A network abstraction of multi-vessel trajectory data for detecting anomalies

Iraklis Varlamis
Department of Informatics & Telematics,
Harokopio University of Athens
Athens, Greece
varlamis@hua.gr

Konstantinos Tserpes
Department of Informatics & Telematics,
Harokopio University of Athens
Athens, Greece
tserpes@hua.gr

Mohammad Etemad
Institute for Big Data Analytics,
Dalhousie University
Halifax, Canada
etemad@dal.ca

Amílcar Soares Júnior
Institute for Big Data Analytics,
Dalhousie University
Halifax, Canada
amilcar.soares@dal.ca

Stan Matwin[1,2]
[1]Institute for Big Data Analytics,
Dalhousie University
Halifax, Canada
[2]Polish Academy of Sciences,
Warsaw, Poland
stan@dal.ca

ABSTRACT

The detection of anomalies in vessel trajectories is a problem of great interest for all maritime surveillance systems, since it may uncover strange, suspicious or difficult situations for vessels. All the existing works in the field examine specific aspects of the problem and propose case specific tools that can hardly generalize or scale-up to a worldwide monitoring system. In this article, we present a methodology for creating a network abstraction of the trajectories of multiple vessels, which uses only the information collected from the vessels' Automatic Identification System (AIS). The resulting network abstraction contains rich information about the vessel behavior in an area and can be processed with network analysis and other data mining techniques in order to uncover hidden outliers, even in an unsupervised manner. Experimental results on a real dataset demonstrate some of the capabilities of the proposed network model and indicate its extension to more complex automatic surveillance tasks.

KEYWORDS

vessel trajectory mining, trajectory analytics, outlier detection

1 INTRODUCTION

Abnormal 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 [5]. This work contributes directly towards that direction, with the provision of a mechanism that classifies vessel behavior between normal and abnormal, using historical information about similar vessels that operate in a particular area. The mechanism can be used for the early detection of vessels in distress or vessels that rush to assist others that are in distress or even for position spoofing in the case of illegal activities.

Although it would be easier to solve the anomaly detection problem using information from other data sources such as coast guard or vessel logs, the critical challenge is to decipher the vessel operations by examining only AIS data from multiple vessels in an area, i.e., data that the vessels themselves regularly and openly transmit regarding their position at a particular time. 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 [24, 27]. So, it is essential, before any further analysis, to simplify vessel trajectories [23] and if possible to abstract the transactional model of AIS streams to a model that fits data mining and analytics.

A large part of the literature on vessel position prediction and anomaly detection [15, 17, 18, 20, 22] focuses on the analysis of momentary GPS coordinates and vessel movement features (e.g., velocity, bearing) [12]. This point-based examination of the trajectories, however, implies the attribution of the same value in the analysis to each GPS point and this may result in weak results in understanding vessel behavior [2]. In reality, there are states in the vessels' trajectories, which are of high importance in the context of the real vessel operation that include the traversal of spatiotemporally defined waypoints (e.g. ports, off-shore platforms, capes, route deviations, vessel towing etc.) and a large number of states of low or no contribution to the solution of the problem at hand.

The intuition behind this work is that a better solution requires the attribution of context-based knowledge to vessel trajectory data, such as i) the waypoints that define their 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 subtrajectories 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 vesselsâĂŹ pattern, such as being stationary or making significant changes in their courses. After, 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.

This work is structured as follows: Section 2 summarizes related works. In Section 3 the proposed network abstraction model is presented in detail and Section 4 describes some of the complex outlier detection methods that can be implemented over a network created from the AIS data of multiple vessels. Section 5 discusses the preliminary results from the application of our anomaly detection methodology and explains how it can be extended to cover a broader range of anomalies and how it can be fine-tuned to capture specific trajectory 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 simplifying the information collected for a set of trajectories within a geographical area. As a simplification method, it compares other methods in the literature that mainly focus on single trajectory simplification and propose 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. 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

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 [6] detects and removes redundant points from a single object trajectory, when they fall within the expected object course (under a given threshold) [30]. However, it ignores the temporal dimension of a ship's route [31], as well as other contextual information (e.g., physical obstacles [26]), which when considered can significantly improve the quality of the simplified trajectory. On the other hand, the Open Window Spatiotemporal Algorithm (OPW-SP) [16] 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 [23] 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 [18], 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 and 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 vesselsâĂŹ course are frequently happening. Besides, we follow a different methodology for detecting waypoints and segmenting trajectories to sub-trajectories, which is further explained in Section 3. However, the main contribution of *our* work is the abstraction of the results of the aforementioned trajectories' analysis to a network model, in which 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.

In [19] 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, but it focuses on data streams and dynamic detection of predefined events, whereas the proposed frameworks performs a post-analysis of collected AIS data and forms an abstraction, which can be the basis for further analytics.

2.2 From vessel trajectories to traffic networks

Several works on maritime surveillance have used the grid of tiles or hexagons model [29] for mapping actual trajectories to polylines and consequently to sequences of key-points [11, 25]. 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 away from each other in contrast to the grid representation that uses neighboring tiles.

From the early works of Rhodes et al. back on 2005 [21] on maritime surveillance to the later works of Holst et al. [10] on maritime anomaly detection and the latest work of Varlamis et al. [28] 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 that last few years that builds maritime traffic network representations from historical AIS data [1, 4]. In the two-layer network of [1]: i) the external layer presents the networkâĂŹs 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 individual vessel layer. An edge in the external layer can be a route from a port to another port of an off-shore platform, whereas 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 of a real dataset for the Baltic Sea that comprised 1.8 million AIS points, from 1,136 actual routes. Using only the 454 complete routes (from port to port) resulted in an internal layer

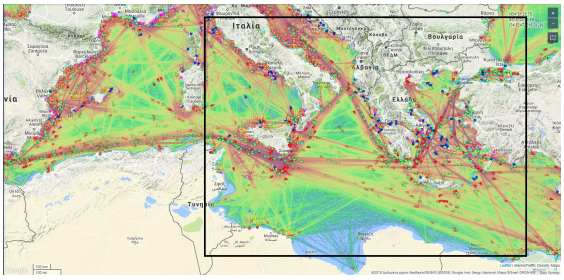

Figure 1: A snapshot of the area monitored in this study.

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 [1]. 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 loosing its descriptive power. To give an idea of the size of information that one must handle in a typical scenario, Figure 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 to a bigger external network and a much more complex internal one than that of [1].

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.

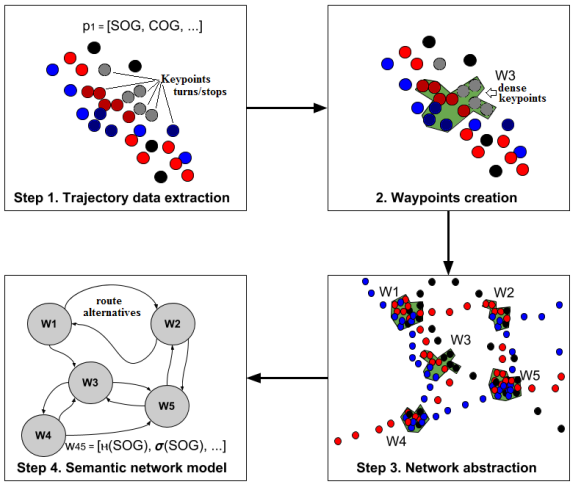

Figure 2: The main steps of the proposed model.

The proposed method is summarized in Figure 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 3.2). The full network abstraction is processed in step 3 (Section 3.3), 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 approach 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 [9], 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 this work was 1 knot, whereas the threshold for the bearing rate was 0.1 degrees/minute. Thresholds have been decided empirically in order to capture very slow speeds or very quick turns. Different thresholds would change the number of keypoints extracted from each trajectory, but small changes are expected not to affect the definition of waypoints, which aggregate information from multiple vessel trajectories.

3.2 Waypoint identification

The second step refers to the spatial clustering of *keypoints* kp_{ij} collected from multiple vessels within a period. The DBScan [7] 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 envelop each cluster and merge overlapping convex hulls (see Figure 3). 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.

3.3 Network abstraction

The next step is the creation of the edges that together with the nodes (i.e., waypoints) constitute the proposed network abstraction model. In order to define the network edges and extract their features, we once again process the AIS data this time using waypoints for trajectory segmentation. For this purpose, we extended the TrajLib library, with a new trajectory segmentation method, which segments a trajectory to subtrajectories that either connect

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

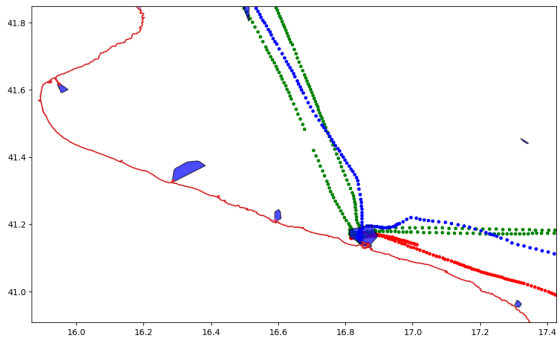

Figure 3: The waypoints formed outside the port of Bari. The main waypoint corresponds to the port as indicated by three sample vessel routes that stop by.

two waypoints (the "between" edges) or traverse a waypoint (the "within" edges) (see Figure 4). Since every waypoint is as a closed polygon, the trajectory of a vessel from departure to destination will be split to a sequence of subtrajectories that correspond to a sequence of alternating "between" and "within" edges.

For each subtrajectory, 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 subtrajectory, 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 subtrajectory, while keeping a lot of information concerning the vessel course and behavior.

A vessel's route from the departure to the destination port will be mapped to a path in the simple network abstraction depicted in the last step of Figure 2. Each route will add sub-trajectory feature vectors to one or more edges that will describe how the specific vessel sailed along the edges that form its path. A simplified representation of this network will be a directed and weighted graph with weighted vertices, where weights on the edges correspond to the number of vessels sailed along the edge and weights on the vertices will correspond to the number of vessels sailed through the waypoint.

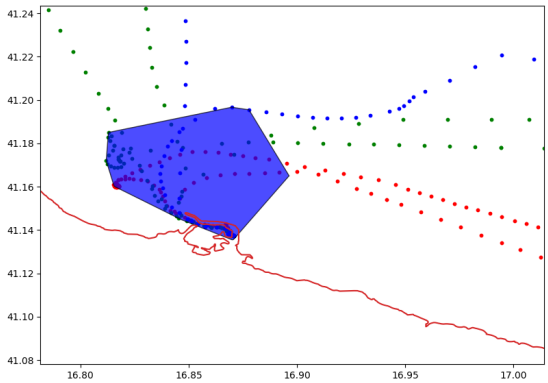

Figure 4: A zoom of figure 3 reveals that parts of the trajectory correspond to movement within the waypoint limits.

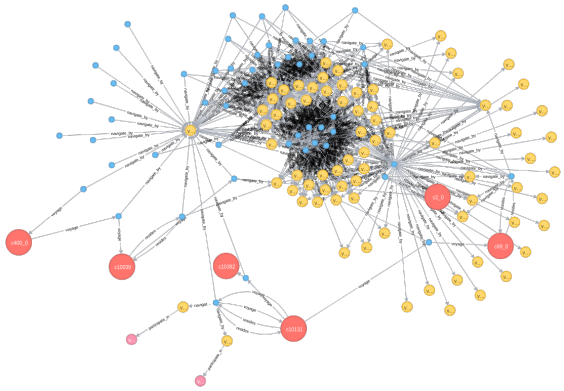

Figure 5: An example of the semantic network model. The red markers are waypoints. The yellow markers are vessels. The edges of the original network abstraction are now mapped to vertices (blue nodes) which also connect to the vessels that traveled each original edge. The pink markers are outlier behaviors associated with a vessel (as in the displayed case) or a specific trip.

3.4 A semantic network model

The resulting network abstraction can be enriched in order to better illustrate the information extracted in the previous steps. For example, since more than one vessels may navigate between two waypoints (i.e. navigate the same edge of the network) or stay within a waypoint (i.e. traverse a 'self-edge' of the network), we can use vertices of different types and directed edges that connect them as shown in Figure 5. In this semantic network model, the red colored vertices correspond to waypoints, and the yellow vertices are used to represent the vessels. The edges of the network abstraction are now converted to vertices (blue color) that lay between the waypoints and are connected with them through directed edges. So a directed edge 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 Îİ ($A \xrightarrow{voyage} N \xrightarrow{voyage} B$). 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 preprocessing for the creation of the abstract network or from the analysis of the feature information that it conveys can also be added to the semantic network model. This can be done with additional types of vertices such as the pink colored vertices depicted in the bottom of Figure 5, which correspond to an outlier behavior.

4 GRAPH ANALYSIS AND 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 [13]. The very recent work of Mao et al. [14] 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.

4.1 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:

$$(st_i, et_i, wp_x) \text{ or } (st_j, et_j, wp_x, wp_y)$$

where st_i and et_i are the start and 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 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 [3]. A caveat of this approach is that it must be used for sequences of the same length. In order to avoid this, we apply a sliding window of constant size over the past waypoint sequences, so that all the sequences have the same length.

4.2 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 ships or more than one times. It is expected that the various vessel trajectories do not match exactly on GPS coordinates nor speed or direction features at every point. However, keeping the whole subtrajectories and compare them point-by-point using RMSE or similar distance metrics in order to find outliers is both resources demanding and over-detailed. The proposed alternative approach is to use a feature vector for every subtrajectory 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 are explained in details in [8]. 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 speed 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 abnormal 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 type 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 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 keypoints, 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 (~500,000) in the trajectories of the monitored vessels. 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. 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 subtractories when

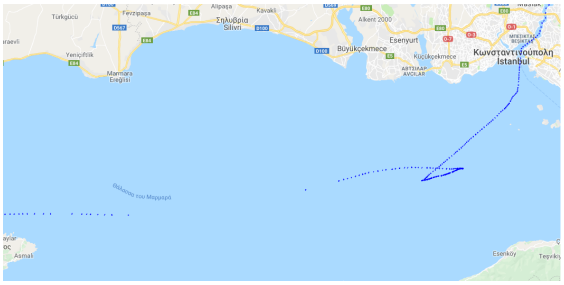

Figure 6: A zoom of the trajectory of a test vessel in the dataset, which has been detected as outlier.

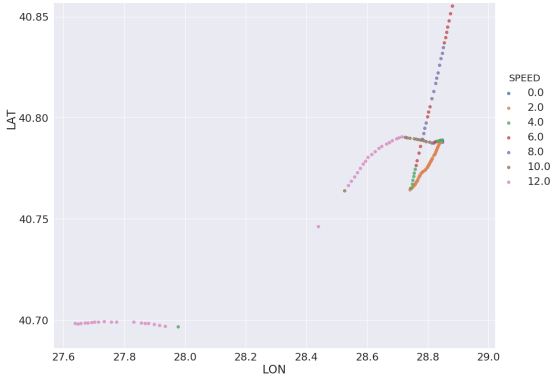

Figure 7: The moving speed details of the trajectory detected as outlier.

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 subtrajectories 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 section, we found cases of vessels 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 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.

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 6 shows an example of such trip, which has been found 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 a very slow moving speed after the gap and that it moved slowly during the maneuver (Figure 7).

5.2 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.

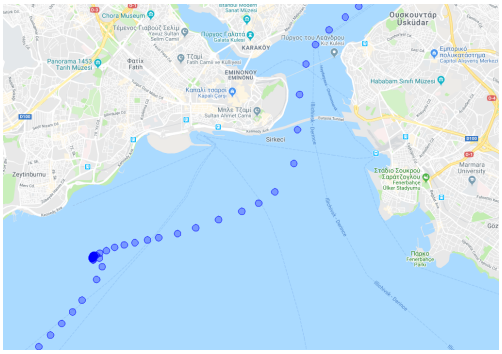

Figure 8: A trajectory that has been found as outlier because of an unusual stop.

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 abnormal behavior, for example, stops and starts, or moves slowly in some parts or during the whole path e.g. because of an engine problem.

In order 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 with a threshold of 95% (this means that the 5% of the points that lie further than all the others from the centroid are considered as outliers), we characterize some vessels as outliers. For a better view of the vessels' trips and in order to avoid short-term deviations, we repeat this process for more neighboring edges.

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 Figure 8) 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 abnormal 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 abnormal behavior at sea that engage more than one vessels.

The proposed network model is quite abstract to achieve a 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 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 constructing 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 abnormalities that these two techniques can detect and ii) to compile a dataset of normal and abnormal behaviors and test the performance of our model in supervised setups.

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 abnormalities to detect and having a human-reviewed dataset with cases of vessels that performed such abnormal behaviors in the area ([19], [28]), will allow us to exploit the proposed model, develop and evaluate new algorithms for the detection of related events.

ACKNOWLEDGEMENTS

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] Virginia Fernandez Arguedas, Giuliana Pallotta, and Michele Vespe. 2018. Maritime Traffic Networks: From historical positioning data to unsupervised maritime traffic monitoring. *IEEE Transactions on ITS* 19, 3 (2018), 722–732.

[2] Luca Cazzanti and Giuliana Pallotta. 2015. Mining maritime vessel traffic: Promises, challenges, techniques. In *OCEANS 2015-Genova*. IEEE, 1–6.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 15.

[4] Pasquale Coscia, Paolo Braca, Leonardo M Millefiori, Francesco AN Palmieri, and Peter Willett. 2018. Multiple Ornstein-Uhlenbeck Processes for Maritime Traffic Graph Representation. *IEEE Trans. Aerospace Electron. Systems* (2018).

[5] Renata Dividino, Amilcar Soares, Stan Matwin, Anthony W Isenor, Sean Webb, and Matthew Brousseau. 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

[6] David H Douglas and Thomas K Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10, 2 (1973), 112–122.

[7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 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, 226–231. http://dl.acm.org/citation.cfm?id=3001460.3001507

[8] Mohammad Etemad. 2018. Transportation Modes Classification Using Feature Engineering. *PhD Thesis, Dalhousie University, CA. arXiv preprint arXiv:1807.10876* (2018).

[9] Mohammad Etemad, Amílcar Soares Júnior, and Stan Matwin. 2018. Predicting Transportation Modes of GPS Trajectories using Feature Engineering and Noise Removal. In *31st Canadian Conference on Artificial Intelligence*. Springer, 259–264.

[10] Anders Holst, Björn Bjurling, Jan Ekman, Åsa Rudström, Klas Wallenius, M Björkman, Farzad Fooladvandi, Rikard Laxhammar, and J Trönninger. 2012. A joint statistical and symbolic anomaly detection system: Increasing performance in maritime surveillance. In *15th International Conf. on Information Fusion*. IEEE, 1919–1926.

[11] Ioannis Kontopoulos, Giannis Spiliopoulos, Dimitrios Zissis, Konstantinos Chatzikokolakis, and Alexander Artikis. 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)*.

[12] Rikard Laxhammar. 2008. Anomaly detection for sea surveillance. In *11th International Conference on Information Fusion*. IEEE, 1–8.

[13] J Mao, C Jin, Z Zhang, and A Zhou. 2017. Anomaly detection for trajectory big data: Advancements and framework. *Ruan Jian Xue Bao/J. Softw.* 28, 1 (2017), 17–34.

[14] Jiali Mao, Pengda Sun, Cheqing Jin, and Aoying Zhou. 2018. Outlier Detection over Distributed Trajectory Streams. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 64–72.

[15] Fabio Mazzarella, Virginia Fernandez Arguedas, and Michele Vespe. 2015. Knowledge-based vessel position prediction using historical AIS data. In *Sensor Data Fusion: Trends, Solutions, Applications*. IEEE, 1–6.

[16] Nirvana Meratnia and A Rolf. 2004. Spatiotemporal compression techniques for moving point objects. In *International Conference on Extending Database Technology*. Springer, 765–782.

[17] Giuliana Pallotta, Steven Horn, Paolo Braca, and Karna Bryan. 2014. Context-enhanced vessel prediction based on Ornstein-Uhlenbeck processes using historical AIS traffic patterns: Real-world experimental results. In *17th international conference on Information Fusion*. IEEE, 1–7.

[18] Giuliana Pallotta, Michele Vespe, and Karna Bryan. 2013. Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction. *Entropy* 15, 6 (2013), 2218–2245.

[19] Kostas Patroumpas, Elias Alevizos, Alexander Artikis, Marios Vodas, Nikos Pelekis, and Yannis Theodoridis. 2017. Online event recognition from moving vessel trajectories. *GeoInformatica* 21, 2 (2017), 389–427.

[20] Lokukaluge P Perera, Paulo Oliveira, and C Guedes Soares. 2012. Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction. *IEEE Transactions on Intelligent Transportation Systems* 13, 3 (2012), 1188–1200.

[21] Bradley J Rhodes, Neil A Bomberger, Michael Seibert, and Allen M Waxman. 2005. Maritime situation monitoring and awareness using learning mechanisms. In *MILCOM 2005*. IEEE, 646–652.

[22] Branko Ristic, Barbara F La Scala, Mark R Morelande, and Neil J Gordon. 2008. Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction.. In *FUSION*. 1–7.

[23] Luis Felipe Sánchez-Heres. 2018. Simplification and Event Identification for AIS Trajectories: the Equivalent Passage Plan Method. *The Journal of Navigation* (2018), 1–14.

[24] Amílcar Soares Júnior, Chiara Renso, and Stan Matwin. 2017. ANALYTiC: An Active Learning System for Trajectory Classification. *IEEE Computer Graphics and Applications* 37, 5 (2017), 28–39.

[25] Emmanuel Stefanakis. 2016. mR-V: Line Simplification through Mnemonic Rasterization. *GEOMATICA* 70, 4 (2016), 269–282.

[26] Titus Tienaah, Emmanuel Stefanakis, and David Coleman. 2015. Contextual Douglas-Peucker simplification. *Geomatica* 69, 3 (2015), 327–338.

[27] Angelos Valsamis, Konstantinos Tserpes, Dimitrios Zissis, Dimosthenis Anagnostopoulos, and Theodora Varvarigou. 2017. Employing traditional machine learning algorithms for big data streams analysis: The case of object trajectory prediction. *Journal of Systems and Software* 127 (2017), 249–257.

[28] Iraklis Varlamis, Konstantinos Tserpes, and Christos Sardianos. 2018. Detecting Search and Rescue missions from AIS data. In *2018 IEEE 34th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 60–65.

[29] Peter Yap. 2002. Grid-based path-finding. In *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer, 44–55.

[30] Liangbin Zhao and Guoyou Shi. 2018. A method for simplifying ship trajectory based on improved Douglas–Peucker algorithm. *Ocean Engineering* 166 (2018), 37–46.

[31] Liangbin Zhao, Guoyou Shi, and Jiaxuan Yang. 2018. Ship Trajectories Preprocessing Based on AIS Data. *The Journal of Navigation* (2018), 1–21.