Geoinformatica (2021) 25:69–97
https://doi.org/10.1007/s10707-020-00421-y

Building navigation networks from multi-vessel trajectory data

Iraklis Varlamis[1] · Ioannis Kontopoulos[1] · Konstantinos Tserpes[1] · Mohammad Etemad[3] · Amilcar Soares[2] · Stan Matwin[3,4]

Received: 5 August 2019 / Revised: 1 June 2020 / Accepted: 16 July 2020 /
Published online: 7 August 2020

Abstract
Building a rich and informative model from raw data is a hard but valuable process with many applications. Ship routing and scheduling are two essential operations in the maritime industry that can save a lot of resources if they are optimally designed, but still, need a lot of information to be successful. Past and recent works in the field assume the availability of information such as the birth time-windows, cargo volumes, and container handling productivity at ports and cruising speed. They employ navigation maps that contain information about the major sailing paths and have knowledge about bigger or smaller ports and offshore platforms. In this work, we present a methodology for extracting information about the navigation network for an area, using data from the trajectories of multiple vessels, which are collected using the Automatic Identification System (AIS). We introduce a method for identifying the points of major interest to the trajectory of a vessel and two clustering techniques for identifying: i) key areas in the monitored region such as ports, platforms or areas where vessels change their course (e.g., capes); and ii) the speed and course patterns of ships of a particular type when they follow a typical route. The resulting information is modeled using a network abstraction where nodes correspond to the areas identified by the first clustering technique. After, edges are enriched with information about the groups extracted using the second clustering technique. The first analysis on a real dataset in the area of the eastern Mediterranean sea demonstrates the capabilities of the proposed model and the information it can provide. The use of the model in an outlier behavior detection task also shows interesting results.

Keywords Vessel trajectory mining · Trajectory analytics · Outlier detection

1 Introduction

The compulsory use of Automatic Identification System (AIS) enforced by naval regulations for many vessel types has created new opportunities for maritime surveillance. AIS

✉ Iraklis Varlamis
varlamis@hua.gr

Extended author information available on the last page of the article.

transponders are rich sources of information that can be collected using an RF receiver and can provide real-time information about vessels' position. If AIS data are correctly processed, it can provide useful information concerning typical vessel routes in the sea, vessels' normal behavior at each location and may uncover interesting anomalous vessel behavior, which can be connected to potentially illegal actions or other risky situations.

Anomalous vessel behavior can be indicative for a set of noteworthy events, such as a vessel in distress or vessel performing illegal activities. The impact of those events is severe and has a multifaceted effect on the environment, society, economy, etc. It is, therefore, crucial to employ technology to allow for the early detection of suchlike events. The opportunity is now more relevant than ever, with distributed data sensors tracking and reporting vessel movements around the globe [9, 33]. A mechanism that monitors vessel behavior and early detects anomalies, such as ships in distress, ships that rush to assist others in distress, or vessels that spoof their position for performing illegal activities would be useful for coast-guards and authorities.

Handling the volume of AIS data, which constitute a vast data stream, is the second challenge, which is a major challenge for traditional data analysis methods and machine learning algorithms [35, 40]. So, it is essential before any further analysis to simplify vessel trajectories [32] and if possible, to abstract the transactional model of AIS streams to a model that fits data mining and analytics.

The original motivation behind our work was to tackle the problem of anomaly detection in the sea using a model of normal behavior as a basis, which can be extracted from existing vessel trajectory data. The intuition was that a better solution requires the attribution of context-based knowledge to vessel trajectory data, such as i) the waypoints that define vessel operations and the sort of movement patterns that they follow in relation to those waypoints (i.e., a region of interest for a given application) over time, ii) the sub-trajectories that compose the trajectory of a vessel and the features that can be extracted for them. The main idea is to use AIS data from multiple vessels to identify the spatial waypoints according to frequently observed vessels' pattern, such as being stationary or making significant changes in their courses. Then, to understand the frequency and transition patterns of vessels moving from one waypoint to another using data from multiple vessels, and finally to generate a network that captures all this information. Given this network abstraction model, trajectory analysis can be performed to detect unexpected vessel behaviors.

In a previous work [42], we introduced a basic network abstraction of maritime traffic, that was based on nodes for representing areas where vessels stay idle for a long time or perform major turns and edges (called traversals) that connect two nodes at a time and correspond to ships moving between the respective nodes. Our analysis showed that nodes could be ports, anchorage areas, capes, offshore platforms, or even areas where vessels are changing their course (e.g., after exiting the port area). The work also highlighted some performance issues related to the number of points that the clustering algorithm had to process.

In this work, we extend the information extraction methodology: i) to distinguish between nodes that correspond to stop areas for vessels and nodes that correspond to areas of course change; and ii) to extract information about the network edges and the patterns of their traversal. To achieve this, we introduce a new clustering technique, which is based on the popular density-based clustering algorithm DBScan [11]. The proposed method modifies the proximity parameter (i.e., $epsilon$) of the algorithm and employs in tandem the difference in i) speed, ii) course and iii) position for defining the distance between two consecutive vessel positions (i.e., two consecutive AIS signals received from the same vessel). The results show that this combination performs significantly better than using only the

spatial distance and, more importantly, results in clusters that have very interesting properties. Besides, to improve the clustering algorithm performance, we pre-process data using a tile-based compression technique, which reduces the amount of data used in the clustering without affecting the clustering output. As a result, the nodes of the network abstraction are enriched with semantic information about their type and the connections with information about the different ways they are usually traversed.

The main contributions of this work are:

- A network abstraction model and its construction methodology, which can be the basis for outlying behavior detection using off-the-shelf methods.
- A new DBScan-based clustering technique, which considers two points (e.g., two vectors with position, direction and speed) to be in the same neighborhood when the vectors' positions are spatially close to each other, but they also have similar direction and speed.

The proposed approach is validated empirically and with running examples that demonstrate how the model can be used and interpreted.

This work is structured as follows: Section 2 summarizes the literature in the field of feature extraction from multiple trajectories and their use for trajectory comparison. Section 3 presents the proposed network abstraction model in detail, describes how trajectory data are compressed and how DBScan is extended to enrich the network abstraction model further. Section 4 illustrates some of the complex outlier detection methods that can be implemented over a network created from the AIS data of multiple vessels. Section 5 presents the results from the application of our methodology on a real dataset, evaluates the ability of the method to detect ports in the area, and demonstrates how node and edge information can be used for the detection of anomalies. Finally, Section 6 concludes the paper with the potential impact of this work in the domain of maritime surveillance by presenting future applications of the proposed network abstraction to the identification of more complex vessel behaviors that engage multiple vessels at the same time.

2 Related work

The proposed network abstraction model offers a method for extracting rich information from the trajectory data collected within a geographical area. As a simplification method, it compares to other methods in the literature that mainly focus on single trajectory simplification and proposes a multi-trajectory alternative. As a network abstraction model for traffic networks, it is comparable to methods that summarize multiple trajectories from historical AIS data, to generate traffic networks and establish the basis for a maritime surveillance system. Finally, as a methodology for anomaly detection it is comparable to techniques that use historical AIS data to detect abnormal or noteworthy patterns or events. Although the proposed methodology can be applied to the trajectories of several different types of moving objects, we limit our literature review to the maritime domain, which is directly related to the experimental work we performed so far.

2.1 Trajectory simplification, clustering and anomaly detection

Simplification algorithms are commonly used on AIS trajectories mainly to remove noise, temporal AIS transmission errors, etc. For example, the Douglas-Peucker (DP) line simplification algorithm [10] detects and removes redundant points from a single object trajectory,

when they fall within the expected object course (under a given threshold) [45]. However, it ignores the temporal dimension of a ship's route [46], as well as other contextual information (e.g., physical obstacles [39]), which when considered can significantly improve the quality of the simplified trajectory. On the other hand, the Open Window Spatiotemporal Algorithm (OPW-SP) [27] accounts for the speed changes and removes points that are within the ship course and within the expected time interval. Finally, the recently proposed Equivalent Passage Plan (EPP) Method [32] segments a vessel's trajectory into three basic behaviors: stop, fixed-course sailing, and turn. All the above methods have been applied in a single vessel trajectory at a time and do not consider historical information, e.g., previous trajectories of the same vessel at the same area, or trajectories from other vessels in the same area. Our work takes advantage of multiple trajectory information, either from the same or different vessels, and creates a general and abstracted navigation model of vessels in a navigation area.

Similarly to the Traffic Route Extraction and Anomaly Detection (TREAD) methodology suggested in [29], our work simplifies a set of trajectories from different vessels by extracting a set of waypoints. The TREAD method considers the spatial clusters of stationary, entry, and exit points from the area of interest as waypoints. It then builds route objects by clustering the extracted vessel flows, which connect two ports (stationary points), or any other pairwise combination of entry, exit, and stationary points. Our work expands the concept of waypoints, by including apart from the entry, exit, and stationary points, the clusters of *turning points*, where significant changes in the vessels' course frequently occur. Besides, we follow a different methodology for detecting waypoints and segmenting trajectories to sub-trajectories, which is further explained in Section 3 and introduce a new clustering methodology for trajectory points (as explained in Section 4), which allows identifying the various movement patterns of vessels across an edge of the network. As a result, the detection of anomalies is performed in a more context-rich, computationally cheaper and simplified way, taking advantage of the work in the area of network analysis.

Clustering algorithms play an important role in trajectory simplification and outlier detection, either they apply on individual vessel positions or on whole trajectories. A well-known approach for trajectory clustering is the *TraClus* [22] algorithm. *TraClus* partitions the trajectories into segments according to two rules, conciseness and preciseness. Conciseness means that the number of segments should be as small as possible. Preciseness means that the difference between the segments and the trajectory itself should be as small as possible. Later, the segments are grouped based on three types of distances, the positional difference between segments from different trajectories, the positional difference between segments from the same trajectory, and the directional difference of the segments. In a similar line, Tra-DBScan [24] clusters line segments using a Line Hausdorff Distance, which only examines the spatial features of trajectory segments. Furthermore, a similar modification of another popular clustering algorithm (k-Means) has been proposed in [23]. The authors modify the distance measure of the K-means algorithm to form moving micro-clusters. Their proposed distance combines position and velocity as well as time, thus assigning similar (in position and velocity) points from different periods to different clusters. Finally, authors in [28] extend the OPTICS clustering algorithm to remove the bias from the (time-)distance parameter of DB-Scan (i.e., *eps*) and cluster trajectories without being affected by temporal skewing. We suggest readers to consult the review work of [44] for a comprehensive coverage of distance/similarity metrics for trajectories and clustering algorithms that can be applied.

Although the aforementioned trajectory segment clustering algorithms also employ distance measures and apply variations of popular clustering algorithms, they differ from our

approach mainly for two reasons. First, because they ignore the vessels' speed, which is an essential factor that denotes the vessel's behavior and is considered in our algorithm. Secondly, because in our case, the trajectory segments are clustered separately, regardless of the trajectory they come from or the moving object that generated them. This allows our solution to generate meaningful cluster representatives that correspond to the way that multiple moving objects moved in a specific trajectory segment, as explained in Section 3.

In [30] authors present a single-pass processing approach ideal for streaming AIS data, which reduces noisy AIS positions, tracks moving vessels, and automatically detects specific event types (single or multi-vessel), such as rendezvous, package pickings, etc. The methodology is similar to the trajectory simplification step of our methodology. Still, it focuses on data streams and dynamic detection of predefined events, whereas the proposed frameworks perform a post-analysis of collected AIS data and create an abstraction, which can be the basis for further data analytics.

Our proposed solution is expected to perform better than related frameworks for anomaly detection from AIS data, which employ the position information of the consecutive vessel signals that constitute its trajectory and use Euclidean or other distance metrics in a two-dimensional space (i.e., latitude and longitude) [15, 16, 18, 34] or probabilistic approaches that partition space into tiles and estimate the probability of vessels to appear in a certain sequence of tiles [20] ignoring speed and direction. Even in approaches that use historical data to extract the average speed [13] or direction of move in a certain area [29], or techniques such as Piecewise Linear Segmentation (PLS) [21], speed and direction information is used only for predicting future vessel position, and the detection of deviation always measures the spatial distance of the actual from the predicted position. From our knowledge, this is the first approach that builds a composite model of speed, direction, and position for trajectories, which is then used to directly detect deviations of any of the three features or any combination of them. It is also expected to provide a richer model for the comparison of whole trajectories or sub-trajectories than the techniques that employ equal length sub-trajectories, or dynamic time warping and spatial distances to compare trajectories [14, 22] or techniques that combine spatial and temporal dimensions for indexing trajectories [38].

2.2 From vessel trajectories to traffic networks

Several graph/network models have been proposed in the past, in the transportation engineering domain, that try to capture and summarize mobility information and support data analytics. Works in road transportation developed navigable data models for supporting vehicle navigation starting from the road network, which is gradually transformed using information from the trajectories of moving objects [36]. The need to develop network abstractions for traffic simulation purposes is also evident in the transportation research [3, 47]. However, methods still rely on a predefined road or transportation network, which is enhanced with traffic pattern information. The current work focuses on the maritime domain, which does not rely on an underlying road or transportation network and allows (almost) any trajectory to occur. This adds more noise to the problem and makes trajectory partitioning necessary for providing fine-grained graph abstractions that approximate as much as possible the actual information.

Several works on maritime surveillance have used the grid of tiles or hexagons model [43] for mapping actual trajectories to polylines and consequently to sequences of key-points [19, 37]. The proposed simplification model is more coarse-grained than single trajectory simplifications that keep the majority of AIS data since it holds only a few points for each trajectory - the waypoints - along with a set of features for each sub-trajectory. As

it is shown in Section 3, the waypoints are distant from each other in contrast to the grid representation that uses neighboring tiles.

From the early works of Rhodes et al. back on 2005 [31] on maritime surveillance to the later works of Holst et al. [17] on maritime anomaly detection and the latest work of Varlamis et al. [41] on the detection of search and rescue missions from AIS data, several representation models have been proposed for describing trajectory information and many algorithms have been used to aid situation awareness, to detect adversarial tactics, previously unobserved events, and combinations of routine events concealing coordinated activities.

Several works have appeared in the last few years that build maritime traffic network representations from historical AIS data [4, 5, 8] or try to visualize large amounts of historical trajectory data [1, 2]. [1] tries to divide the area under surveillance into suitable areas or waypoints, which are then used to create and visualize flow maps. These flow maps indicate with arrows the direction of the trajectories (from waypoint to waypoint) as well as the number of trajectories per arrow (density). The authors later extend their work and present several visual analytics techniques that aim at understanding various aspects of movement in trajectory data [2]. In the two-layer network of [4]: i) the external layer presents the network's basic structure using waypoints as nodes/vertices and routes as edges/lines and ii) the internal layer is composed by nodes - *breakpoints* that reflect the vessels constant and stable changes of behavior and edges - *tracklets* that represent the vessel trajectory. The external layer is a coarse-grained abstraction of the traffic network, whereas the internal layer is a fine-grained version of the network that provides precision and granularity to the individual vessel layer. An edge in the external layer can be a route from a port to another port or an offshore platform. In contrast, an edge in the internal layer will comprise all the simplified (using DP algorithm) vessel trajectories that sailed across this route.

The complexity of the internal layer of the network and the scalability issues it creates is evident in the analysis that the authors performed in a real dataset for the Baltic Sea that comprised 1.8 million AIS points, from 1,136 actual routes [4]. According to the study, using only the 454 complete routes (from port to port) resulted in an internal layer composed of 2,095 tracklets. However, the aim of that work to reduce the RMSE between abstract routes and the actual courses and to monitor a rather small area (the area of Baltic Sea is only 377,000 km^2) explains its complexity. The level of abstraction of our model is similar to that of the external layer of [4]. However, we replace the over-detailed internal layer with statistical information extracted from the sub-trajectories of the various vessels to reduce the information stored by the model without losing its descriptive power. In addition to the above, in this work, we employ a tile-based compression technique to reduce the number of points that we keep for a trajectory, which mainly ignores multiple consecutive stationary points. More specifically, we divide the area into small square tiles (of side equal to 0.01 degrees), and when consecutive points of a trajectory fall in the same tile, we keep only the first point. This is very useful near ports or other stop areas, where vessels remain idle for longer periods and keep transmitting AIS messages.

To give an idea of the size of information that one must handle in a typical scenario, Fig. 1 shows a snapshot of more than 3,000 vessels that sail the Mediterranean sea on a typical day and the rectangle frames the area from Istanbul and Cyprus in the East to Genoa and Tunis in the West that we monitor. This is an area of 1.5 million km^2 for which 2.9 million AIS points have been collected in a month period from 1,716 cargo (only) vessels. This results in a bigger external network and a much more complex internal one than that of [4].

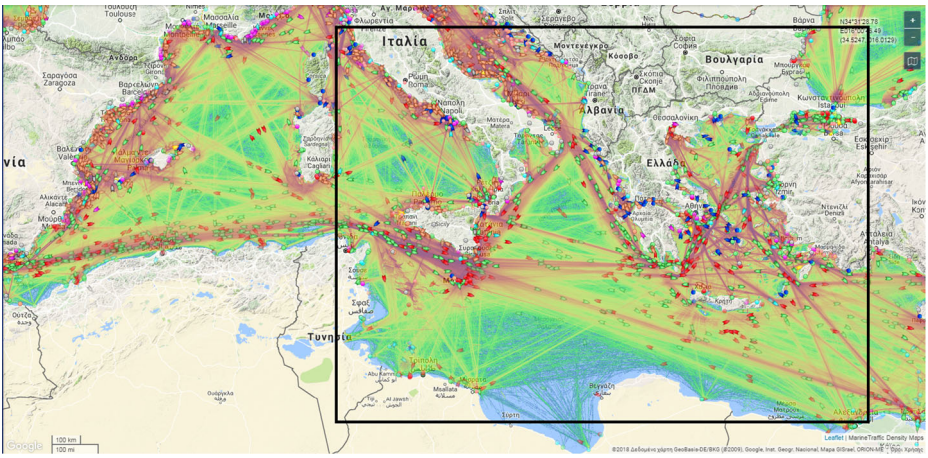


Fig. 1 A snapshot of the area monitored in this study

3 The proposed method

The proposed method is applied to trajectory data collected from multiple vessels of similar type (e.g., cargo vessels) for a period in a particular geographical area, but can be easily extended to cover larger areas and time-spans, or multiple types of vessel. Its only input is the AIS data reported by the vessels, which is processed and used to build a network abstraction of the collective vessel trajectory information.

The proposed method is summarized in Fig. 2. In step one, the trajectories (e.g., AIS messages) from multiple vessels are enriched with features that can be computed using geo-location and time (Section 3.1). After, trajectory points with particular characteristics (e.g., stops or points with high bearing rate) are clustered in waypoints that will be transformed in the nodes of our network (Section 3.3). The full network abstraction is processed in step 3 (Section 3.4), where trajectory segments' information that connects waypoints are used to create the edges of the model's network. Finally, the output of our method is a graph that represents a semantic network model that can be used for many different problems in the trajectory domain.

3.1 Trajectory data extraction

The first step of the processing workflow is the identification of the *keypoints* kp_{ij} in the trajectory T_i of a vessel. We consider as keypoints the points where the vessel stopped or moved slowly for a period of time or the points where the vessel quickly performed a major turn. The library TrajLib[1] was used to process the basic information collected from AIS (e.g., geo-location and time-stamp) for a vessel and extract information regarding the vessel speed, bearing, and bearing rate. This is done dynamically, as we collect geo-location and time-stamp information for a vessel. By applying the segmentation methods described in [13], we identify kp_{ij} as the segmentation points where the speed is below a threshold (i.e., very slow or stationary vessel) or the bearing rate is above a threshold (i.e., a major and quick change in the vessel's route). The speed threshold employed in the experiments of

[1] https://github.com/metemaad/TrajLib

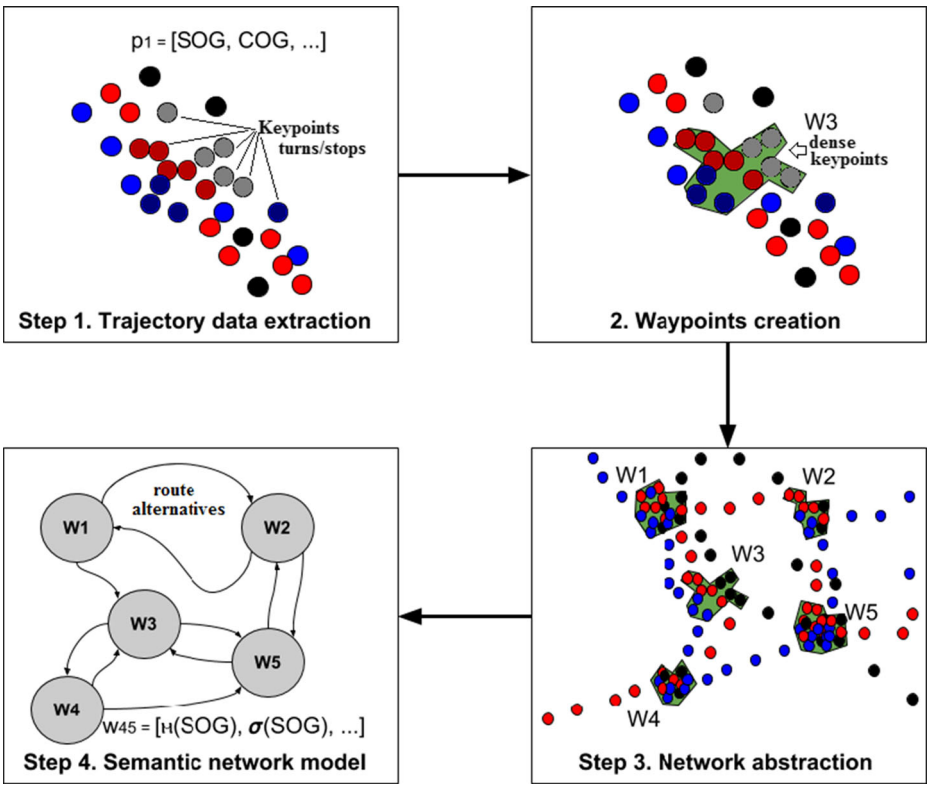


Fig. 2 The main steps of the proposed model

this work was 1 knot, whereas the threshold for the bearing rate was 0.1 degrees/minute. Thresholds have been decided empirically to capture very slow speeds or very quick turns. Different thresholds would change the number of keypoints extracted from each trajectory. Still, small changes are expected not to affect the definition of waypoints, which aggregate information from multiple vessel trajectories.

3.2 Redundant data cleaning

In a pre-processing step, the trajectory data are cleared from redundancies, which may occur when they remain stationary for a period (e.g., inside a port or at an offshore platform). To compress the data to speed up the process of the next step (Section 3.3), which refers to the identification of the waypoints, and to remove positions that do not alter the results of the following processes significantly, we use a tile-based technique. When vessels are stationary in a port, an offshore platform, or at an anchorage area, they keep transmitting AIS messages reporting zero speed, which results in many positions at the same spot that do not add any significant information. For that reason, we segment the surveillance area into very small tiles (0.01 degrees each side of the tile) of $1km^2$ each. Then, for each vessel, only the temporally first position received in each tile is kept, removing a huge amount of redundant positions per vessel. Due to the huge amount of stationary positions per vessel in ports or anchorage areas, keeping only the first position of each vessel results in a huge data

compression without any significant loss of information. Figure 3 illustrates the small tiles created in the port of Piraeus.

3.3 Waypoint identification

The second step refers to the spatial clustering of *keypoints* kp_{ij} collected from multiple vessels within a period. The DBScan [11] density-based algorithm is used to spatially group the keypoints to a set of arbitrary shaped clusters, that we call *waypoints* wp_k. Since the clusters produced by DBScan can have arbitrary shapes, we use closed polygons that envelops each cluster and merge overlapping convex hulls (see Fig. 4). DBScan parameters are also empirically chosen to support a comprehensive network abstraction. *Waypoints* are the nodes of our network abstraction model, and several features are associated with each one of them. The size of each cluster (i.e., number of keypoints it contains), the area it covers, its density, and the number of distinct vessels that contributed to it, are some of the features stored for each waypoint.

To add more semantics to the nodes of our model, we apply the clustering algorithm separately to the stop and the turn points keypoints, thus generating the stop and turn waypoints, respectively. Stop waypoints correspond to ports, anchorage areas, or offshore platforms, while turn waypoints correspond to points in space where vessels alter course.

Two examples of stop waypoints can be seen in Fig. 5. In Fig. 5a, a stop waypoint has been detected in the port of Napoli, Italy. More specifically, the geometry includes the entire port along with the areas where vessels wait or reduce speed, just before entering the port. Another representative example can be seen in Fig. 5b, where two clusters have been

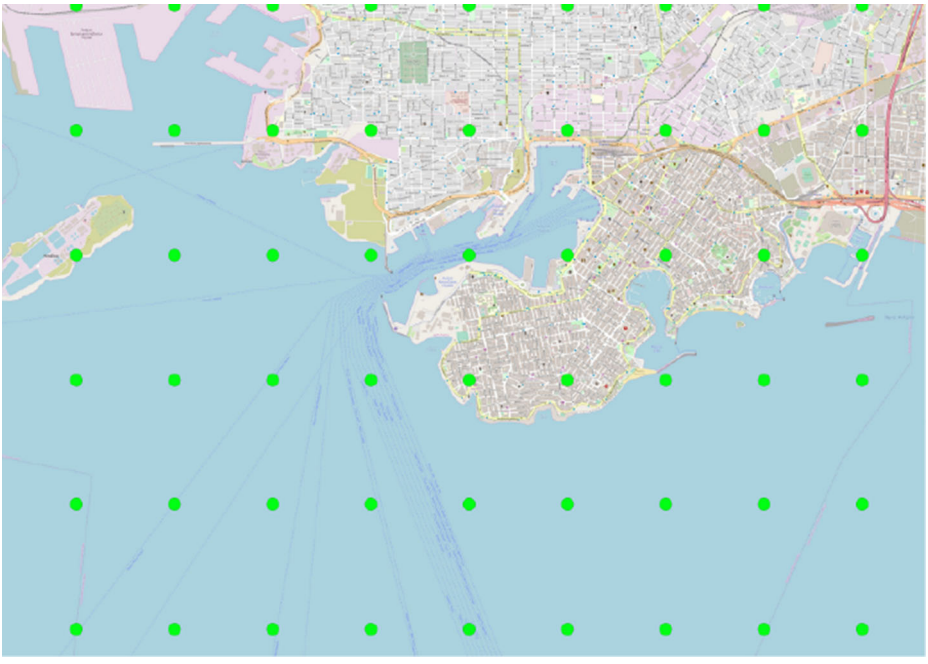


Fig. 3 An example of the small tiles in the port of Piraeus. Green dots denote the lower left corner of each tile

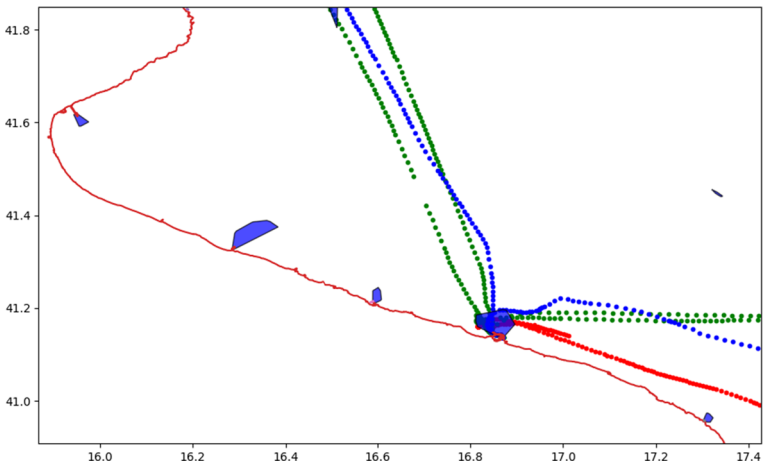


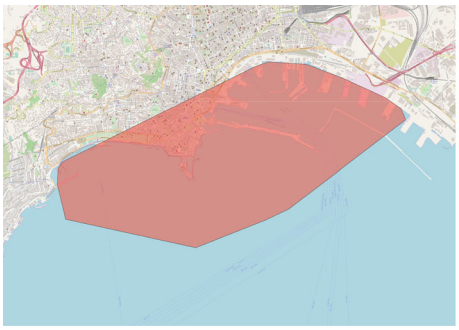
Fig. 4 The waypoints formed outside the port of Bari. The main waypoint corresponds to the port as indicated by three sample vessel routes that stop by

formed. The first cluster, near the shore, corresponds to the port of Thessaloniki, Greece, and the second cluster corresponds to the anchorage area near that port.

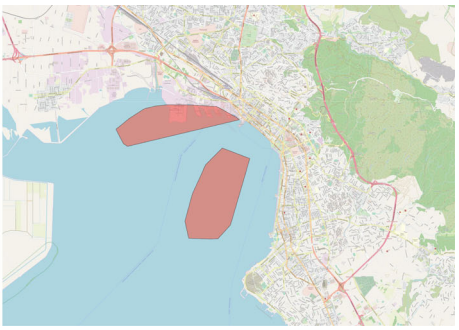
Turn waypoints are clusters that correspond to areas where many vessels change their course. The change in course for vessels traveling in the Aegean sea is due to the islands that exist in the vessels' route. An illustration of a turn waypoint can be seen in Fig. 6, where multiple clusters have been formed in the sea, close to cape Cavo D'Oro at South Evia. Turn waypoints can also be formed just outside the ports, because vessels just before their arrival or departure, reduce speed and change course to enter or exit the gates of the port.

3.4 Network abstraction

The next step is the creation of the edges, which, together with the nodes (i.e., waypoints), constitute the proposed network abstraction model. Since network edges correspond to vessel trajectories that move between two waypoints, in this step we collectively process



(a) An example of a waypoint that corresponds to the port of Napoli.



(b) An example of a waypoint that corresponds to the anchorage area near the port of Thessaloniki.

Fig. 5 Stop waypoints in ports

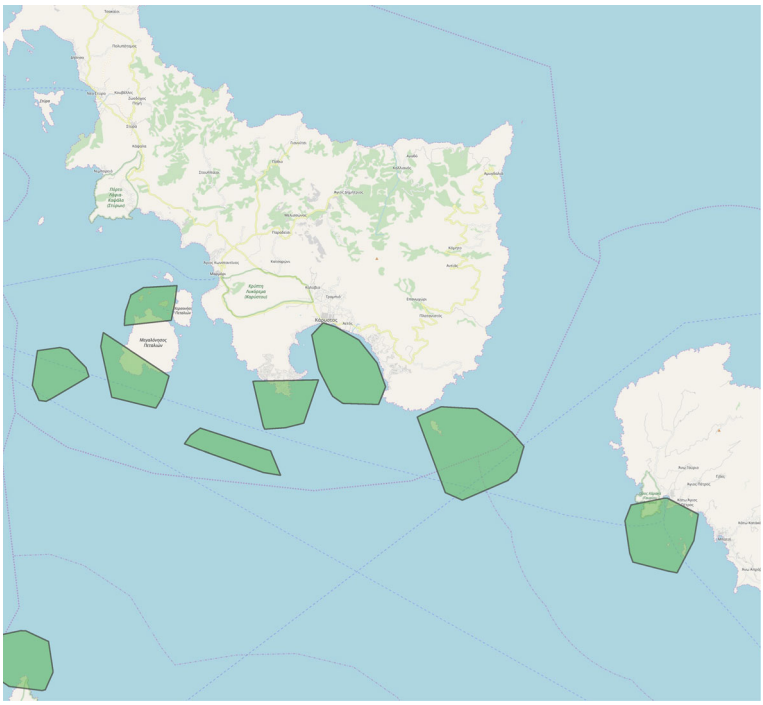


Fig. 6 An example of waypoints that correspond to route change areas, east of the Cape at the south Evoia, Cavo d'oro

the points from multiple sub-trajectories, using a clustering technique first, and a feature extraction technique for each cluster next. So, the first step is to use the waypoints for trajectory segmentation. For this purpose, we extended the TrajLib library [13], with a new trajectory segmentation method, which segments a trajectory to sub-trajectories that either connect two waypoints (the "between" edges) or traverse a waypoint (the "within" edges) (see Fig. 7). Since every waypoint is as a closed polygon, the trajectory of a vessel from departure to the destination will be split to a sequence of sub-trajectories that correspond to a sequence of alternating "between" and "within" edges.

The clustering of all the points that belong to the sub-trajectories of an edge is performed by applying a clustering algorithm, a variation of DBScan that is explained below. The DBScan algorithm is a density-based spatial clustering method that takes two parameters: *epsilon*, which specifies how close two points must be to be considered neighbors, and *minPts*, that specifies the number of neighbors a point must have to be included in a cluster. Our modified DBScan version uses three parameters to specify the proximity of candidate vessel AIS signals (positions):

– s: absolute difference of the speed between two positions (speed-based)
– h: absolute difference of the course over ground between two positions (heading-based)
– eps: harvesine distance between two positions (spatial-based)

Therefore, each vessel position contains three types of information: i) the vessel speed at this position; ii) the vessel course over ground at this position; and iii) the latitude and

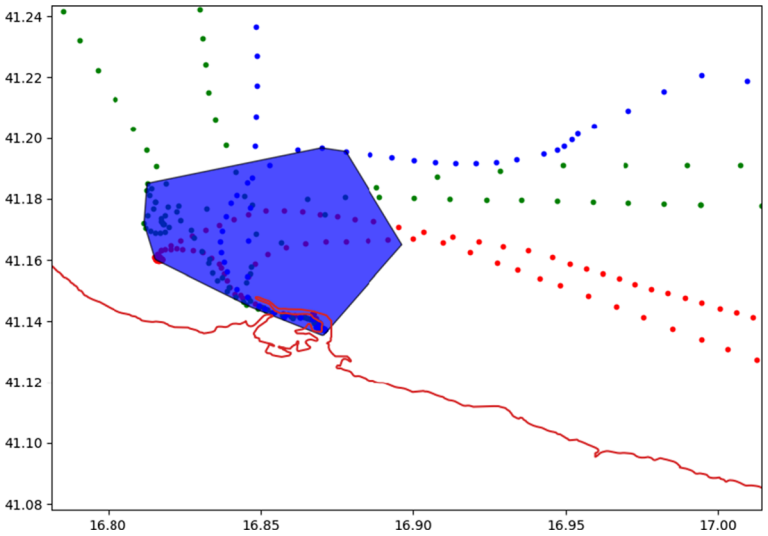


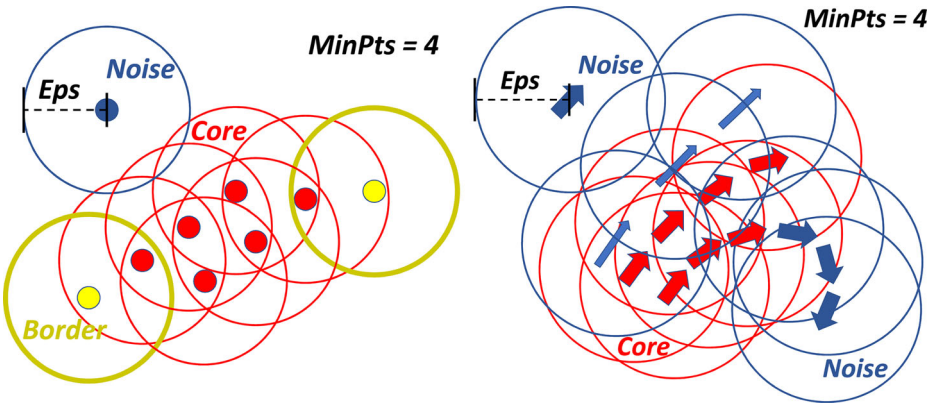
Fig. 7 A zoom of Fig. 4 reveals that parts of the trajectory correspond to movement within the waypoint limits

longitude of the position. Also, for a vessel position to be clustered together with another vessel position, the absolute difference in their speed must be below a threshold s, the absolute difference in their heading below a threshold h and the distance between them must be below a threshold eps at the same time. This type of clustering groups together trajectory points that have similar speed, heading and are close to each other.

The parameters of the algorithm can be fine-tuned by taking into account the frequent values of the dataset under certain conditions. For instance, in our case, a speed threshold s could be set to 3 knots, since the minimum speed value of vessels outside the port boundaries is approximately 3 knots. Regarding the h threshold, a difference of 3 degrees in heading empirically shows that it takes into account even the slow-turning vessels during a wide-angle turn. A distance threshold eps of 5 km will at least capture positions that originated from the same vessel while it may also capture positions transmitted from different vessels in more sparse areas.

An example of this type of clustering can be seen in Fig. 8, which compares the two implementations of the DBScan algorithm. Figure 8a shows the typical DBScan implementation, which creates a cluster if points are spatially close to each other. On the other hand, Fig. 8b illustrates the modified DBScan for the positions of moving objects, which considers two points (actually two vectors with the position, direction and speed) to be in the same neighborhood when the vectors' positions are spatially close to each other. Still, they also have a similar direction and speed. In the modified version blue arrows indicate noise vectors, which are either away, or have a different speed or have a different direction from all their neighboring vectors.

Clustering results are better if all the trajectory points inside the respective waypoints are excluded. Since waypoints are areas of interest through which vessels frequently pass, it can be easily inferred that the waypoints might be ports, platforms, canals or waterways. Inside these waypoints, vessels tend to alter their speed or heading frequently, which may corrupt the clustering results.



(a) Typical DBScan implementation. (b) Modified DBScan.

Fig. 8 Comparison of DBScan implementations

For the resulting clusters, we extract a list of features that are related to the distance covered, speed, acceleration, bearing, and bearing rate between every consecutive AIS signal collected for a vessel. So, instead of keeping all the intermediate GPS points and timestamps for a sub-trajectory, we maintain a vector that describes its mean, minimum, maximum, and intermediate percentile values of speed, distance, bearing, etc. as they have been calculated at each point. This significantly reduces the information stored for a cluster of sub-trajectories while keeping a lot of information concerning the vessel course and behavior.

The result of the clustering algorithm is a set of clusters, each one containing several positions from one or more vessels that constitute sub-trajectories of each vessel's route. Consequently, the trajectories of multiple vessels heading from one waypoint to another will be segmented, and segments will be clustered. The clusters of sub-trajectories can be combined to create route alternatives, as depicted in the last step of Fig. 2. Each cluster corresponds to an alternative edge connecting the same pair of nodes, in the network abstraction model. The edge is characterized by the extracted feature vectors, which describe how the vessels move across that edge. Multiple edges connecting the same nodes mean that there is more than one way to move between two waypoints. An example of such behavior is the route from one waypoint to another with an island in between. While many vessels may use a specific route to avoid the island, others may use an alternate one. This results in a route that splits in two and then gradually converges back to a single one. Finally, although a route may be segmented due to a turning point, it may be further segmented due to sudden changes of movement with respect to the neighboring positions.

3.5 The semantics of the network model

The resulting network abstraction can be represented using a network model such as the one depicted in Fig. 9. In this network model, the red-colored vertices correspond to waypoints, and the yellow vertices are used to represent the vessels. The convex hulls that were the edges that connected waypoints in the network abstraction are also vertices (blue color) of the network model. Convex hull vertices are connected with waypoints using directed edges that denote the direction of the moving vessels from one waypoint to the other. So a

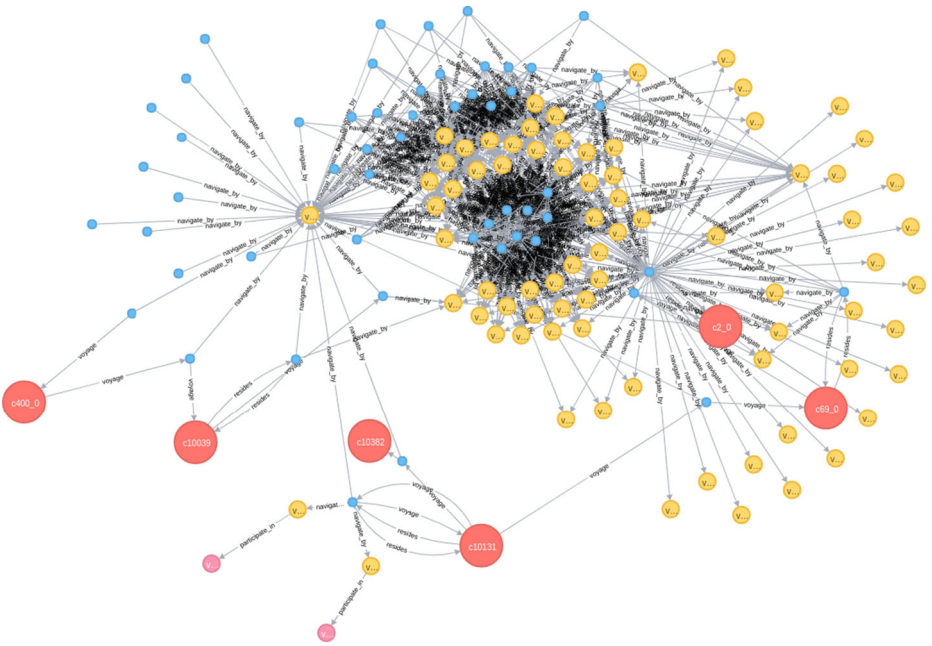


Fig. 9 An example of the semantic network model. The red markers are waypoints. The yellow markers are vessels. The edges of the original network abstraction are now mapped to vertices (blue nodes) which also connect to the vessels that traveled each original edge. The pink markers are outlier behaviors associated with a vessel (as in the displayed case) or a specific trip

convex hull from waypoint A to waypoint B (A→B) in the original network will be mapped to two edges in the semantic network that connect A and B through a connecting node N ($A \xrightarrow{voyage} N \xrightarrow{voyage} B$). Node N is marked with blue and is used to interconnect waypoints or pairs of waypoints with the vessels that traveled between the waypoint pair (directed 'voyage' edge) or stayed within a waypoint (directed 'resides' edge).

Any additional information that is extracted during the creation and enrichment of the network abstraction (e.g., average speed and standard deviation, direction, number of vessels that contributed to the convex hull, area covered by a waypoint), can be added as information to the vertices of the network model. It can also be represented using additional types of vertices, such as the pink-colored vertices depicted in the bottom of Fig. 9, which correspond to an outlier behavior.

4 Network analysis

4.1 Outlier detection methods

The problem of detecting outlier vessel behaviors usually aims in locating individual vessels that behave significantly different from all other vessels of the same type that operate in the same area [25]. The very recent work of Mao et al. [26] proposed a feature-grouping based outlier detection framework for distributed trajectory streams, which considers in a

tandem spatial proximity of trajectories and differences in multiple features such as speed, direction, etc.

The proposed network abstraction allows implementing both simple methods that detect spatial outliers (e.g., vessels that suddenly appear in an unexpected location) and more complex methods that use speed, direction, and their changes as features to detect more complex outlier behaviors.

4.2 Probabilistic graph traversal

The abstraction of an AIS dataset to a network that connects waypoints with traversal edges, allows us to describe the route of a vessel from the departure port to the destination port as a sequence of transition events between states (entering/exiting a waypoint) of the form:

(st_i, et_i, wp_x) or (st_j, et_j, wp_x, wp_y)

where st_i and et_i are the start, and end time of a "within" waypoint x traversal event (i.e., the time that the vessel entered and exited waypoint w_x), st_j and et_j are the start and end time of a transition from waypoint x to waypoint y (i.e., the time that the vessel exited waypoint w_x and the time it entered waypoint w_y respectively).

A straightforward use of this abstraction would be to learn the transition probabilities from one state to another using the route information of all vessels in an area for a time period. Training a Markov Chain model with this information will allow getting the probability of every future state given the previous states that a vessel attained in its route.

The detection of an outlier behavior during a route will be based on detecting a state transition of low probability. In simple words, this means that the vessel passed from several waypoints and then moved to waypoint that few or no other vessel with a similar route has been found before. In our analysis, we train discrete-time Markov chain models of order 1 and 2 using the first part of our timestamped dataset and evaluate the remaining data for transitions of low probability. This split assumes that training uses information for a specific time period, and then the model is used to detect outliers in the time period that follows.

By calculating the first-order (or higher) transition probability matrix using the historical data of all past waypoint sequences, we can detect anomaly sequences by simply looking at low probability values [7].

A requirement of such Markovian techniques (e.g., Finite State Automata, Hidden Markov Models, and Probabilistic Suffix Trees) is that input trajectories must have a limited length since they examine a fixed or maximum number of previous states [6] and cannot handle large sequences of trajectory points. In order to handle the varying trajectory length, we apply a sliding window of constant size over the past waypoint sequences, so that all the sequences used for transition probabilities' inference have the same length.

4.3 Outlier detection using subtrajectory features on edges

The network abstraction methodology presented in Section 3 for an AIS dataset that contains data from multiple vessels results in a graph with edges that have been traversed by more than one ship or more than one time. It is expected that the various vessel trajectories do not match precisely on GPS coordinates nor speed or direction features at every point. However, keeping the whole sub-trajectories and compare them point-by-point using RMSE or similar distance metrics to find outliers are both resources demanding and over-detailed. The proposed alternative approach is to use a feature vector for every sub-trajectory that contains distance, speed, bearing and bearing rate, and percentile values as features.

The set of features and the methodology employed to extract them from the timestamped GPS data is explained in detail in [12]. Since the AIS information is not continuous, the methodology assumes that a trajectory or sub-trajectory is a set of contiguous segments, for which it computes the following 'point' features: the duration, the distance covered, the acceleration, the jerk, the bearing rate and the rate of the bearing rate. Based on these 'point' features, the methodology computes global and local trajectory features, which are the minimum, maximum, mean, median, and standard deviation of the point features and different percentiles that describe the behavior within the trajectory. These features allow us to distinguish between a vessel that moved slowly and then speeds up to cover the distance and another vessel that had a smoother course, or between a long detour and a straight line sub-trajectory or between a vessel that made many maneuvers before reaching the final destination and a vessel that followed a simpler route.

The comparison of a set of trajectories or sub-trajectories that match in the start and end waypoint, with the aforementioned features, will reveal potential outlier behaviors, which can then be further examined. Outliers will be vectors that are far away from all other vectors either in a sub-space or the vector space of all features.

Both outlier detection methods described in this section are unsupervised since they do not require prior knowledge of normal or strange behaviors. The stochastic model used for outlier detection relies on the fact that a large AIS dataset for an area and a period, mostly contains normal routes that define the probabilities of normal and anomalous transitions. Using historical data to learn probabilities and new data to search for rare paths or transitions of low-probability may reveal potential outliers, such as ID (MMSI) spoofing or AIS switch-off. The vector-based representation of sub-trajectories and the use of centroid-based clustering algorithms are also unsupervised methods. It may reveal behavioral patterns, such as, for example, how different types of vessels move from one waypoint to the other, and outlier behaviors that do not match any existing feature vector. Using the same network abstraction with supervised methods is also possible, but it is harder to find training samples, so it is outside the scope of this work.

5 Preliminary results

The basis for building our graph model is a dataset containing 2.9 million AIS records that describe the trajectories of 1,716 distinct "cargo" vessels as they operated in the eastern half of the Mediterranean Sea during the period Aug. 01, 2015 to Aug 28, 2015. Since we did not have any additional knowledge about suspicious behaviors concerning this dataset, we decide to employ unsupervised/descriptive techniques to detect potential outliers. Each outlier has to be examined separately to understand the reason for being selected and reveal the specific characteristic of unusual behavior.

The first step of the preprocessing of the AIS dataset, requires the identification of key-points, which represent the major turn and stop points for the cargo vessels. Using a speed threshold of 1 knot and a bearing rate threshold of 0.1 degrees per minute, we located several thousand stops and turns ($\approx$ 500,000) in the trajectories of the monitored vessels. One interesting fact is that approximately 1 out of 6 positions is characterized as a stop or turn. As explained in Section 3.2, when the vessels are stopped, they keep transmitting AIS messages resulting in a huge amount of positions reporting zero speed in the majority of the trajectory lifetime. Furthermore, due to the selected area under surveillance, which contains many small islands (i.e., Aegean sea), vessels tend to make more maneuvers and frequent turns. When vessels are stopped, or at anchor, they make a slight movement (drifting) due

to the currents of the water; thus the AIS might report a minor speed ($0 \leq speed \leq 1$) that rarely exceeds the threshold of 1 knot. A bearing rate threshold of 0.1 degrees per minute detects even the slow turning vessels in wide-angled turns. The next step is the spatial clustering of the keypoints to waypoints. At this step, we used a minimum number of ten keypoints (MinPts=10) within a minimum radius of 2km (eps=2000) for distinguishing between core and noise points. A radius of $2km$ corresponds to an average-sized circle of the area small or large ports cover. For each $1km^2$ of area, there is only one position per vessel due to the compression step; thus a minimum number of 10 positions (10 vessels) yields the smallest possible cluster for turn points, even in areas with low traffic (sparse positions). The clustering algorithm resulted in 617 clusters, which are the nodes of our model.

At the second step of the preprocessing, we parsed the dataset a second time and segmented the trajectory history of each vessel as follows: i) first we split the trajectory into sub-tractories when the destination port changes assuming that a vessel changes its destination and begins a new trip when it arrives at the previous destination, ii) then we split each trip to sub-trajectories based on the points where it enters or exits a waypoint. The result of this preprocessing step is the distinct edge traversals in the proposed network model, which for the specific dataset are 53,391. These traversals correspond to 'between' and 'within' edges, some of them being traversed by more than one vessel. For each node traversal, we compute the distribution percentiles for all the features, as explained in Section 3.1.

Following the structure of the previous sections, we first provide some example results from the waypoint detection and the sub-trajectory clustering techniques (in Section 5.1) for illustrating the semantics of nodes and edges as described in Section 3. For illustrating the analysis described in Section 4, we provide example cases of vessels (in Section 5.2) that had an unusual behavior: i) in terms of the sequence of the waypoints they visited in their course and ii) in terms of the way they moved between two waypoints.

5.1 Network construction

5.1.1 Results of the waypoint detection technique

In order to evaluate the ability of the proposed methodology in detecting waypoints, we compare extracted clusters with polygons of real-world ports. The basis for creating the geometries of real-world ports was the World Port Index dataset, provided by the National Geospatial Intelligence Agency[2]. The dataset contains the location (longitude and latitude) of major ports and terminals worldwide. To distinguish between ports and offshore vessels, we filtered the AIS input data and kept only the positions where ships had zero speed and are located within $3km$ from the ports in the dataset. We applied DBScan to the remaining AIS positions to extract clusters, which are then converted into convex hulls. The resulting waypoints are fine-grained geometries representing, to a great extent, the real geometries of the ports. Figure 10 visualizes the geometry of the port of Napoli, Italy, after the creation of the convex hulls.

The waypoints and the original port geometries are then compared to evaluate whether waypoints actually represent ports. More specifically, we counted the number of waypoints that overlap with the geometries of the real-world ports. Out of the 239 ports located inside the surveillance area of this study, 217 were detected correctly (true positives) since they overlap with the detected waypoints, and only 22 were not (false negatives), resulting in a

[2] https://msi.nga.mil/NGAPortal/MSI.portal

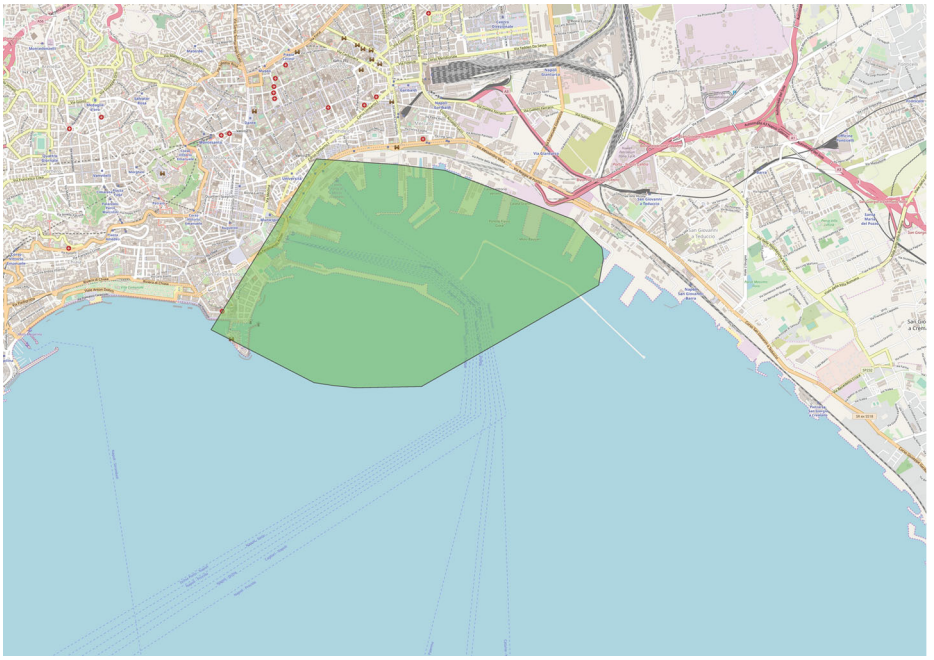


Fig. 10 Geometry of the port of Napoli, Italy

recall of 90.79%. In addition, 46 waypoints correspond to anchorage areas near the ports (false positives), resulting in a precision of 82.5% and f1-score of 86.44%. To evaluate the compactness of the resulting waypoints in comparison with that of the actual port shapes, we provide a distribution plot of the areas (in m^2) both for waypoints and ports in Fig. 11. The figure presents the distribution of shape sizes, using a solid yellow line for ports and a red dashed line for waypoints. The comparison shows that waypoints are smaller in size than the port shapes, which can be explained because waypoints can be areas of frequent stops

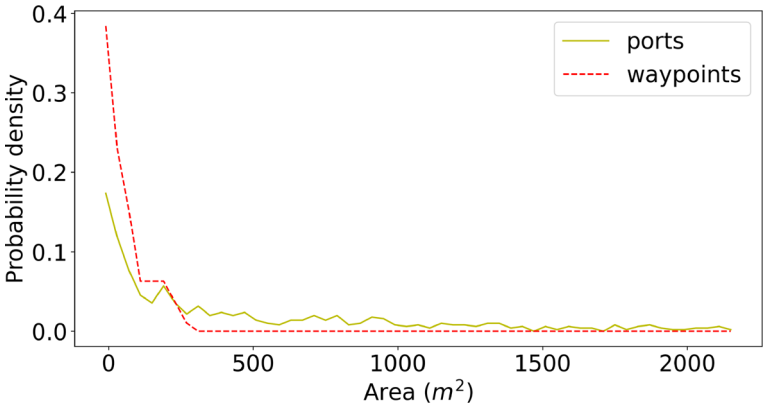


Fig. 11 Distribution plot of area of waypoints and ports

or turns, outside of the port areas or even in the open sea. This is also explained by the raw number of waypoints, which is almost double than that of ports (almost 300 in our dataset).

Finally, to get an idea of the density of AIS points in the detected waypoints, we depict in Fig. 12 the distribution of waypoint densities (number of AIS positions/m^2). Almost 1/3 of the waypoints contains 1 or less than one position per m^2, whereas another 1/3 contains more than 5 (and up to 30) AIS positions per m^2. The average density in the area we examine (1.5 million km^2) is 1.9×10^{-6} positions/m^2 (a total of 2,905,541 cargo positions have been collected during one month).

5.1.2 The output of the sub-trajectory clustering technique

The main idea behind the trajectory clustering using the modified DBScan presented in Section 3.4 is to extract clusters or convex hulls from vessel voyages which indicate the spatial and behavioral boundaries of the vessels' movement. Spatial boundaries represent the area that vessels can normally move. Behavioral boundaries represent the range of speed and heading the vessels can have inside the spatial boundaries.

A visual illustration of the results can be seen in Fig. 13, which demonstrates the pathways that cargo vessels follow when traveling from one port to another. Specifically, Fig. 13a illustrates the pathway between Catania, Sicily, and Malta, Fig. 13b visualizes the pathway between Istanbul, Turkey, and Piraeus, Greece, Fig. 13c demonstrates the pathway between Izmir, Turkey, and Thessaloniki, Greece and finally Fig. 13d illustrates the pathway between Napoli, Italy and Pireaus, Greece. Figure 13 shows that voyages are segmented into smaller parts, each part consisting of a polygon that contains all of the AIS positions with similar characteristics. Vessels traveling outside of these polygons for a certain amount of time or traveling inside these polygons with different speed and heading can be considered outliers.

A straightforward comparison of the proposed algorithm with other state-of-the-art algorithms is not possible. For example, the TraClus [22] algorithm groups trajectory segments into clusters and then generates one or more representative trajectories, which are composed of the representative segments of each cluster. Thus TraClus' edges are line-segments, whereas the edges of our algorithm (i.e. the convex hulls) are polygons. In order to allow a comparison of the two algorithms:

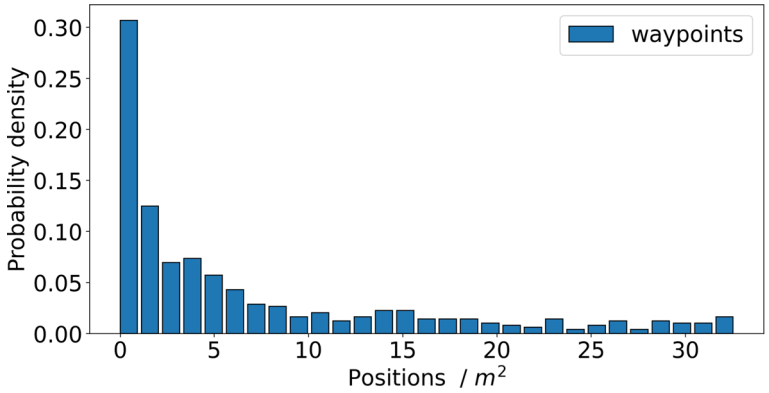
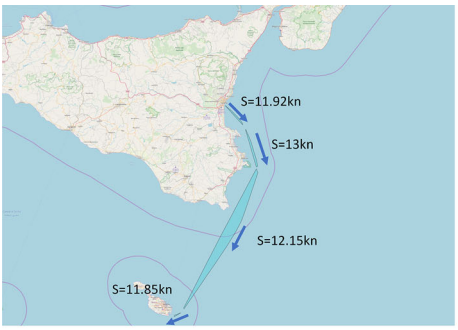
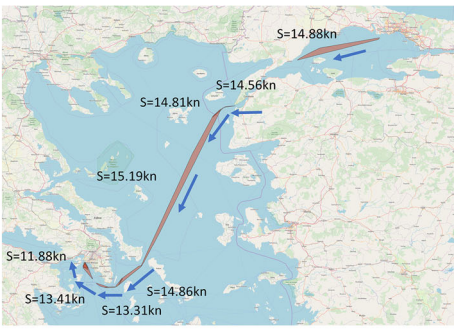


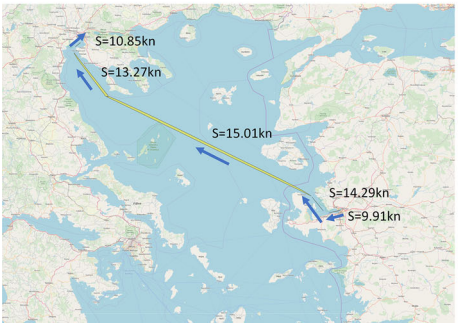
Fig. 12 Distribution plot of number of positions per square meter



(a) Common pathway between Catania, Sicely and Malta.



(b) Common pathway between Istanbul, Turkey and Piraeus, Greece.



(c) Common pathway between Izmir, Turkey and Thessaloniki, Greece.

(d) Common pathway between Napoli, Italy and Piraeus, Greece.

Fig. 13 Geometries representing common pathways

– we find the minimum (rotated) bounding rectangle (MBR) for each convex hull of our algorithm and consequently find the length and width of each edge (see Fig. 14). Then we plot the distribution of width/height ratio of edges, which depicts the compactness of our edges (see Fig. 15).
– since TraClus line segments do not have width, we apply the same width value in all edges (i.e., the average width of our convex hulls, which is 5.95 km) and plot the same distribution of width/height ratio as above in Fig. 15.

To evaluate the compactness of the convex hulls (edges of the network), we visualized the distribution of their width/height ratio in Fig. 15, with the ratio indicating how long the convex hulls are. Ratio value close to 0 indicates that the convex hulls are elongated, and ratio close to 1 indicates that the convex hulls form a square or a circle. From Fig. 15, we can observe that most of the convex hulls are elongated, which shows that they indeed represent trajectories that have the same destination and are close to each other. A similar distribution holds for the representative line segments produced by TraClus. For TraClus, we used a publicly available python implementation[3] with the default parameters.

Finally, Fig. 16 shows the patterns of the vessels around one of the largest marine protected areas in the Mediterranean Sea, the National Marine Park of Alonissos and the

[3] https://github.com/apolcyn/traclus_impl

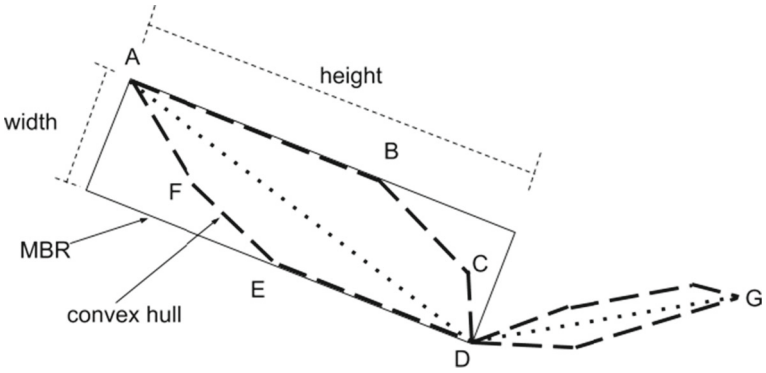


Fig. 14 An example of convex hull edges (e.g. <A,B,C,D,E,F>) and their Minimum rotated Bounding Rectangle, and TraClus line segments (e.g. <A,D>, <D,G>)

Northern Sporades. Polygons with different colors refer to different vessel voyages. Our technique revealed that vessels do not pass through the protected area and tend to go around the islands. The visualizations demonstrate that our proposed method can extract information about vessel voyages even in the most challenging areas with high traffic such as the Aegean sea. Furthermore, patterns around protected areas can be identified, which makes easier the detection of illegal activities.

5.2 Network analysis for outlier detection

5.2.1 Outlier detection using transition probabilities:

For this type of analysis, we employ part of the output of the preprocessing step, and more specifically, only the ids of the waypoints that have been visited by the cargo vessels of the dataset. This means that we use the sequence of waypoints in all the consecutive 'between' edges of each vessel trip. This resulted in 5,782 distinct trips performed by the 1,716 vessels during the one month period.

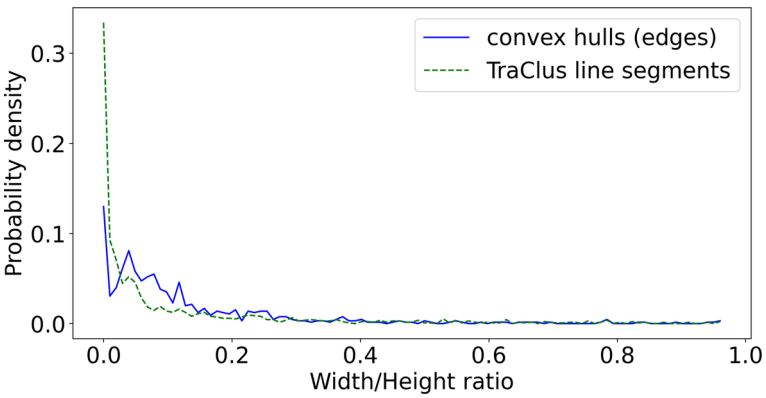


Fig. 15 Distribution of the width/height ratio of convex hulls and TraClus line segments

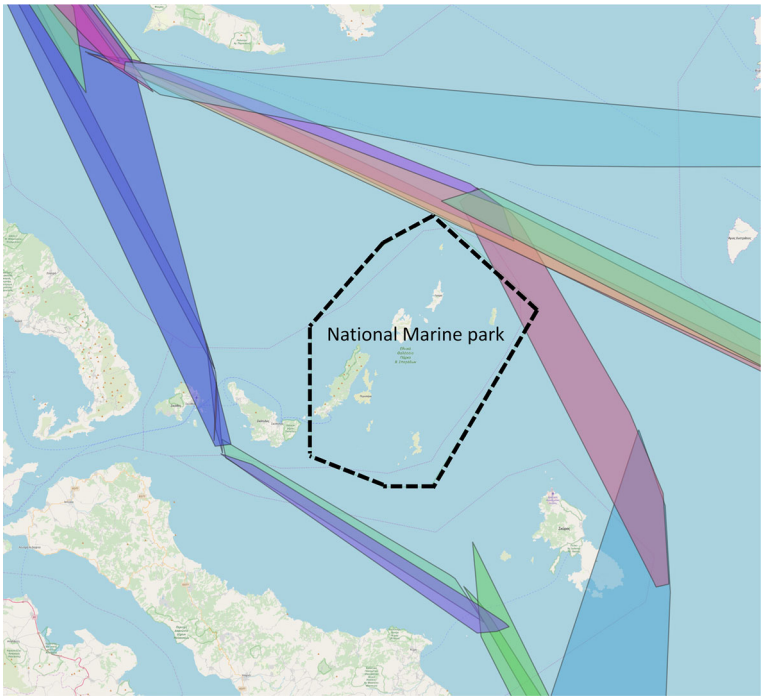


Fig. 16 Maritime traffic patterns in the National Marine Park of Alonissos and Northern Sporades

Our goal was to simulate a real scenario of training a surveillance model for a period and then using this model to detect potential outliers. So since the trips contain timestamps, we split the set of distinct trips sequentially in an 80-20% split using the least recent trips for training the transition probability matrix and the most recent to search for outliers. From the 1,156 trips that have been used as a test, only 10 have been found to have a low transition probability. Figure 17 shows an example of such a trip, which has been found as an outlier. The figure focuses on the problematic section of the trip, in the sea of Marmara, where it is evident that there is a considerable gap in the vessel trajectory, either because AIS information is missing or because the vessel is moving at a very high speed. Also, before that gap, we can see that the vessel does a strange maneuver, which must be further examined. A detailed examination of the trajectory features reveals that the vessel was moving fast before the gap but appeared with very slow moving speed after the gap and that it moved slowly during the maneuver (Fig. 18).

5.2.2 Outlier detection using edge traversal features:

A second approach in detecting outliers is to use the detailed information stored with the edges of our semantic network model. This information contains the distribution of values of all vessels that traveled across the edge and can be used to detect outlier behaviors that cannot be detected with the method described previously. These are the cases where a vessel moves across a frequently traversed path but has an anomalous behavior, for example, stops and starts, or moves slowly in some parts or during the whole path e.g., because of an engine problem.

Fig. 17 A zoom of the trajectory of a test vessel in the dataset, which has been detected as outlier

To detect such outliers, we perform a centroid-based clustering to the feature vectors of all vessels (trips) that traversed an edge. Based on the distance from the centroid and a percentile-based outlier detection method that keeps the majority of values around the means (95%) and drops off a small percentage (5%) that is either too low or too high we characterize some vessels as outliers. Any other distribution based technique could be used instead, such as dropping values that exceed a threshold around means that is related to the inter-quartile range. For a better view of the vessels' trips and in order to avoid short-term deviations, we repeat this process for more neighboring edges.

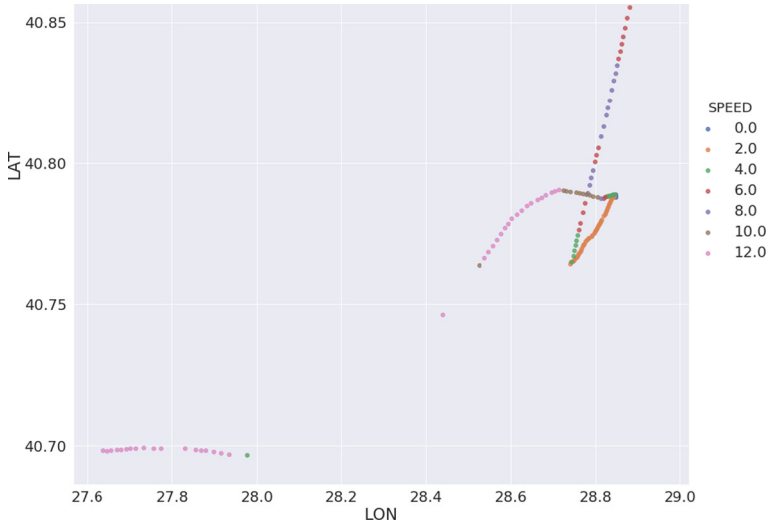


Fig. 18 The moving speed details of the trajectory detected as outlier

More specifically, we examine a very frequent sequence of edges in our dataset that relates to the route of vessels through the sea of Marmara, near Istanbul. There exist 359 vessel trips that traversed the same sequence of waypoints - of length 3, i.e., 2 edges - and among them, we locate 5 trips, for which the feature vector was in the top-5 percentile for both edges. One of the outliers was a high-speed vessel that moved at a speed of 20 knots, which is very unusual for cargo vessels in that area. A second outlier was a cargo vessel (shown in Fig. 19) that stopped for an extended period right after it left the port of Istanbul and then continued its trip.

6 Impact and future steps

A critical challenge for the detection of anomalous vessel behavior is to decipher the vessel operations by examining only AIS data, i.e., data that the vessels themselves regularly and openly transmit regarding their position at a particular time, their destination, and essential vessel characteristics such as their name and identity. Based on this data, more interesting information can be extracted to enhance a trajectory, such as the heading, speed, or bearing rate. Correlating the trajectory information collected from multiple vessels can be extremely beneficial to the task at hand. First, because the collective behavior of multiple-vessels may establish the behavioral norm in an unknown situation and second because there are several patterns of anomalous behavior at sea that engage more than one vessel.

The proposed network model is quite abstract to achieve good compression of vast amounts of data collected from thousands of vessels that operate in an area. At the same time, it is very comprehensive in the information it keeps for vessels' trajectories and allows

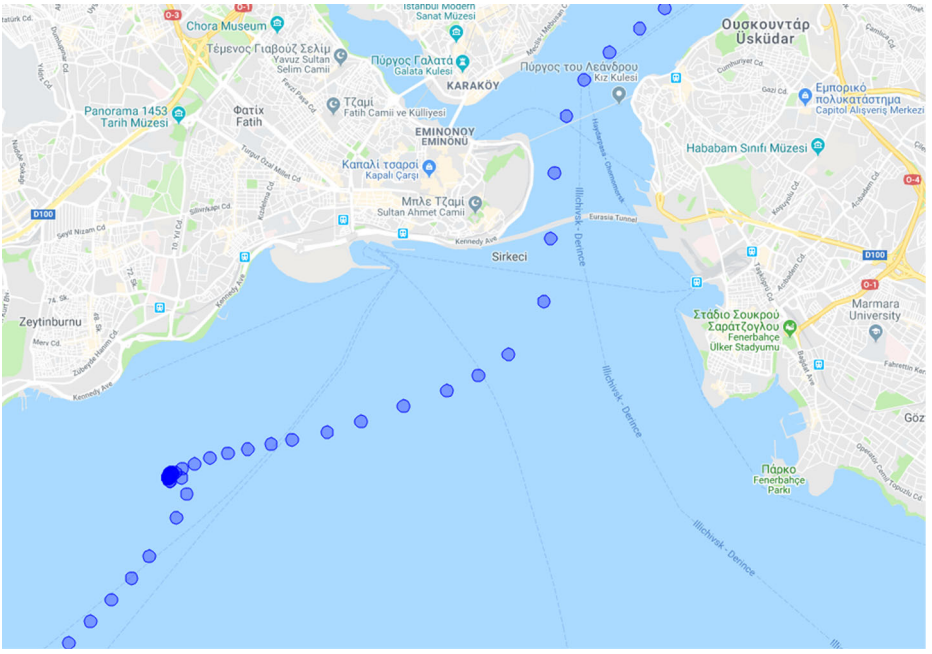


Fig. 19 A trajectory that has been found as outlier because of an unusual stop

more complex analysis to be performed, such as clustering or classification of movement patterns. The network abstraction of vessel trajectories for a region, can be used for processing new AIS data that come as a stream for this region, and quickly detect vessels that move from one waypoint to another or deviate from the predefined routes.

In this work, we presented the methodology for building the network abstraction and performed the first analysis using two unsupervised outlier detection techniques, which show two simple ways to exploit the network abstraction model. The next steps in this direction are: i) to identify the different types of anomalies that these two techniques can detect and ii) to compile a dataset of normal and anomalous behaviors and test the performance of our model in supervised setups. A more in-depth comparison between our method and TraClus by fine-tuning the methods' input parameters will be conducted for a more thorough comparison.

The main contribution relies on the network abstraction model and its construction methodology and not on the off-the-shelf outlier detection methods that we employed. Selecting specific types of anomalies to detect and having a human-reviewed dataset with cases of vessels that performed such anomalous behaviors in the area ([30], [41]), will allow us to exploit the proposed model, develop and evaluate new algorithms for the detection of related events.

Acknowledgments This work has been developed in the frame of the MASTER project, which has received funding from the European Union' s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 777695.

References

1. Andrienko N, Andrienko G (2011) Spatial generalization and aggregation of massive movement data. IEEE Trans Vis Comput Graph 17:205–19. https://doi.org/10.1109/TVCG.2010.44
2. Andrienko N, Andrienko G (2013) Visual analytics of movement: an overview of methods, tools, and procedures. Information Visualization https://doi.org/10.1177/1473871612457601
3. Andrienko N, Andrienko G, Rinzivillo S (2015) Exploiting spatial abstraction in predictive analytics of vehicle traffic. ISPRS Int J Geo-Inf 4(2):591–606
4. Arguedas VF, Pallotta G, Vespe M (2018) Maritime traffic networks: From historical positioning data to unsupervised maritime traffic monitoring. IEEE Trans ITS 19(3):722–732
5. Carlini E, de Lira VM, Soares A, Etemad M, Machado BB, Matwin S (2020) Uncovering vessel movement patterns from ais data with graph evolution analysis. In: Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference, vol 2578. CEUR Workshop Proceedings, Copenhagen. http://ceur-ws.org/Vol-2578/BMDA5.pdf
6. Chandola V (2009) Anomaly detection for symbolic sequences and time series data. PhD Thesis, University of Minnesota
7. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Computi Surv (CSUR) 41(3):15
8. Coscia P, Braca P, Millefiori LM, Palmieri FA, Willett P (2018) Multiple ornstein-uhlenbeck processes for maritime traffic graph representation. IEEE Transactions on Aerospace and Electronic Systems
9. Dividino R, Soares A, Matwin S, Isenor AW, Webb S, Brousseau M (2018) Semantic integration of real-time heterogeneous data streams for ocean-related decision making. In: Big Data and Artificial Intelligence for Military Decision Making, STO. https://doi.org/10.14339/STO-MP-IST-160-S1-3-PDF
10. Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: Int J Geograph Inf Geovis 10(2):112–122
11. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: SIGKDD'96. AAAI Press, pp 226–231. http://dl.acm.org/citation.cfm?id=3001460.3001507
12. Etemad M (2018) Transportation modes classification using feature engineering. PhD Thesis, Dalhousie University. CA arXiv preprint arXiv:180710876

13. Etemad M, Soares Júnior A, Matwin S (2018) Predicting transportation modes of gps trajectories using feature engineering and noise removal. In: 31st Canadian Conference on Artificial Intelligence. Springer, pp 259–264
14. Etemad M, Júnior AS, Hoseyni A, Rose J, Matwin S (2019) A trajectory segmentation algorithm based on interpolation-based change detection strategies. In: Proceedings of the Workshops of the EDBT/ICDT 2019 Joint Conference, EDBT/ICDT 2019, Lisbon. http://ceur-ws.org/Vol-2322/BMDA_4.pdf
15. Fu Z, Hu W, Tan T (2005) Similarity based vehicle trajectory clustering and anomaly detection. In: IEEE International Conference on Image Processing 2005, vol 2. IEEE, pp II–602
16. Hexeberg S, Flåten AL, Brekke EF et al (2017) Ais-based vessel trajectory prediction. In: 2017 20Th international conference on information fusion (Fusion). IEEE, pp 1–8
17. Holst A, Bjurling B, Ekman J, Rudström Å, Wallenius K, Björkman M, Fooladvandi F, Laxhammar R, Trönninger J (2012) A joint statistical and symbolic anomaly detection system: Increasing performance in maritime surveillance. In: 15th International Conf. on Information Fusion. IEEE, pp 1919–1926
18. Junior AS, Times VC, Renso C, Matwin S, Cabral LA (2018) A semi-supervised approach for the semantic segmentation of trajectories. In: 2018 19Th IEEE international conference on mobile data management (MDM). IEEE, pp 145–154
19. Kontopoulos I, Spiliopoulos G, Zissis D, Chatzikokolakis K, Artikis A (2018) Countering Real-time stream poisoning: An architecture for detecting vessel spoofing in streams of ais data. In: 4Th IEEE international conference on big data intelligence and computing (datacom 2018)
20. Laxhammar R, Falkman G, Sviestins E (2009) Anomaly detection in sea traffic - A comparison of the Gaussian Mixture Model and the Kernel Density Estimator. In: 2009 12th International Conference on Information Fusion, pp 756–763
21. Le Guillarme N, Lerouvreur X (2013) Unsupervised extraction of knowledge from s-ais data for maritime situational awareness. In: Proceedings of the 16th International Conference on Information Fusion. IEEE, pp 2025–2032
22. Lee JG, Han J, Whang KY (2007) Trajectory clustering: a partition-and-group framework. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of data. ACM, pp 593–604
23. Li Y, Han J, Yang J (2004) Clustering moving objects. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, KDD '04, New York, p 617–622. https://doi.org/10.1145/1014052.1014129
24. Liu LX, Song JT, Guan B, Wu ZX, He KJ (2012) Tra-dbscan: a algorithm of clustering trajectories. In: Applied mechanics and materials, vol 121
25. Mao J, Jin C, Zhang Z, Zhou A (2017) Anomaly detection for trajectory big data: Advancements and framework. Ruan Jian Xue Bao/J Softw 28(1):17–34
26. Mao J, Sun P, Jin C, Zhou A (2018) Outlier detection over distributed trajectory streams. In: Proceedings of the 2018 SIAM International Conference on Data Mining. SIAM, pp 64–72
27. Meratnia N, Rolf A (2004) Spatiotemporal compression techniques for moving point objects. In: International Conference on Extending Database Technology. Springer, pp 765–782
28. Nanni M, Pedreschi D (2006) Time-focused clustering of trajectories of moving objects. J Intell Inf Syst 27(3):267–289
29. Pallotta G, Vespe M, Bryan K (2013) Vessel pattern knowledge discovery from ais data: a framework for anomaly detection and route prediction. Entropy 15(6):2218–2245
30. Patroumpas K, Alevizos E, Artikis A, Vodas M, Pelekis N, Theodoridis Y (2017) Online event recognition from moving vessel trajectories. GeoInformatica 21(2):389–427
31. Rhodes BJ, Bomberger NA, Seibert M, Waxman AM (2005) Maritime situation monitoring and awareness using learning mechanisms. In: MILCOM 2005. IEEE, pp 646–652
32. Sánchez-Heres LF (2018) Simplification and event identification for ais trajectories: the equivalent passage plan method. J Navigat:1–14
33. Soares A, Dividino R, Abreu F, Brousseau M, Isenor AW, Webb S, Matwin S (2019) Crisis: Integrating ais and ocean data streams using semantic web standards for event detection. In: International Conference on Military Communications and Information Systems
34. Soares Júnior A, Moreno BN, Times VC, Matwin S, Cabral LdAF (2015) Grasp-uts: an algorithm for unsupervised trajectory segmentation. Int J Geogr Inf Sci 29(1):46–68
35. Soares Júnior A, Renso C, Matwin S (2017) Analytic: an active learning system for trajectory classification. IEEE Comput Graph Appl 37(5):28–39
36. Speičys L, Jensen CS (2008) Enabling location-based services—multi-graph representation of transportation networks. GeoInformatica 12(2):219–253
37. Stefanakis E (2016) mr-v: Line simplification through mnemonic rasterization. Geomatica 70(4):269–282

38. Tampakis P, Pelekis N, Andrienko N, Andrienko G, Fuchs G, Theodoridis Y (2018) Time-aware sub-trajectory clustering in hermes@ postgresql. In: 2018 IEEE 34Th international conference on data engineering (ICDE). IEEE, pp 1581–1584
39. Tienaah T, Stefanakis E, Coleman D (2015) Contextual douglas-peucker simplification. Geomatica 69(3):327–338
40. Valsamis A, Tserpes K, Zissis D, Anagnostopoulos D, Varvarigou T (2017) Employing traditional machine learning algorithms for big data streams analysis: The case of object trajectory prediction. J Syst Softw 127:249–257
41. Varlamis I, Tserpes K, Sardianos C (2018) Detecting search and rescue missions from ais data. In: 2018 IEEE 34Th international conference on data engineering workshops (ICDEW). IEEE, pp 60–65
42. Varlamis I, Tserpes K, Etemad M, Júnior AS, Matwin S (2019) A network abstraction of multi-vessel trajectory data for detecting anomalies. In: EDBT/ICDT Workshops
43. Yap P (2002) Grid-based path-finding. In: Conference of the Canadian Society for Computational Studies of Intelligence. Springer, pp 44–55
44. Yuan G, Sun P, Zhao J, Li D, Wang C (2017) A review of moving object trajectory clustering algorithms. Artif Intell Rev 47(1):123–144
45. Zhao L, Shi G (2018) A method for simplifying ship trajectory based on improved douglas–peucker algorithm. Ocean Eng 166:37–46
46. Zhao L, Shi G, Yang J (2018) Ship trajectories pre-processing based on ais data. J Navigat:1–21
47. Zhu L, Chiu YC, Chen Y (2017) Road network abstraction approach for traffic analysis: framework and numerical analysis. IET Intell Transp Syst 11(7):424–430

Iraklis Varlamis is an Associate Professor at the Department of Informatics and Telematics of the Harokopio University of Athens. He holds a PhD in Informatics from Athens University of Economics and Business, Greece and an MSc in Information Systems Engineering from UMIST, UK. He has been involved as a technical coordinator in a number of EU funded projects concerning knowledge management, data mining and Machine Learning. He has also coordinated several national R&D projects concerning data management and personalized delivery of information. He has authored more than 130 articles concerning text and graph mining and intelligent applications in social networks and the web and received more than 1900 citations. For more information visit: http://www.dit.hua.gr/∼varlamis.

Ioannis Kontopoulos received his BSc from the department of Informatics and Telematics of Harokopio University of Athens in 2016. He is a PhD candidate currently working at the same department as well as MarineTraffic. His major research interests revolve around Distributed Systems, Big Data analysis and processing, real-time stream processing, spatio-temporal and trajectory analysis. He has been involved in two H2020 projects, Datacron and Smartship.

Konstantinos Tserpes is an Assistant Professor at the Department of Informatics and Telematics of the Harokopio University of Athens. He holds a PhD in the area of Distributed Systems from the school of Electrical and Computer Engineering of the National Technical University of Athens (2008). His research interests revolve around distributed systems, software and service engineering and Big Data analysis. He has been involved in several EU and National funded projects conducting research for solving issues related to scalability, interoperability, fault tolerance, and extensibility in application domains such as multimedia, e-governance, post-production, finance, e-health and others. He has served as the scientific or general coordinator in several collaborative research projects such as +Spaces, SocIoS, Consensus, Fortissimo (FP7) and recently BASMATI (H2020). Currently, he is a partner in the Marie-Curie project: MASTER (H2020). Since 2015, he has been engaged in bilateral technology exchange collaboration with MarineTraffic for tackling issues related to trajectory analysis. For more information visit: http://www.dit.hua.gr/∼tserpes.

Mohammad Etemad is a Ph.D. candidate from the Dalhousie University, conducting research related to Spatial-Temporal Data mining with a focus on processing AIS Data. His research interests are Mobility Data Analysis, Data Mining and Spatial-Temporal data semantics. He has worked in the software Industry since 2007 and polished his software development skills in different ways. He is a certified Project Manager Professional (PMP) and an Agile practitioner (PMI-ACP). For more information visit: https://web.cs.dal.ca/∼etemad/.

Amilcar Soares is an Assistant Professor at the Memorial University of Newfoundland at the Department of Computer Science. His research interests include spatiotemporal data segmentation, classification, enrichment, and visualization. He holds a Ph.D. in computer science from Federal University of Pernambuco. He has been involved in several research projects funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), Department of Fisheries and Oceans (DFO), Transport Canada (TC), and Defence Research and Development Canada (DRDC). https://web.cs.dal.ca/∼soares/.

Stan Matwin is the Director of the Institute for Big Data Analytics at Dalhousie University, Halifax, Nova Scotia, Canada, and a Professor and Canada Research Chair (Tier 1) at the Faculty of Computer Science. He is also a Distinguished Professor Emeritus at the University of Ottawa, and a State Professor at the Institute for Computer Science of the Polish Academy of Sciences. He holds the titles of Fellow of the European Coordinating Committee on AI, and Fellow of the Canadian AI Association (CAIAC). He has received the 2019 CAIAC Lifetime Achievement Award. In 2017 he was elected to serve on the Board of Directors of CS-Can/Info Can, an organization serving as a focal point for Computer Science research and education in Canada. Internationally recognized for his work in text mining, applications of machine learning, and data privacy, he is a member of the Editorial Boards of leading journals in machine learning and data mining. He was the General Chair of KDD 2017 in Halifax, Canada. He has authored and co-authored more than 250 refereed papers and supervised more than 50 graduate students. He has extensive experience and interest in innovation and technology transfer. For more information visit: https://web.cs.dal.ca/~stan/.

Affiliations

Iraklis Varlamis[1] · Ioannis Kontopoulos[1] · Konstantinos Tserpes[1] · Mohammad Etemad[3] · Amilcar Soares[2] · Stan Matwin[3,4]

Ioannis Kontopoulos
kontopoulos@hua.gr

Konstantinos Tserpes
tserpes@hua.gr

Mohammad Etemad
etemad@dal.ca

Amilcar Soares
amilcarsj@mun.ca

Stan Matwin
stan@dal.ca

1 Department of Informatics, Telematics, Harokopio University of Athens, 9 Omirou Str., GR16122, Athens, Greece

2 Department of Computer Science, Memorial University of Newfoundland, S.J. Crew Building, EN-2021, St. John's, NL A1B 3X5, Canada

3 Institute for Big Data Analytics, Dalhousie University, Halifax, Canada

4 Polish Academy of Sciences, Warsaw, Poland