39	**1.0**	1.1	6.0	1.3	21.7	1.5	31.1
Container vessels										
Long Beach	Manzanillo	2266	26	**8.4**	67.8	87.6	–	–	–	–
New York	Savannah	1378	144	**17.1**	27.3	37.5	–	–	–	–
Santos	Salvador	1777	15	**15.6**	24.3	35.8	–	–	–	–
Tanger Med	Rotterdam M.	2604	20	3.0	**3.0**	-1.2	–	–	–	–
Piraeus	La Spezia	1826	25	**7.3**	7.5	2.8	–	–	–	–
Colombo	Tanj. Pelepas	2955	32	**7.5**	13.7	45.2	–	–	–	–
Busan	Hong Kong	2193	22	28.7	**28.1**	-1.9	–	–	–	–
Montreal	Algeciras	6129	6	**58.7**	110.3	46.8	–	–	–	–
Durban	Singapore	9124	10	**48.2**	547.9	91.2	–	–	–	–
Callao	Guayaquil	1316	71	**15.0**	17.3	13.2	–	–	–	–
Anchorage	Kodiak	489	21	**5.2**	9.0	42.5	–	–	–	–
Rotterdam W.	Reykjavik	2271	12	**28.3**	34.0	16.8	–	–	–	–
Algeciras	Piraeus	2833	15	**7.6**	7.9	3.7	–	–	–	–
Tauranga	Botany	2516	20	**5.0**	164.0	96.9	–	–	–	–
Marsaxlokk	Port Said	1719	19	**12.9**	14.7	11.9	–	–	–	–
Jebel Ali	Kandla	1761	8	**11.0**	22.0	50.1	–	–	–	–
Total (25 trips)		**1330 traj.**			**+20.40%**		**+35.58%**		**+51.12%**	

TABLE 3. Performance results of underway queries for both EnvClus* (EC*) and EnvClus (EC), as calculated by the DTW (in km).

origin	destination	EC* [A]	EC* [B]	EC* [C]	EC[A]	EC[B]	EC[C]
Passenger vessels							
Piraeus	Santorini	**17.1**	**4.8**	**1.6**	18.5	8.1	2.0
Rafina	Marmari	**0.5**	0.5	**0.4**	0.6	0.5	0.5
Chios	Mytilini	**1.1**	1.0	0.8	1.3	1.0	0.8
Sifnos	Kimolos	**0.5**	0.5	0.5	0.6	0.5	0.5
Piraeus	Milos	**1.1**	0.8	**0.6**	1.2	0.8	0.7
Piraeus	Kythnos	0.8	0.9	0.6	0.8	**0.8**	0.6
Piraeus	Heraclio	1.7	1.4	0.8	1.7	1.4	0.8
Piraeus	Paros	2.7	0.7	0.6	**2.5**	0.7	0.6
Syros	Patmos	1.1	0.7	0.8	1.1	**0.6**	**0.6**
Container vessels							
Long Beach	Manzanillo	**9.2**	**10.1**	9.2	60.2	18.5	**5.7**
New York	Savannah	**16.7**	13.2	6.9	23.1	**11.3**	**5.4**
Santos	Salvador	**15.7**	**9.8**	3.4	25.3	13.1	**2.2**
Tanger Med	Rotterdam M.	**3.1**	**2.9**	**1.6**	3.2	3.1	1.7
Piraeus	La Spezia	7.2	7.0	3.3	**7.0**	**6.2**	**2.3**
Colombo	Tanj. Pelepas	**8.2**	7.5	**2.0**	12.9	**6.9**	2.1
Busan	Hong Kong	**20.8**	**10.8**	**9.4**	22.3	12.5	12.6
Montreal	Algeciras	**54.1**	**26.7**	**15.1**	103.5	48.8	22.2
Durban	Singapore	**45.6**	**24.3**	**7.7**	208.0	143.1	12.7
Callao	Guayaquil	15.5	14.6	13.5	**14.3**	**6.8**	**4.0**
Anchorage	Kodiak	**6.2**	**8.7**	1.3	10.7	12.0	**1.2**
Rotterdam W.	Reykjavik	33.7	**34.6**	**8.7**	**31.7**	37.2	13.4
Algeciras	Piraeus	**7.7**	**6.3**	6.2	7.8	6.5	**5.6**
Tauranga	Botany	**4.7**	**4.1**	3.7	118.4	14.1	**2.1**
Marsaxlokk	Port Said	**11.9**	**8.1**	4.6	14.3	10.0	**3.8**
Jebel Ali	Kandla	**12.4**	**9.1**	**4.3**	25.9	16.1	4.7

in the provided forecasts compared to *EnvClus*, for the container dataset. For the trips where *EnvClus* outperforms *EnvClus**, the differences between the DTW distances of the two techniques are insignificant, and are mainly caused by the fact that the latter system centers its forecasts at the middle of these wide envelopes and not necessarily at the most dense location.

TABLE 4. Performance results for short-term route forecast. The results presented are the mean absolute error for the three forecasted hours, measured in nautical miles.

origin	destination	EnvClus*	Capo	impr(%)
Passenger vessels				
Piraeus	Santorini	**13.5**	41.3	67.3
Piraeus	Heraclio	**8.0**	21.1	62.1
Container vessels				
Long Beach	Manzanillo	**14.27**	24.60	42.0
New York	Savannah	11.53	**11.33**	-1.8
Santos	Salvador	8.53	**8.47**	-0.7
Tanger Med	Rotterdam Maasvlakte	**6.70**	10.07	33.5
Piraeus	La Spezia	8.70	**8.27**	-5.2
Colombo	Tanjung Pelepas	**8.60**	9.17	6.2
Busan	Hong Kong	**14.6**	15.8	7.6

2) UNDERWAY LONG-TERM PREDICTION

In Table 3 we present the performance of our technique making route forecast queries at different positions of the test trajectories. A key aspect of *EnvClus** is that it models the entire space where vessels move, differentiating us from *OSM*, that provides a single path between two ports. During the evaluation, we split each test trajectory into three equal parts (A, B and C accordingly) and generate 7 query points for each part.

For the majority of the queries *EnvClus** still results to smaller DTW distance in comparison to *EnvClus*. The results indicate that *EnvClus** 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 *Santos* → *Salvador*).

3) UNDERWAY SHORT-TERM PREDICTION

For the short-term forecasting experiments we only use trips that take at least 6 hours to complete, as our experiments need three hours as input and output respectively. Hence, for this scenario we have a selection of the evaluation trips (Table 4). Although our approach does not prevail in all trips, the ones where it underperforms do not show a difference for more than 5.2%. On the other hand, there are several occasions with remarkable improvement, compared to the *Capo* technique. Focusing on such trips one may attribute the performance difference to the trips' complexity, as in *Piraeus* → *Santorini* depicted in Figure 8 (a-b). While, for trips like *New York* → *Savannah*, where the vessels mainly tend to follow one main wide corridor, *Capo* slightly outperforms *EnvClus**. The reason behind this lies on the fact that the *EnvClus** route forecast considers the centers of the corridors as indicators, even if the vessel moves at the edge of the corridor. In turn, this points to the fact that regression techniques are not that capable for modeling multi-branched scenarios with significant differences between them, giving our approach the edge.

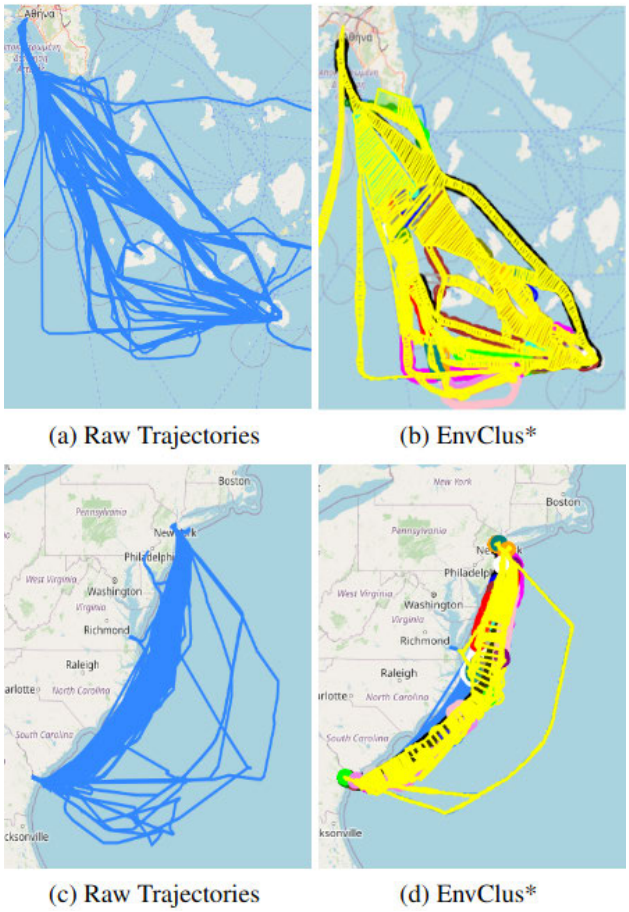

FIGURE 8. The raw trajectories (blue) and the corresponding envelopes (yellow) and corridors detected by *EnvClus for the trips: *Piraeus* → *Santorini* (top) and *New York* → *Savannah*.**

V. CONCLUSION

In this paper we present a complete framework for providing full-path trajectory forecasts for vessels. The proposed method uses patterns from historical data in order to create corridors of movement and allows for forecasts tailored to a vessel's characteristics, by the addition of effective classification models upon branch points. Along with presenting the extended methodology, we provide a comprehensive evaluation through three different scenarios. The results of this evaluation indicate a significant improvement over the current state of the art. For full path and long term forecasting, a notable improvement over the baseline methods is observed (33.35% overall), and especially for the Container vessels dataset. In terms of short-term forecasting the results

indicate that our approach can compete with state-of-the-art approaches, such as neural network architectures, with an improvement that may rise over 65% in some instances. The accuracy of our approach in highly complex trips highlights the importance of the classification models included in our method. Overall, based on these results along with the interpretability of our models, the proposed approach has proven to be a high performing solution, capable of handling different types of queries in a global scale.

There are several opportunities for exploiting the effectiveness of our system. Forecasting the representative trajectory from a given query location could be used in order to estimate the vessel's time of arrival or to detect anomalous movements. For instance, the case where a vessel moves away from the detected corridors of its route could refer to an anomaly and should be further investigated.

In the future, we intend to further improve the accuracy of our approach by superinducing additional features regarding the trip characteristics (*i.e.* weather data) and experimenting with other classification models. Finally, we intend to develop a scalable framework for effectively providing forecasts for multiple vessels at the same time, in a distributed way.

REFERENCES

[1] IT Union. *M.1371: Technical Characteristics for an Automatic Identification System Using Time-Division Multiple Access in the VHF Maritime Mobile Band*. Accessed: Sep. 20, 2021. [Online]. Available: https://www.itu.int/rec/R-REC-M.1371/en

[2] H. Georgiou, S. Karagiorgou, Y. Kontoulis, N. Pelekis, P. Petrou, D. Scarlatti, and Y. Theodoridis, "Moving objects analytics: Survey on future location & trajectory prediction methods," 2018, *arXiv:1807.04639*.

[3] Z. Xiao, X. Fu, L. Zhang, and R. S. M. Goh, "Traffic pattern mining and forecasting technologies in maritime traffic service networks: A comprehensive survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1796–1825, May 2020.

[4] J. Liu, X. Mao, Y. Fang, D. Zhu, and M. Q.-H. Meng, "A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2021, pp. 978–985.

[5] F. Leon and M. Gavrilescu, "A review of tracking and trajectory prediction methods for autonomous driving," *Mathematics*, vol. 9, no. 6, p. 660, Mar. 2021.

[6] J. Polevskis, M. Krastins, G. Korats, A. Skorodumovs, and J. Trokss, "Methods for processing and interpretation of AIS signals corrupted by noise and packet collisions," *Latvian J. Phys. Tech. Sci.*, vol. 49, no. 3, pp. 25–31, Jan. 2012.

[7] M. Yang, Y. Zou, and L. Fang, "Collision and detection performance with three overlap signal collisions in space-based AIS reception," in *Proc. IEEE 11th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Jun. 2012, pp. 1641–1648.

[8] E. d'Afflisio, P. Braca, and P. Willett, "Malicious AIS spoofing and abnormal stealth deviations: A comprehensive statistical framework for maritime anomaly detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 4, pp. 2093–2108, Aug. 2021.

[9] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G.-B. Huang, "Exploiting AIS data for intelligent maritime navigation: A comprehensive survey from data to methodology," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1559–1582, May 2018.

[10] S. Capobianco, L. M. Millefiori, N. Forti, P. Braca, and P. Willett, "Deep learning methods for vessel trajectory prediction based on recurrent neural networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 6, pp. 4329–4346, Dec. 2021.

[11] J. Chen, X. Li, Y. Xiao, H. Chen, and Y. Zhao, "FRA-LSTM: A vessel trajectory prediction method based on fusion of the forward and reverse sub-network," 2022, *arXiv:2201.07606*.

[12] N. Zygouras, G. Spiliopoulos, and D. Zissis, "Detecting representative trajectories from global AIS datasets," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 2278–2285.

[13] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–41, 2015.

[14] Z. Feng and Y. Zhu, "A survey on trajectory data mining: Techniques and applications," *IEEE Access*, vol. 4, pp. 2056–2067, 2016.

[15] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-temporal data mining: A survey of problems and methods," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–41, Jul. 2019.

[16] D. Wang, T. Miwa, and T. Morikawa, "Big trajectory data mining: A survey of methods, applications, and services," *Sensors*, vol. 20, no. 16, p. 4571, Aug. 2020.

[17] S. Wang, Z. Bao, J. S. Culpepper, and G. Cong, "A survey on trajectory data management, analytics, and learning," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–36, Mar. 2022.

[18] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, "A review of moving object trajectory clustering algorithms," *Artif. Intell. Rev.*, vol. 47, no. 1, pp. 123–144, Jan. 2017.

[19] J. Bian, D. Tian, Y. Tang, and D. Tao, "A survey on trajectory clustering analysis," 2018, *arXiv:1802.06971*.

[20] C. Leite da Silva, L. May Petry, and V. Bogorny, "A survey and comparison of trajectory classification methods," in *Proc. 8th Brazilian Conf. Intell. Syst. (BRACIS)*, Oct. 2019, pp. 788–793.

[21] K. Zheng, Y. Zheng, X. Xie, and X. Zhou, "Reducing uncertainty of low-sampling-rate trajectories," in *Proc. IEEE 28th Int. Conf. Data Eng.*, Apr. 2012, pp. 1144–1155.

[22] C. Chen, C. Lu, Q. Huang, Q. Yang, D. Gunopulos, and L. Guibas, "City-scale map creation and updating using GPS collections," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1465–1474.

[23] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, Jun. 2007, pp. 593–604.

[24] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, vol. 96, no. 34, pp. 226–231.

[25] J. Gudmundsson, A. Thom, and J. Vahrenhold, "Of motifs and goals: Mining trajectory data," in *Proc. 20th Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2012, pp. 129–138.

[26] L.-Y. Wei, Y. Zheng, and W.-C. Peng, "Constructing popular routes from uncertain trajectories," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 195–203.

[27] H. Zhu, J. Luo, H. Yin, X. Zhou, J. Z. Huang, and F. B. Zhan, "Mining trajectory corridors using Fréchet distance and meshing grids," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, Hyderabad, India, 2010, pp. 228–237.

[28] N. Zygouras and D. Gunopulos, "Corridor learning using individual trajectories," in *Proc. 19th IEEE Int. Conf. Mobile Data Manag. (MDM)*, Jun. 2018, pp. 155–160.

[29] P. Schmitt, M. L. Bartosiak, and T. Rydbergh, "Spatiotemporal data analytics for the maritime industry," in *Maritime Informatics*. Berlin, Germany: Springer, 2021, pp. 335–353.

[30] D. Zissis, K. Chatzikokolakis, G. Spiliopoulos, and M. Vodas, "A distributed spatial method for modeling maritime routes," *IEEE Access*, vol. 8, pp. 47556–47568, 2020, doi: 10.1109/ACCESS.2020.2979612.

[31] G. Wang, J. Meng, and Y. Han, "Extraction of maritime road networks from large-scale AIS data," *IEEE Access*, vol. 7, pp. 123035–123048, 2019.

[32] J.-S. Lee, H.-T. Lee, and I.-S. Cho, "Maritime traffic route detection framework based on statistical density analysis from AIS data using a clustering algorithm," *IEEE Access*, vol. 10, pp. 23355–23366, 2022.

[33] A. Troupiotis-Kapeliaris, G. Spiliopoulos, M. Vodas, and D. Zissis, "13 discovering shipping networks from raw vessel movement," in *Port Systems in Global Competition: Spatial-Economic Perspectives on the Co-Development of Seaports*. Oxfordshire, U.K.: Routledge, 2023, p. 265. [Online]. Available: https://www.routledge.com/Port-Systems-in-Global-Competition-Spatial-Economic-Perspectives-on-the/Ducruet-Notteboom/p/book/9781032327730#

[34] I. Kontopoulos, I. Varlamis, and K. Tserpes, "A distributed framework for extracting maritime traffic patterns," *Int. J. Geographical Inf. Sci.*, vol. 35, no. 4, pp. 767–792, Apr. 2021.

[35] B. Murray and L. P. Perera, "Ship behavior prediction via trajectory extraction-based clustering for maritime situation awareness," *J. Ocean Eng. Sci.*, vol. 7, no. 1, pp. 1–13, Feb. 2022.
[36] G. Tang, J. Lei, C. Shao, X. Hu, W. Cao, and S. Men, "Short-term prediction in vessel heave motion based on improved LSTM model," *IEEE Access*, vol. 9, pp. 58067–58078, 2021.
[37] G. Su, T. Liang, and M. Wang, "Prediction of vessel traffic volume in ports based on improved fuzzy neural network," *IEEE Access*, vol. 8, pp. 71199–71205, 2020.
[38] D. Ma, B. Sheng, S. Jin, X. Ma, and P. Gao, "Short-term traffic flow forecasting by selecting appropriate predictions based on pattern matching," *IEEE Access*, vol. 6, pp. 75629–75638, 2018.
[39] X. Zhou, Z. Liu, F. Wang, Y. Xie, and X. Zhang, "Using deep learning to forecast maritime vessel flows," *Sensors*, vol. 20, no. 6, p. 1761, Mar. 2020.
[40] S. Wang, J. Cao, and P. S. Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3681–3700, Aug. 2022.
[41] E. Chondrodima, P. Mandalis, N. Pelekis, and Y. Theodoridis, "Machine learning models for vessel route forecasting: An experimental comparison," in *Proc. 23rd IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2022, pp. 262–269.
[42] D.-D. Nguyen, C. Le Van, and M. I. Ali, "Vessel trajectory prediction using sequence-to-sequence models over spatial grid," in *Proc. 12th ACM Int. Conf. Distrib. Event-Based Syst.*, Jun. 2018, pp. 258–261.
[43] C.-H. Yang, C.-H. Wu, J.-C. Shao, Y.-C. Wang, and C.-M. Hsieh, "AIS-based intelligent vessel trajectory prediction using bi-LSTM," *IEEE Access*, vol. 10, pp. 24302–24315, 2022.
[44] L. You, S. Xiao, Q. Peng, C. Claramunt, X. Han, Z. Guan, and J. Zhang, "ST-Seq2Seq: A spatio-temporal feature-optimized Seq2Seq model for short-term vessel trajectory prediction," *IEEE Access*, vol. 8, pp. 218565–218574, 2020.
[45] P. Dierckx, *Curve and Surface Fitting With Splines*. Oxford, U.K.: Oxford Univ. Press, 1995.
[46] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Evanston, IL, USA: Routledge, 2017.
[47] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, Jul. 1992, pp. 144–152.
[48] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. KDD Workshop*, 1994, vol. 10, no. 16, pp. 359–370.

NIKOLAS ZYGOURAS received the B.Sc. degree in computer engineering and informatics from the University of Patras, the M.Sc. degree in artificial intelligence from The University of Edinburgh, and the Ph.D. degree from the National and Kapodistrian University of Athens. While with the National and Kapodistrian University of Athens, he was a member of the Knowledge Discovery in Databases Laboratory. His research interests include many aspects of mobility data mining and applications of machine learning for mobility data.

ALEXANDROS TROUPIOTIS-KAPELIARIS (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree with the Intelligent Transportation Systems Laboratory, Department of Product and Systems Design Engineering, University of the Aegean, Greece. From 2019 to 2022, he was an Associate Researcher with NCSR Demokritos. He is also a member with the Intelligent Transportation Systems Laboratory, Department of Product and Systems Design Engineering, University of the Aegean. He has participated in a few European projects, including INFORE, VesselAI, and CREXDATA. His research interests include machine learning, big data, and streaming systems.

DIMITRIS ZISSIS (Senior Member, IEEE) is currently an Associate Professor of information and communication systems with the Department of Product and Systems Design Engineering, University of the Aegean, Greece, the Head of the Intelligent Transportation Systems Laboratory, and the Director of the Postgraduate Program "Maritime Robotics and Informatics." His academic work has been published in more than 100 research articles, while the focus of his publications is on information and communications systems, big data, AI, and machine learning, which have received more than 4000 citations to date.