

# Unsupervised Extraction of Knowledge from S-AIS Data for Maritime Situational Awareness

Nicolas Le Guillaume<sup>\*†</sup>, Xavier Leroureur<sup>†</sup>

<sup>\*</sup>National Institute of Applied Sciences  
Avenue de l'Université, 76801 Saint-Étienne-du-Rouvray, France  
nicolas.le\_guillaume@insa-rouen.fr

<sup>†</sup>Cassidian - An EADS Company  
Parc d'Affaires des Portes, 27106 Val-de-Reuil, France  
{nicolas.leguillaume,xavier.leroureur}@cassidian.com

**Abstract**—Automatic vessel behaviour analysis is a key factor for maritime surveillance and relies on an efficient representation of knowledge about vessels activity. Emerging technologies such as space-based AIS provides a new dimension of service and creates a need for new methods able to learn a maritime scene model at an oceanic scale. In this paper, we propose such a framework: a probabilistic normalcy model of vessel dynamics is learned using unsupervised techniques applied on historical S-AIS data and used for anomaly detection and prediction tasks, thus providing functionalities for high-level situational awareness (level 2 and 3 of the JDL).

## I. INTRODUCTION

Approximately 90% of all international trade is carried out by sea, which represents more than 50,000 vessels sailing the oceans each day. Apart from this normal traffic, the maritime domain is also an important theater for unlawful activities, such as terrorism, piracy, drug/arms smuggling and illegal immigration. Permanent surveillance of oceans is therefore of first importance for homeland security. Maritime situational awareness is a key factor for maritime security. It aims at analysing and understanding maritime activity, at a given time, in order to perform effective decision making and situation projection. The AIS (Automatic Identification System), a self-reporting system used to exchange dynamic (position, speed, heading, etc.) and static (name, cargo, destination, etc.) information between ships and shore-based stations in near real time, is currently the main source of information for maritime surveillance [1]. However, its restricted range (20 to 40 nautical miles) limits its use to coastal areas. Space-based AIS, with a coverage of approximately 2700 nautical miles ( $\approx 5000$  km) in diameter [2], provides a global coverage of the maritime domain including in the open sea. However, this extended range implies some technical issues that affect the detection rate and the information update frequency, making it less suitable for real-time applications, especially in high-density areas. As a consequence, both systems should be considered as complementary tools for global maritime situational awareness. Moreover, due to their collaborative nature, AIS/S-AIS data can be made erroneous, intentionally or not. The reliability of the transmitted information thus depends on the ship's will to cooperate. We state that an operational anomaly detection

system must rely on both type of data (collaborative and non-collaborative).

In the context of maritime surveillance, an important aspect of maritime situational awareness is the capability of a system to detect an anomalous behaviour, i.e. "*a deviation from the expected*" [3]. When considering the global density of maritime traffic, we can consider suspicious behaviours as rare and unusual events deviating from normal ship activity. Consequently, when monitoring a situation, human operators are often overwhelmed with a massive amount of irrelevant data corresponding to normal behaviours, which can be an obstacle to effective anomaly detection. There is a clear need for automatic behaviour analysis tools able to assist human operators by detecting deviations from the maritime traffic routine. A popular approach consists in using trajectory dynamics analysis to characterise objects behaviour in a scene from tracking data and to provide low-level situational awareness [4]. A more complex model of normality can then be obtained using higher-level information.

The purpose of this paper is to present a methodology that makes it possible to extract relevant knowledge from space-based AIS data through the use of unsupervised learning techniques. These techniques assume that the majority of the training instances correspond to a normal behaviour. The proposed algorithm builds a normalcy model of the maritime situation from historical S-AIS data. Using trajectory learning, our method discovers some specific patterns in the data (mainly maritime lanes or stop areas). These patterns and the transitions between them are probabilistically modeled, thus giving birth to a two-level normalcy model which can be used to perform statistical maritime anomaly detection, path prediction and AIS data consistency analysis. This paper is organised as follows. In Section 2, we discuss the topic of trajectory learning and we show that it has not been extensively used for maritime surveillance and anomaly detection. Section 3 describes the methodology we use to learn our normalcy model of vessel behaviour from S-AIS data. In Section 4, we show how this normalcy model can be used to perform various behaviour analysis tasks. Finally, Section 5 concludes the paper and states future directions.

## II. RELATED WORK

The goal of trajectory dynamics analysis is to create a scene model from normal trajectory data that can be used for behaviour analysis and understanding. As normal trajectories are numerous, repetitive and rather constrained by the environment, this normalcy model of vessel dynamics can be learned from data using machine learning techniques. This data-driven approach turns out to be more adapted to large-scale surveillance applications than knowledge-driven systems, especially in the context of maritime surveillance since operators may face quite unconventional situations [5]. A common framework for scene modeling using trajectory dynamics analysis is POI/AP learning [4]. In a maritime surveillance scenario, points of interest (POIs) will correspond to possible destinations of vessels (ports, fishing areas, etc.) and will be characterised by a specific dynamic behaviour (e.g. vessels remain stationary for a certain amount of time [6]). Activity path (AP) learning aims at discovering motion-patterns which model the motion of an object between two POIs, thus defining a graph structure for the scene model [7]. This framework has been extensively used in the context of video surveillance for path discovery [8], [9], anomaly detection [10]–[12] and activity prediction [13], [14].

### A. Trajectory learning for visual surveillance

Trajectory learning is the process of learning motion-patterns from trajectory data using unsupervised techniques, mainly clustering algorithms. According to Morris and Trivedi [4], trajectory learning generally follows a three-step procedure: trajectory preprocessing, trajectory clustering and path modeling. Trajectory data are variable length time-series due to disparities in sampling rates and observation period durations. In [15], Warren Liao distinguishes two major approaches for time-series clustering: the raw-data-based approach consists in modifying existing algorithms, generally by replacing the similarity measure, in order to handle unequal length data; in the second approach, raw data are preprocessed using normalization [9], [13] and/or reduction techniques [8] before a standard clustering algorithm is applied. However, normalization techniques, often based on resampling, may result in a loss of temporal information and the efficiency of reduction techniques is totally dependent on the quality of the trajectory model chosen a priori.

The goal of trajectory clustering is to discover paths in a set of unlabeled data by grouping similar trajectories together. The average Euclidean distance is a common choice to measure the similarity between equal length trajectories [11], [13]. On the contrary, the Hausdorff distance [10] and alignment-based measures can handle unequal length data. In [16] and [17], different similarity measures are evaluated on trajectory clustering tasks. It appears that performance is mainly dependent on trajectory properties: the Euclidean distance with reduction techniques (e.g PCA) performs well on long trajectories while alignment measures are more effective when speed is considered. Based on the surveys by Morris and Trivedi [4] and Warren Liao [15], we can identify five major categories of algorithms used for trajectory clustering : iterative optimization (fuzzy c-means [13], spectral clustering [11], [18]), online adaptive [12], hierarchical methods [9], [10], self-organizing maps and density-based methods [19].

According to us, the choice of a clustering algorithm for trajectory learning is led by two main requirements: 1) it must not require the number of clusters to be known a priori and 2) it must be robust to outliers since the learning set may contain anomalous trajectories. Density-based clustering seems to be particularly well-suited for trajectory learning since it meets the aforementioned requirements [20].

Path modeling consists in creating a compact representation of clusters in order to perform efficient inference. Two main approaches can be found in the literature. Complete path modeling deals with entire trajectories. A path is generally represented by the average trajectory of the cluster, which describes the overall movement within this path, and an envelope describing its spatial extent or the variance of sample trajectories around the average trajectory [10], [13], [14]. Sub-path modeling results from the clustering of subtrajectories. Trajectories are first partitioned in atomic elements and then clustered as regular trajectories [18], [19]. Resulting sub-paths can be modeled using an average trajectory + envelope representation. In addition, the connections between sub-paths must be defined, for example statistically [12].

### B. Previous approaches to maritime surveillance

Maritime surveillance is another popular application scope for scene modeling using trajectory dynamics analysis [1]. However, most of the methods do not involve an explicit stage of trajectory clustering in the model learning procedure and focus on individual vessel observations instead of trajectories. A very common approach consists in estimating a statistical model of observations distribution: multivariate GMM [5], Parzen windows [21], Bayesian networks [22], Gaussian processes [23], etc. Point-based methods are limited because they do not take into account the behaviour over time since they do not model the relationship between successive data points. Some trajectory-based approaches were proposed to adress this issue. In [24], Dahlbom and Niklasson apply the online trajectory learning method previously proposed in [12]. The authors raise some technical issues when applying this clustering algorithm in a coastal surveillance scenario. However, the identified weaknesses are inherent to prefix matching-based algorithms, questioning the usefulness of such algorithms for maritime anomaly detection. The authors propose an alternative spline-based clustering method which is not based on prefix matching. In [25], a two-level approach is described : the distribution of observations is first modeled as a GMM using the EM algorithm. Then, trajectories are represented as sequences of Gaussian components and used to learn an HMM structure. In [26], the effects of trajectory compression on various machine-learning tasks (clustering, classification and outlier detection) are studied. Experiments on AIS data show that trajectory compression has a positive effect on all tasks, especially for outlier detection where performance is significantly improved. Finally, authors describe a method to enrich the model with contextual knowledge which leads to better clustering and classification results.

## III. LEARNING A TWO-LEVEL NORMALCY MODEL

In this section, we describe a method to build a normalcy model of ship trajectories using the POI/AP framework and trajectory learning. This method consists of three main parts:

trajectory partitioning, clustering and path modeling. Our vessel tracking data are S-AIS position reports. From raw AIS messages, we extract vessel trajectories using the following representation. Let  $T$  be a vessel trajectory

$$T = (MMSI, \{(f_1, t_1), (f_2, t_2), \dots, (f_{|T|}, t_{|T|})\}),$$

where the  $MMSI$  is a unique 9-digit ship identifier and  $f_i$  is a feature vector describing the vessel dynamics at time  $t_i, t_i < t_{i+1} \forall i \in \{1, \dots, |T| - 1\}$ . Feature selection is an important aspect of POI/AP learning and depends on the expected level of representation. The minimal set of information for path discovery is the location (latitude and longitude) of the ship. However, if we want to distinguish two lanes belonging to the same path but oriented in opposite directions, we need directional information, namely the Course Over Ground (COG). Since we also want to detect POIs corresponding to stop areas, we incorporate the Speed Over Ground (SOG) in our trajectory representation. Consequently,  $f_i = [x_i, y_i, v_i, \theta_i]^T$ , where  $(x_i, y_i)$  are the cartesian coordinates of the ship obtained using an appropriate projection method,  $v_i$  is the SOG and  $\theta_i$  is the COG at time  $t_i$ . We work with raw trajectory data (no normalization, no reduction technique). Our choice is motivated by three main facts : 1) we keep the intuitive notion of similarity between trajectories, 2) the issue of unequal length data can be addressed without loss of information by using an appropriate similarity measure and 3) the choice of an appropriate trajectory model when performing reduction techniques is a critical issue that can affect the significance of clusters [4].

#### A. Trajectory partitioning

We illustrate the main advantages of trajectory partitioning over a whole trajectory approach with two examples.

**Example 1.** Consider the trajectories in Figure 1 describing a situation quite typical of a maritime crossroads. If we perform clustering on these trajectories as a whole, we cannot detect the common part of the different behaviours, hence we are not able to correctly model the crossroads structure.

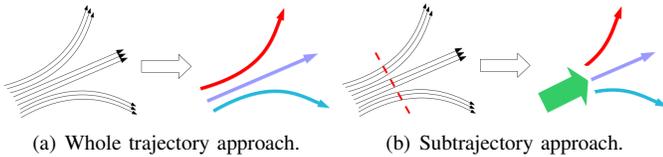


Fig. 1. Trajectory clustering using both approaches.

In contrast, if we adopt a subtrajectory approach by partitioning the trajectories at appropriate locations, the clustering algorithm is able to detect similar portions of trajectories. The clusters represent more significative, atomic motion-patterns, thus making the model more informative.

**Example 2.** Consider the trajectories in Figure 2. Obviously, the overall behaviour is the same for all trajectories. However, one of them has an outlying portion which may correspond to an abnormal behaviour. Detecting such an anomalous subtrajectory is of highest importance since our data set may contain abnormal trajectories we want to filter out before learning the model. The whole trajectory approach fails to detect this outlying section because the dissimilarity is

averaged out over the whole trajectory. On the contrary, using trajectory partitioning, the anomalous part is correctly recognized as dissimilar with respect to the neighboring trajectories and removed from the learning set.

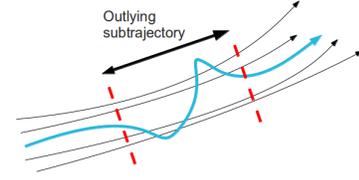


Fig. 2. An outlying subtrajectory.

In the following, we describe a two-level partitioning method. The first level consists in a semantic partitioning of trajectories in order to identify stop and move sections. The second level is a spatial partitioning of move subtrajectories which aims at extracting atomic elements of motion.

1) *Semantic partitioning*: A trajectory can be seen as a sequence of moves and stops [27]. A stop is a non-empty time interval during which the object does not move from the application perspective. Two stops are necessarily disjoint. A move is a subtrajectory delimited by two consecutive stops or the first/last observation of the trajectory. Two main benefits emerge from the partitioning of trajectories into stop and move subtrajectories. First, stops are indicative of interesting places and can then be used to discover POIs in vessel trajectory data [1]. Secondly, stops are an important source of noise for trajectory clustering and do not contain any motion information which could be used for path learning. Stop sections can then be removed from the learning set to improve the quality of learned paths.

Semantic partitioning of trajectories is carried out using the CB-SMoT (Clustering-Based Stops and Moves of Trajectories) algorithm [6], which consists in detecting the slowest sections of a trajectory using spatio-temporal clustering. CB-SMoT is based on the well-known density-based clustering algorithm DBSCAN. Given a set of points  $D$ , this one-pass algorithm iteratively looks for core points, i.e. points  $p_i \in D$  such that  $|N_\epsilon(p_i)| \geq \text{minLns}$ , where  $N_\epsilon(p_i)$  is the  $\epsilon$ -neighborhood of  $p_i$ . Once such a point is found, a new cluster is built. Then each point in its  $\epsilon$ -neighborhood is tested and if it verifies the core condition, its own  $\epsilon$ -neighborhood is merged with the cluster. This procedure is repeated until no new point can be added to the new cluster, i.e. we reach the border points. Some basic concepts of DBSCAN are illustrated in Figure 3.

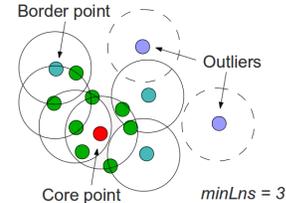


Fig. 3. Some basic concepts of DBSCAN for  $\text{minLns} = 3$ .

Following the same principles, CB-SMoT looks for core observations  $f_i$  in a trajectory  $T$  and builds a subtrajectory by aggregating other points in the neighborhood if their speed

is less than a *maxSpeed* value and if the average speed of the cluster remains under an *avgSpeed* threshold. Once the cluster is maximal, it is considered as a stop if the duration between the first and the last observation exceeds a *minTime* parameter. After execution of CB-SMoT, a trajectory  $T$  is partitioned into a set of move subtrajectories and a set of stop subtrajectories. Each move starts and/or ends in a stop.

2) *Spatial partitioning*: We now apply a spatial partitioning algorithm on move trajectories in order to extract atomic elements of motion, generally corresponding to rather linear behaviours, i.e. the heading is almost constant over the whole subtrajectory. This approach helps us to address the issues mentioned in the two previous examples. A common choice is the Piecewise Linear Segmentation (PLS) [26] but we prefer a simple Sliding Window algorithm which is particularly attractive because of its simplicity, its intuitiveness and, as long as the application is concerned, the fact that it is an online algorithm. Moreover, in some interesting cases, the Sliding Window algorithm is less disposed to over-fragmentation, for example in the case of a regular U-turn. This algorithm consists in iteratively checking if the distance between a point of the trajectory and its orthogonal projection on the straight line defined by the two previous observations exceeds or not a threshold  $\lambda$ . If it is the case, the trajectory is split at the previous observation. This method allows to iteratively detect changes in vessel heading based on location information, which is more reliable than the COG for trajectory partitioning.

## B. Trajectory clustering

In this section, clustering is applied on stop and move subtrajectories separately in order to detect POIs in stops and motion-patterns in moves. A clustering method is defined by a similarity measure, a clustering procedure and an evaluation criteria [15].

1) *Similarity measures*: As stated before, the two main approaches to measure the similarity between unequal length time series are the Hausdorff distance and alignment measures. However, the former cannot distinguish the trajectories in the same path but oriented in opposite directions [16]. Many different alignment measures exist in the literature but it can generally be reduced to two families : the Dynamic Time Warping (DTW) on one hand and the Longest Common SubSequence (LCSS) and other forms of Edit Distance on the other hand. The DTW is a well-known alignment measure extensively used in speech recognition, computer vision and signal processing [28]. The DTW looks for a non-linear, continuous alignment between two sequences (all points of both sequences are aligned) that minimizes a warping cost. On the contrary, the LCSS allows some elements to be unmatched, thus making the measure less sensitive to noise and outliers [17]. We adopt the DTW to measure the similarity between move trajectories for two main reasons : 1) this measure is parameter-free and 2) it is sensitive to outlying observations, which helps us detect and filter out anomalous trajectories from our learning set.

Stops were previously identified as low speed portions of trajectories. Our objective is to cluster these stops in order to detect some POIs. Here we do not want to learn how vessels move but where they stop, i.e. the location and the spatial

extent of these POIs. Consequently, we no longer take into account the dynamics of the ship, characterised by the SOG, the COG and the time ordering of observations. We forget the trajectory structure of stops and deal only with individual observations. We can then use a simple Euclidean distance to calculate the spatial proximity between two ship observations.

2) *Clustering procedure*: At this stage of the method, the distances calculated for moves and stops are stored in their respective distance matrix. The implementation of a distance matrix is a critical issue since it requires the storage of one value for each pair of move subtrajectories and for each pair of stop observations. The size of the matrix then grows very quickly even with a quite limited data set. In our implementation, each line of both matrices is stored in a distinct file. These matrices will then serve as an input for a one-pass clustering algorithm. Consequently, each line is read once. Density-based clustering methods seem to be the most appropriate approach for both moves and stops clustering. The key idea of these algorithms is that for each member of a cluster, the cardinality of its  $\epsilon$ -neighborhood must exceed a *minLns* threshold. Since outliers generally belong to sparse regions of the feature space, they are not assigned to any cluster.

The OPTICS algorithm, described by Ankerst *et al.* in [29], is a generalization of DBSCAN that solves many of its technical issues, including its sensitivity to input parameters [20]. OPTICS is a one-pass algorithm that stores the order in which data are processed and the information that would be used by DBSCAN to assign cluster membership, namely the core-distance and the reachability-distance. Given a generating distance  $\epsilon$ , the core-distance of a core trajectory  $T$  is the smallest distance  $\epsilon'$ ,  $0 \leq \epsilon' \leq \epsilon$ , between  $T$  and a trajectory in  $N_\epsilon(T)$  such that  $T$  would be a core trajectory. The reachability-distance of a trajectory  $T'$  w.r.t.  $T$  is generally the distance between both objects unless they are too close, then this value is normalized using the core-distance of  $T$ . The order of objects and their reachability-distance are represented in a *reachability plot*. This bar chart is a graphical representation of the density-based clustering structure of the data set. The reachability-distance of a trajectory can be seen as the minimum distance from the set of all its predecessors [20]. Consequently, a high reachability-distance means that the object is located far away from all other objects, i.e. it belongs to a quite sparse area, and a small reachability-distance means that the trajectory belongs to a dense region of the feature space. It is quite intuitive then that clusters correspond to valleys separated by low density regions represented by peaks in the reachability plot. This correspondence is illustrated in Figure 4.

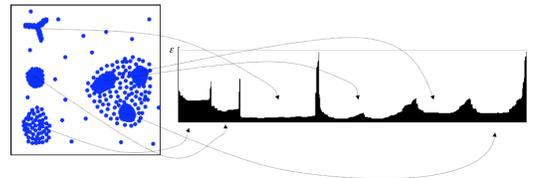


Fig. 4. Correspondence between the clustering structure of data and the reachability plot.

Ankerst *et al.* also describe a method to automatically extract clusters at different levels of granularity. The general idea of this algorithm is to identify all potential starts and ends

of clusters (respectively downward and upward-trend areas of the reachability plot) and to combine this information to detect the whole hierarchy of clusters as well as the outlying trajectories. Thanks to this algorithm, we can handle situations where clusters exist at different density levels, such as the one depicted in Figure 5. Using the method above, we can simultaneously discover clusters A, B, C and a, b, c, which would not be possible using a global density parameter.

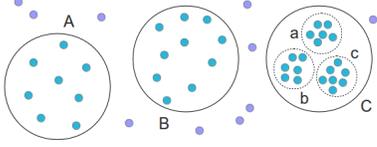


Fig. 5. Clustering using different density parameters.

3) *Evaluation of clusters quality*: The OPTICS algorithm can not only cluster data according to their density in the feature space but also learn the whole hierarchy of clusters extracted at different density levels. However, all these clusters are not equally informative: some of them are either too high in the hierarchy (many relevant clusters are merged in one non-informative cluster) or too low (several clusters are part of a more informative cluster at a lower density level). The key question is to automatically determine which clusters are relevant for building our normalcy model and which clusters are non informative because too general or too specific. This question is directly linked to the issue of evaluating clusters quality which is not a trivial task since we cannot rely on any ground truth. Currently, we need some expert knowledge to filter out irrelevant clusters prior to model motion-patterns and POIs. However, there is a strong need for an automatic method capable of selecting the more informative clusters.

### C. Multi-scale path modeling

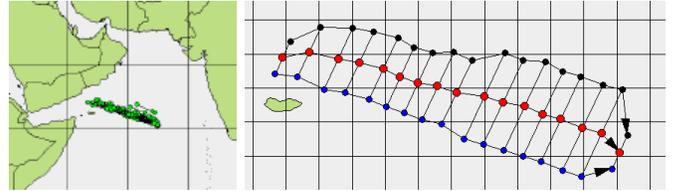
We adopt a graph structure for our scene model: the nodes are the states of the model and the edges are the transitions between states. A state is a compact, probabilistic representation of a cluster. We define three types of states: stops, motion-patterns and junctions. Transitions are then modeled using a simple Markov chain. States can be seen as local activity models whereas the connections between states form the global model thus giving birth to a two-scale normalcy model.

1) *States modeling*: A motion-pattern, or activity path, is a representation of the motion of vessels in a restrained area. Ideally, motion-patterns correspond to shipping routes and are learned from the clusters extracted from move trajectories using position and heading as features. The speed information is incorporated during the modeling process. We then need a probabilistic representation that can handle three types of information: position (spatial extent of the path), speed (normal speed range) and direction (normal heading range). Our approach is largely inspired by work of Makris and Ellis [7]. A motion-pattern is represented as a sequence of node, each node being characterised by:

- a node center  $n_i = (x_i, y_i)$ . The sequence of node centers defines the average trajectory;
- a normal vector  $\mathbf{n}_i$ , orthogonal to the global average direction  $\bar{\theta}$  of the motion-pattern;

- three normal distributions: signed distance to  $n_i$ , speed and heading of observations along the normal vector;
- a left boundary and a right boundary that encode the spatial extent of the path along the normal vector. The set of boundaries defines the envelope of the path.

We use a sweep line approach similar to the one described in [19] to learn the sequence of nodes. The method consists in sweeping a vertical line in the average direction  $\bar{\theta}$ . Each time the line meets a new observation, virtual observations corresponding to intersections with the sample trajectories are linearly interpolated. The average coordinates of those virtual observations define a new node center and the value of their signed distance, speed and heading is used to estimate the different distributions. Finally, the boundary points are learned using the 68 – 95 – 99.7 rule, thus defining the envelope of the path as a tolerance interval for vessels position. Figure 6 shows the path representation of a cluster extracted from move trajectories in the Arabian Sea.



(a) Cluster of trajectories. (b) Motion-pattern (95% tolerance interval).  
Fig. 6. A move cluster in the Arabian Sea and the corresponding path.

A stop is a POI that represents an area where a vessel is stationary or moving very slowly. In a maritime context, it may correspond to ports, offshore platforms or fishing areas that are potential destinations for ships. Stops are modeled from the clusters discovered in stop observations. We want our representation to estimate the position and the spatial extent of a stop. As a first approximation, we choose a very simple approach that consists in modeling a stop using a bivariate normal distribution  $(\mu = (\hat{x}, \hat{y})^T, \Sigma)$ . A tolerance region is then defined as the set of vector  $k = (x, y)^T$  satisfying

$$(k - \mu)^T \Sigma^{-1} (k - \mu) \leq \chi_{1-\alpha}^2(2),$$

where  $\chi_{1-\alpha}^2(n)$  is the quantile of probability  $1 - \alpha$  of the chi-squared distribution with 2 degrees of freedom. This equation also defines the Standard Deviation Ellipse (SDE) which captures the shape of a 2D-distribution by introducing a directional bias. The value of  $\alpha$  depends on the size of the confidence interval.

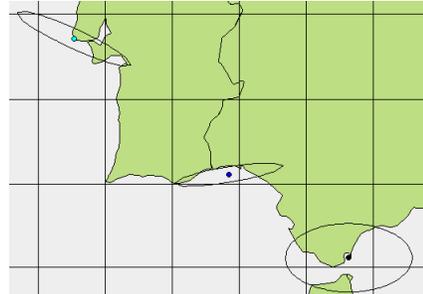


Fig. 7. Some stops discovered along the coast of the Iberian Peninsula.

This approach for stop modeling has two major limitations. First, since ports are located along the coast, a representation which assumes that ships are distributed "around" an average position cannot model accurately such a POI and introduce a bias in the estimation of the location of the port (see the second stop in Figure 7). However, it is quite fitted to the representation of non-coastal POIs. Secondly, the speed information is not incorporated in the stop model.

Junctions correspond to spatial areas where it is normal to find a significant density of ships but not necessarily with similar behaviours. They are used to increase the spatial coverage of our model by introducing some continuity between consecutive motion-patterns/stops. Junctions are learned from the clustering of *normal outlying observations*, i.e. vessel observations belonging to the clusters used to learn the model (hence considered as normal) but which are classified as anomalous w.r.t. the normalcy model by our anomaly detection method. Similarly to stop observations, normal outlying observations are clustered using location as the only feature and junctions are modeled using the same bivariate normal distribution. Even if this practical choice does not model well the points distribution, it is enough, as a first approximation, to illustrate the importance of junctions for model completeness.

2) *Transitions modeling*: We build our global normalcy model using a simple Markov chain to represent the connections between the different local models (the states). A Markov chain is a random process that satisfies the Markov property: given the present state, the future and the past states are independent. A Markov chain is entirely defined by the pair  $(S, M)$  where  $S$  is the state space and  $M$  is the transition matrix which can be learned from the data using the MLE method. Let  $S = \{s_0, s_1, \dots, s_m\}$  be the set of possible states (motion-patterns, stops, junctions and "not in any state" state). First, all normal move trajectories are represented as finite sequences of states. Then, let  $n_{ij}$  be the total number of transitions  $s_i \rightarrow s_j, i \neq j$  among all the trajectories and  $n_i$  the number of transitions from  $s_i$  to any other state different from itself, the transition probability  $p_{ij}$  is given by

$$p_{ij} = \frac{n_{ij}}{n_i}.$$

Since  $p_{ii} \gg p_{ij}, i \neq j$  due to the nature of our states, we need to assume  $p_{ii} = 0, \forall i$  to make path prediction possible.

An example of a normalcy model learned from real-world S-AIS data using the above methodology is given in Figure 8.

#### IV. BEHAVIOUR ANALYSIS

Our two-scale normalcy model can be used for behaviour analysis tasks such as anomaly detection and path prediction. The local model is used to detect dynamic anomalies (incoherent location, speed, heading). The local anomaly detection procedure consists in checking if a new vessel observation fits into one of the states of the model. The global model is used to detect anomalies at the journey-level (unexpected routing, heading towards the wrong destination). It relies on the path prediction mechanism provided by the Markovian representation.

#### A. Online anomaly detection

The online, local anomaly detection process can be divided into two main stages: observation type identification and classification. When receiving a new AIS position report, we first need to determine if this observation belongs to a move or to a stop portion of the trajectory of interest. The observation type identification process is described in Figure 9.

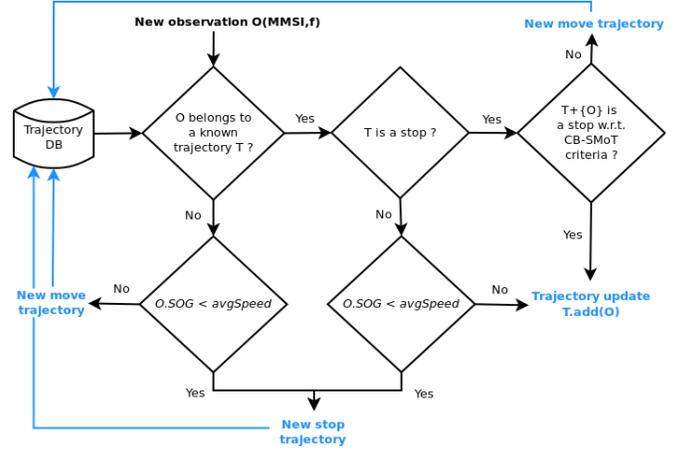


Fig. 9. Observation type identification process.

Once the observation as been identified as either a stop or a move, we can apply the procedure described in Figure 10 to classify the observation as normal, anomalous or unknown.

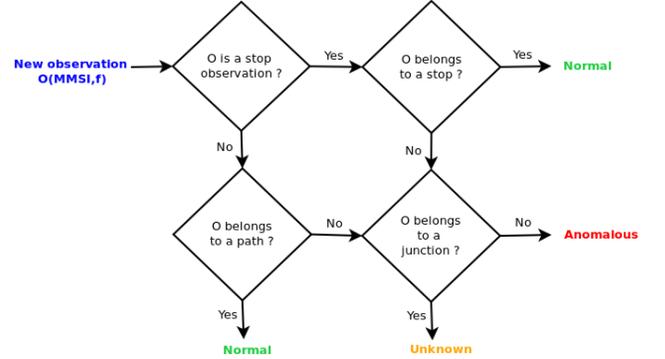


Fig. 10. Observation classification process.

We consider that a new observation  $O$  belongs to a state  $s_i \in S = \{s_0, s_1, \dots, s_m\}$  if  $i = \operatorname{argmax}_{k \in \{0, \dots, m\}} \tau_k(O)$  where  $\tau_k(O) \in [0, 1]$  is the *normalcy score* of  $O$  w.r.t. state  $s_k$ . If  $\tau_i(O) = 0$  then  $O$  is an anomalous observation (it does not belong to any state of the model). The way  $\tau_k(O)$  is calculated depends on the nature of the state. If  $s_k$  is a motion-pattern, the procedure consists in determining the consecutive nodes  $n_1^k$  and  $n_2^k$  such that  $O$  lies within the trapezoid defined by their boundary points. If such a bounding box does not exist,  $O$  is considered as anomalous (since it does not lie within the spatial extent of the path) and  $\tau_k(O) = 0$ . Otherwise, a virtual node  $\hat{n}$  is linearly interpolated so that  $O$  would lie along the normal vector of  $\hat{n}$ . We then perform a simple chi-squared test (assuming the independency between variables) with an  $\alpha$  level of significance by accepting the null hypothesis " $O$  is normal w.r.t.  $s_k$ " if the test statistic  $T_{s_k}(O) \leq \chi_{1-\alpha}^2(3)$ . The

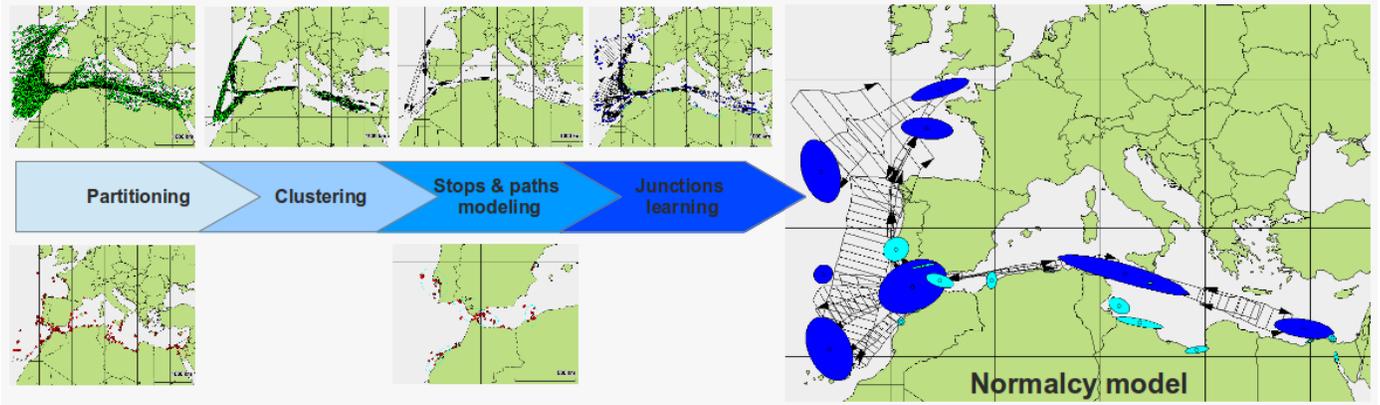


Fig. 8. Our normalcy model learning methodology is applied to a real-world data set containing 3 days of S-AIS position reports recorded over the Mediterranean Sea and the Straight of Gibraltar. After a denoising procedure, our learning set is reduced to 1600 trajectories (8250 vessel observations) that are partitioned in 1450 move subtrajectories and 230 stop observations. The states (25 motion-patterns, 23 stops and 8 junctions) are modeled using a 95% confidence interval. As illustrated by this example, our method can learn a normalcy model of vessels activity at a continental scale in an unsupervised way.

value of  $\alpha$  depends on the size of the confidence interval used when learning the model. The normalcy score  $\tau_k(O)$  is then defined by

$$\tau_k(O) = 1 - \frac{T_{s_k}(O)}{\chi_{1-\alpha}^2(3)}.$$

If  $s_k$  is a stop or a junction, testing if  $O$  belongs to the SDE of  $s_k$  is equivalent to a chi-squared test with correlated variables. The null hypothesis is accepted if  $T_{s_k}(O) = (f - \mu_k)^T \Sigma_k^{-1} (f - \mu_k) \leq \chi_{1-\alpha}^2(2)$ , where  $f$  is the feature vector of  $O$ , and the normalcy score  $\tau_k(O)$  is then defined by

$$\tau_k(O) = 1 - \frac{T_{s_k}(O)}{\chi_{1-\alpha}^2(2)}.$$

The normalcy score represents the degree of normality of an observation and measure our confidence in the result of the detection procedure. A few examples of the anomaly detection capabilities of our system are given in Figure 11.

### B. Path prediction

When traveling along a maritime lane, the journey of a ship is highly predictable. When reaching the end of its current path, different scenarios are conceivable: the ship reaches its destination (a POI) or it enters a new maritime lane. Using the transition model, we can determine, given the current state  $S_t = s_i$  of the ship, the most probable future state  $\hat{S}_{t+1}$ :

$$\hat{S}_{t+1} = \underset{s_j}{\operatorname{argmax}} P(S_{t+1} = s_j | S_t = s_i) = \underset{j}{\operatorname{argmax}} M(i, j).$$

It is important to keep in mind that, according to our model,  $t + 1$  does not represent the time of the next observation but the next transition between two states. A possible strategy for routing anomaly detection is to create a set  $S_\alpha$  of states so that  $P(S_{t+1} \in S_\alpha | S_t = s_i) \geq 1 - \alpha$  and  $\forall s_j \in S_\alpha, \forall s_k \notin S_\alpha, p_{ij} > p_{ik}$ . Then, the routing is considered anomalous if  $S_{t+1} \notin S_\alpha$ . The destination information is stored implicitly at the state-level in our normalcy model. Knowing the current state of a ship, it is possible to predict the most probable destination by simply iterating the future state prediction procedure until a stop is detected as the most probable destination, or until no new stop with a significant probability can be added to the set of most probable destinations. If the destination declared

in the AIS report does not belong to this set, it is then likely that the ship is not heading towards it and an anomaly is raised. However, this procedure assumes that a stop necessarily corresponds to a port, which cannot be ensured in our model.

### C. Consistency of AIS data

Different conclusions can be drawn from the anomaly detection procedure depending on whether we consider AIS static data (destination, type of ship, etc.) as reliable or not. In the first case, we will consider that the behaviour of the ship is normal or abnormal w.r.t. the information in the AIS reports. For instance, if the predictions obtained using the model are different from the declared destination, we raise a "not heading towards the right destination" anomaly. On the contrary, considering this information as untrustworthy, we can use the model to detect inconsistency in AIS data. In the same situation, the anomaly is now "the destination information is inconsistent w.r.t. the vessel's behaviour".

## V. CONCLUSION AND FUTURE WORK

In this paper we presented a framework for maritime scene model learning using unsupervised techniques applied on S-AIS trajectory data. The resulting probabilistic, two-scale normalcy model can be used for behaviour analysis tasks, including statistical anomaly detection, path prediction and AIS consistency analysis. The future work will consist in automatically extracting relevant clusters from the whole clusters hierarchy so that the method does not need expert knowledge anymore. We will also work on the improvement of POIs modeling, especially ports and junctions whose representation is not appropriate, and look for some methods to improve the completeness of the model, particularly in coastal areas. Another objective would be to incorporate additional AIS information into the model, such as the type of the ship since the behaviour of vessels is highly dependent on their nature (tankers, cargos, fishing boats, etc), as well as contextual information, for instance for port identification. Finally, it would be necessary to define a performance measure and to carry out an evaluation campaign in order to determine if our anomaly detector can perform well in real-world (or at least simulated) conditions.

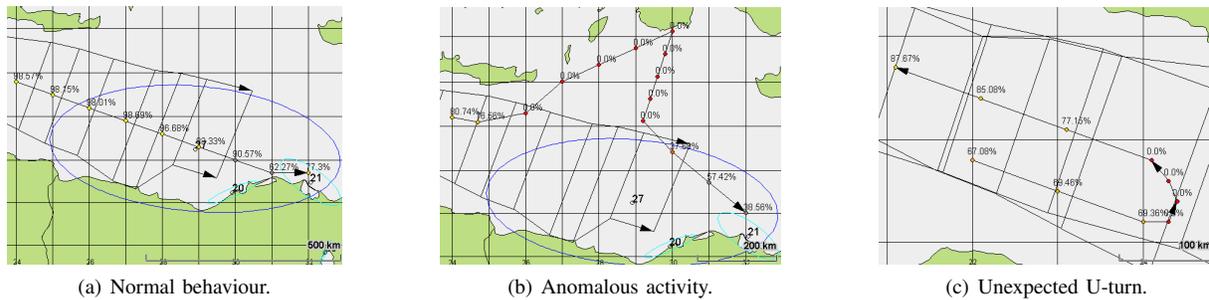


Fig. 11. Illustrations of the anomaly detection capabilities of our method. The normalcy score of each observation is displayed. The blue POI is a junction and light blue ones are stops. In example (a) corresponding to a normal behaviour, state 27 is recognized as the most probable future state (62,49%) and state 21 as the most probable destination (23,07%).

#### ACKNOWLEDGMENT

We would like to thank Roland Barrot, System Architect at ASTRIUM Consulting Toulouse for providing us the real-world space-based AIS data used in our experiments. Nicolas Le Guillaume is now a PhD student in the MAD team of the GREYC laboratory - University of Caen Basse Normandie.

#### REFERENCES

- [1] M. Vespe, I. Visentini, K. Bryan, and P. Braca, "Unsupervised learning of maritime traffic patterns for anomaly detection," in *Data Fusion & Target Tracking Conference (DF&TT 2012): Algorithms & Applications, 9th IET*. IET, 2012, pp. 1–5.
- [2] IALA, "Recommendation A-124 appendix 19 : Satellite AIS considerations," International Association of Marine Aids to Navigation and Lighthouse Authorities, Tech. Rep., 2011.
- [3] J. Roy, "Anomaly detection in the maritime domain," in *Proceedings of SPIE*, vol. 6945. Society of Photo-Optical Instrumentation Engineers, 2008.
- [4] B. Morris and M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1114–1127, 2008.
- [5] R. Laxhammar, "Anomaly detection for sea surveillance," in *11th International Conference on Information Fusion, 2008*. IEEE, 2008, pp. 1–8.
- [6] A. Palma, V. Bogorny, B. Kuijpers, and L. Alvares, "A clustering-based approach for discovering interesting places in trajectories," in *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2008, pp. 863–868.
- [7] D. Makris and T. Ellis, "Path detection in video surveillance," *Image and Vision Computing*, vol. 20, no. 12, pp. 895–903, 2002.
- [8] F. Porikli, "Trajectory pattern detection by HMM parameter space features and eigenvector clustering," in *8th European Conference on Computer Vision, Prague, 2004*.
- [9] X. Li, W. Hu, and W. Hu, "A coarse-to-fine strategy for vehicle motion trajectory clustering," in *18th International Conference on Pattern Recognition, 2006*, vol. 1. IEEE, 2006, pp. 591–594.
- [10] I. Junejo, O. Javed, and M. Shah, "Multi feature path modeling for video surveillance," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004*, vol. 2. IEEE, 2004, pp. 716–719.
- [11] Z. Fu, W. Hu, and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," in *IEEE International Conference on Image Processing, 2005*, vol. 2. IEEE, 2005, pp. II–602.
- [12] C. Picciarelli and G. Foresti, "On-line trajectory clustering for anomalous events detection," *Pattern Recognition Letters*, vol. 27, no. 15, pp. 1835–1842, 2006.
- [13] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1450–1464, 2006.
- [14] B. Morris and M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 425–437, 2008.
- [15] T. Warren Liao, "Clustering of time series data - a survey," *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.
- [16] Z. Zhang, K. Huang, and T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," in *18th International Conference on Pattern Recognition, 2006*, vol. 3. IEEE, 2006, pp. 1135–1138.
- [17] B. Morris and M. Trivedi, "Learning trajectory patterns by clustering: Experimental studies and comparative evaluation," in *IEEE Conference on Computer Vision and Pattern Recognition, 2009*. IEEE, 2009, pp. 312–319.
- [18] Y. Yang, Z. Cui, J. Wu, G. Zhang, and X. Xian, "Trajectory analysis using spectral clustering and sequence pattern mining," *Journal of Computational Information Systems*, vol. 8, no. 6, pp. 2637–2645, 2012.
- [19] J. Lee, J. Han, and K. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of data*. ACM, 2007, pp. 593–604.
- [20] M. Nanni and D. Pedreschi, "Time-focused clustering of trajectories of moving objects," *Journal of Intelligent Information Systems*, vol. 27, no. 3, pp. 267–289, 2006.
- [21] R. Laxhammar, G. Falkman, and E. Sviestins, "Anomaly detection in sea traffic—a comparison of the Gaussian Mixture Model and the Kernel Density Estimator," in *12th International Conference on Information Fusion, 2009*. IEEE, 2009, pp. 756–763.
- [22] F. Johansson and G. Falkman, "Detection of vessel anomalies—a bayesian network approach," in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information, 2007*. IEEE, 2007, pp. 395–400.
- [23] K. Kowalska and L. Peel, "Maritime anomaly detection using Gaussian Process active learning," in *15th International Conference on Information Fusion, 2012*. IEEE, 2012, pp. 1164–1171.
- [24] A. Dahlbom and L. Niklasson, "Trajectory clustering for coastal surveillance," in *10th International Conference on Information Fusion, 2007*. IEEE, 2007, pp. 1–8.
- [25] Š. Urban, M. Jakob, and M. Pěchouček, "Probabilistic modeling of mobile agents' trajectories," *Agents and Data Mining Interaction*, pp. 59–70, 2010.
- [26] G. de Vries and M. Van Someren, "Machine learning for vessel trajectories using compression, alignments and domain knowledge," *Expert Systems with Applications*, 2012.
- [27] S. Spaccapietra, C. Parent, M. Damiani, J. De Macedo, F. Porto, and C. Vangenot, "A conceptual view on trajectories," *Data & knowledge engineering*, vol. 65, no. 1, pp. 126–146, 2008.
- [28] G. ten Holt, M. Reinders, and E. Hendriks, "Multi-dimensional dynamic time warping for gesture recognition," in *In Proc. of the conference of the Advanced School for Computing and Imaging (ASCI 2007)*, 2007.
- [29] M. Ankerst, M. Breunig, H. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," *ACM SIGMOD Record*, vol. 28, no. 2, pp. 49–60, 1999.