Abstract

Maritime anomaly detection requires an efficient representation and consistent knowledge of vessel behaviour. Automatic Identification System (AIS) data provides ships state vector and identity information that is here used to automatically derive knowledge of maritime traffic in an unsupervised way. The proposed approach only utilises AIS data, historical or real-time, and is aimed at incrementally learning motion patterns without any specific a priori contextual description. This can be applied to a single AIS terrestrial receiver, to regional networks or to global scale tracking. The maritime traffic representation underpins low-likelihood behaviour detection and supports enhanced Maritime Situational Awareness by providing a characterisation of vessels traffic.

1 Introduction

Today maritime transportation represents 90% of global trade volume. The challenges related to safety and security aspects are therefore of high priority at the international level. Self-reporting systems such as AIS, although conceived for safety applications, are currently employed to achieve a higher level of situational awareness. Ships of 300 gross tonnage and upwards in international voyages, 500 and upwards for cargos not in international waters and passenger vessels are obliged to be fitted with AIS equipment [1]. Moreover, AIS will be required for fishing vessels with a length greater than 15 m and sailing in water under the jurisdiction of Member States of the European Union [2]. Although only giving indications on a portion of maritime traffic, AIS is a worldwide standard and therefore a coherent source of information for global traffic analysis. Moreover, with the advent of AIS terrestrial networks, and satellite receiver constellations [3], AIS nowadays can represent a global and near real-time source of information.

In addition to AIS broadcast data inconsistencies and anomalies related to vessels manoeuvres, security threats can be associated to generic deviations from the maritime traffic routine. In this paper, such contextual "normality" is first modelled, then learned for ultimate use as a reference for the detection of abnormal behaviours. Given the large amount of AIS data at global scale, both the learning and anomaly detection processes are required to be achieved with a high degree of automation in order to reduce and synthesise the vast amount of information available to human operators.

The problem of learning motion patterns has been analysed in different domains like video surveillance (e.g. [4] and [5]) and mobility data classification [6]. In contrast to other applications, AIS provides not only the vessel state vector (e.g. position, speed with GPS accuracy) but also identification (e.g. Maritime Mobile Service Identity (MMSI), call sign etc.) and classification (e.g. vessel type and size) information. Although the possibility of spoofing is an issue with self-reporting systems, this is the exception and is not addressed in this analysis.

Some AIS anomaly detection methodologies have been developed by subdividing the area of interest into a grid and describing the vessels state vector properties within each cell [7, 8, 9]. This approach is particularly well suited for port security and small area surveillance since the number of grid elements can be managed; however, it would be infeasible to manage a grid cell approach for global scale applications, and an accurate description of vessels behaviours requires a small enough cell size. Also, in areas where traffic is complex, the grid based approach is less efficient. If two or more routes intersect, the behavioural description in the relevant cells would be described by multimodal distributions increasing the complexity of the representation and algorithms.

An alternative method to the grid cell approach and well suited to global analysis is a vectorial representation of traffic, i.e. trajectories can be thought of as a set of lines connecting nodes. In [10, 11], nodes are mainly considered in proximity of land masses, reconstructing ocean journeys along Great Circle routes. Nevertheless, in areas characterised by complex traffic, it is necessary to introduce nodes in correspondence to changes of Course Over Ground (COG) [12]. A problem similar to AIS based maritime trajectory analysis is the airspace monitoring using flight tracks data [13]. Turning points are detectable on the basis of course changes, and their densities identify waypoints areas. Historical data are then clustered and their trajectories centroids defined. In contrast to air traffic control applications, maritime surveillance problems involve two-dimensional space, which reduces the trajectory domain complexity. Nonetheless, the motion of vessels is some-

---

1 The term "spoofing" is here used to address the intentional transmission of fake static (e.g. identity) and dynamic (e.g. state vector) information.
what irregular, making the detection of turning points rather
difficult especially in unregulated areas.

In this paper, maritime motion patterns are automatically
learned using unsupervised algorithms by analysing the
AIS data flow, suitable to historic or real time analysis. This
is done by modelling vessels as objects whose behaviour is
progressively used to detect and shape waypoints (i.e. ports,
offshore platforms, turning points, entry or exit points), sea
lanes and routes. The methodology, described in Section
2, is also conceived to allow incremental learning, in order
to dynamically adapt the traffic representation to changing
(e.g. seasonal patterns) or evolving situations. Results us-
ing terrestrial and satellite based AIS messages are shown
in Section 3, whereas conclusions and future work direc-
tions presented in Section 4.

2 Traffic Data Model

Unsupervised knowledge discovery from historical or real-
time AIS data is a task whose reliability depends on how
well the context is captured from the observations. In order
to achieve a consistent traffic representation, it is necessary
to take into account large datasets, covering the region of
interest both in space and time. This allows not only the
description of traffic densities in space, but also mapping
seasonal patterns, cyclic routes and specific context related
behaviours. As a consequence, the information extraction
complexity requires the definition of ad hoc approaches in
order to represent the traffic in a robust and compressed
way. The cooperative nature of AIS allows the use of static
information almost uniquely identifying vessels and track-
ing them with no need for filtering algorithms. As a con-
sequence, AIS derived maritime traffic can be analysed in
more details if compared to other non-cooperative means
like radar, sonar, Earth Observation, etc.

Movement data are typically represented as temporal se-
quences of positions from two reference points (see e.g.
[14]). Given the large volume of AIS data, maritime traffic
data can be compressed into a list of waypoints. A subclass
of waypoints is represented by turning points, i.e. areas
where vessels’ COG change exceeds a pre-defined thresh-
old (see [12]). Port and offshore platform areas identify
another subclass of waypoints, where vessels show zero
Speed Over Ground (SOG). Typical in and/or out direc-
tions that connect direct predecessors and successors can
be identified for every waypoint. This means that the net-
work can be represented as a directed graph, i.e. waypoints
are connected only to a few others and a path between two
waypoints is not necessarily two-way. Moreover, the trajec-
tory between two nodes can be assumed rectilinear since
the motion of SOLAS vessels (i.e. class A AIS) can be mod-
eled as such, especially away from port areas. This brings
to the definition of sea lanes that are segments connecting
two waypoints, which can be either ports, turning points or
entering/exiting areas. Sea lanes are characterised also by
statistical properties observed directly from the data and re-
lated to the COG, SOG and spatial deviation from the seg-
ment. A sequence of sea lanes are here referred to as route
objects.

Algorithm 1 Unsupervised waypoint graph creation

Require: messages // set of AIS messages containing fields e.g.
//MMSI, COG, SOG, x, y, timestamp

Require: θ // steering angle (degrees)

Require: τ // time needed before labelling the track 'lost'

Require: VEs = ⊥

Require: EEs = ⊥

Require: TPs = ⊥

Require: POs = ⊥

Require: SLs = ⊥

Require: Rs = ⊥

for all message ∈ messages do
  // every ∆days, look for vessels that have not been updated
  // in the last τ interval and update the EEs list
  if mod(timestamp, ∆days) = 0 then
    for all V E ∈ VEs do
      if V E.last_update > τ then
        // the last recorded position of the vessel is used
        // to modify the EEs list: if an EE obj exists in
        //such position update it, otherwise create a new one
        EEs ← create/update(EEs, V E.(x, y), ‘out’)
        // the vessel list of waypoints is updated accordingly
        V E.waypoints ← add(EEs[V E.(x, y), ‘out’])
        // creation/update of both sea lane and
        // route objects (see Algorithm 2)
        manage(SLs, Rs, V E)
        // the vessel status is set as 'lost'
        V E.status ← ‘lost’
      end if
    end for
    // merge existing compatible objects (see Section 2.7)
    merge(TPs, POs, EEs, SLs, Rs)
  end if
end if

if V E[MMSI] then
  v ← V E[MMSI] // Vessel Of Interest
  // update an existing vessel object parameters
  v.params ← ⟨(x, y, COG, SOG,...),...
  ∆COG ← |v.COG−v.previous_COG|
  if v.SOG > 0 then
    if (∆COG > 0) then
      // the vessel is moving and direction changed
      TPs ← create/update(TPs,x,y,∆COG,...)
      // if a TP is either updated or created, the vessel
      // list of waypoints is updated accordingly
      v.waypoints ← add(TPs[x,y,∆COG,...])
      manage(SLs, Rs, v)
    end if
  else
    // a port or platform object is discovered/updated,
    POs ← create/update(POs,x,y,...)
    // update the vessel list of waypoints, SLs and Rs
    v.waypoints ← add(POs[x,y])
    manage(SLs, Rs, v)
  end if
else
  if the vessel does not exist, a new one is created
  V E ← add(V E[MMSI])
  V E[MMSI].params ← ⟨(x, y, COG, SOG,...),...
  // an entry point is then either created or updated
  EEs ← create/update(EEs,x,y,’in’)
  // the vessel list of waypoints is updated accordingly
  V E[MMSI].waypoints ← add(EEs[x,y,’in’])
  // the vessel status is initialised as 'alive'
  V E[MMSI].status ← ‘alive’
end if
end for

return VEs, TPs, POs, SLs, Rs, EEs

The problem of traffic data mining based on AIS informa-
tion can be effectively represented by object oriented pro-
rogramming, where vessel objects detect and reinforce (or penalise) waypoints, ultimately delineating sea lanes and subsequently routes. The waypoints graph creation process is summarised by Algorithm 1.

2.1 Class Vessel

Vessel objects (VE) are created or updated according to the information content of each decoded AIS message (or database record when performing historical data analysis). The class Vessel contains static and dynamic properties. While the former are related to the identification of the vessel (e.g., type, MMSI, call sign, name, International Maritime Organization (IMO) number, size), the latter information is related to state vector and route patterns and is progressively updated. Consistent changes in COG have been observed where traffic is regulated by routing systems such as traffic separation schemes, precautionary areas or areas to be avoided for safety reasons, deep water routes etc. As a consequence, once a COG change is detected, a method that creates or updates the corresponding i-th turning point $TP_i$ is defined (see Section 2.2 and Algorithm 1). Moreover, from the AIS data stream, the status of the vessel objects can be derived, which can be either 'alive' if recently updated or 'lost' in case the track has exited the area of interest and it has not recently been updated.

2.2 Class Turning Point

Turning Point is an area characterised by a high density of vessels changing COG. Turning points are here identified by a traffic density value, a polygon, and mass distribution of the vessels direction entering and exiting the area. The implemented $TP$ clustering algorithm is an on-line version of the $K$-means with specified maximum cluster size. AIS is characterized by a reporting rate that varies depending on the vessels kinematic properties. This means that AIS information can be updated at different refresh rates. Moreover, in areas with low coverage, it is possible to lose the AIS track and reacquire it when the vessel approaches the coverage of another receiver in the network. It is therefore necessary to monitor the COG change only when the vessel is persistently self-reporting in order not to upset its statistics.

2.3 Class Ports and Offshore Platforms

A special subclass of waypoints is represented by Ports and Offshore Platforms (PO). This class of objects is instanced by vessels having zero-velocity, and its centroid progressively shaped by other vessels following the same behaviour. Depending on the centroid distance from land of the cluster of vessels positions registered within these objects, they are classified either as ports or offshore platforms.

2.4 Class Entry or Exit Point

Entry or Exit Point (EE) objects need to be defined in order to describe the motion patterns within a confined area of interest. Whenever a vessel object leaves (enters) the area under analysis, an exit (entry) point is created/updated. Similarly to the other waypoints (Ports and Turning Points), EEs are described by a preferred entering or exiting direction, a list of transiting vessel objects, and a volume of traffic.

2.5 Class Sea Lane

A Sea Lane (SL) object is identified by a straight and oriented path that links two waypoints, either POs, EEs or $TP$s. The speed and course are relatively stable properties within a sea lane, and are statistically described in terms of probability density functions estimated from the data. Sea Lane objects are created by vessels spawned waypoints (see Algorithm 2). Subsequently, lanes are activated when the relevant traffic reaches a specified density of vessels.

2.6 Class Route

Route (R) objects are a series of connected SL objects linking two ports, an entry point and a port, a port and an exit point or an entry point and an exit point. Route objects can also be identified by a sequence of intermediate turning points. The series of nodes can therefore be represented by a string and similarities between patterns can be measured by approximate string matching techniques. For example, if a vessel on a routine path makes a deviation, the distance between current and usual routes can be measured. This allows to flag different levels of anomalies. Similarly to the other waypoints route objects are characterised by counts of registered vessels.

### Algorithm 2: Sea lanes and routes management

```plaintext
Require: v // vessel object under analysis
Require: EEs
Require: TPs
Require: POs
Require: SLs
Require: Rs

// a new waypoint has been added to the v.waypoint list;
// if the vessel has passed through at least two waypoints
if length(v.waypoints) > 1 then
    wps ← v.waypoints[end − 1 : end]
    SLs ← create/update(SLs, wps)
    // update the list of routes
    Rs ← create/update(Rs, SLs)
    // update the vessel object list of SLs and Rs
    v.patterns ← add(SLs, Rs)
end if

return Rs, SLs, v
```

2.7 Objects Creation Updating and Merging

The activity of waypoints PO, EE, TP, R and SL objects is incrementally learned from vessels traffic and described by a 7-day moving window list of daily number of distinct vessels. This description, designed in order to further capture temporal patterns and their space and time variability, is achieved through the add/update function called in Algorithm 1 and 2. The function applied to a generic class waypoint WP takes as inputs the list of class objects and
the relevant detection attributes \( X = (x, y, \Delta \text{COG}, \ldots) \), used to update the \( i \)-th object \( WP_i \):

\[
\text{argmin}_i \| X_{WP_i} - X \| \leq th
\]  

(1)

where \( th \) is a maximum distance threshold and \( X_{WP_i} \) the vector of object attributes. If (1) is not satisfied by any element of the \( WP_i \) list, then a new \( WP_{i[end + 1]} \) is created. In order to build the statistic object description, the running average of the waypoint new detection \( d_{i+1} \) is computed for the relevant \( n \)-th attribute \( X_{WP_i}(n) \), e.g. centroid, in/out directions, etc.:

\[
X_{WP_i}(n) \leftarrow \frac{X_{WP_i}(n) \cdot d_i + X(n)}{d_i + 1}
\]  

(2)

and the number of detections updated \( d_i \leftarrow d_i + 1 \). Waypoints with low activity (i.e. number of detections) are routinely deleted, while compatible objects are first detected:

\[
\text{argmin}_{k \neq j} \| X_{WP_k} - X_{WP_j} \| \leq th
\]  

(3)

and then merged into the one with highest number of detections. If \( d_k > d_j \) then:

\[
X_{WP_k}(n) \leftarrow \frac{X_{WP_k}(n) \cdot d_k + X_{WP_j}(n) \cdot d_j}{d_i + d_j}
\]  

(4)

The number of detections is also updated \( d_k \leftarrow d_k + d_j \) before \( WP_j \) is deleted from the \( WP_i \) list.

3 Results

The methodology is applied to data collected from i) coastal receivers over the Adriatic Sea and ii) Satellite AIS over the Red Sea and Gulf of Aden.

3.1 Adriatic Sea

The Adriatic Sea has been selected since well covered by AIS terrestrial receivers.

![Figure 1: Three months maritime traffic (blue) over the Adriatic Sea, areas to be avoided (red), traffic separation schemes (yellow) and precautionary areas (orange).](image1)

Figure 2: \( TP_{51} \) analysis: the cluster of detections is used to derive the object centroid, whereas the in/out directions are inferred from the entering and leaving COG distributions.

The region is a closed basin, therefore the maritime motion patterns can be consistently described also because of the large traffic. The methodology can be entirely validated by automatically learning the traffic patterns without \( a \ priori \) knowledge of the manifold routeing schemes that rule the navigation in area (see Figure 1), e.g. precautionary or restricted areas.

The cluster of detections in Figure 2 distributed around the turning point centroid shows consistency of directions entering and leaving the waypoint. When a new COG change is detected, the in/out COG distributions are checked before clustering the new detection and modifying the statistical properties of the turning point as generically described in Section 2.7. The methodology allows the superposition of waypoints in space since a further degree of in/out COG compatibility is checked. This is shown in Figure 3, where the complex system of routeing ruling the navigation over the North Adriatic Sea is consistently captured.

![Figure 3: Turning points automatically detected over the North Adriatic Sea.](image3)
Although the dataset covers a three-month period, after analysing the data belonging to a transient time of a few days, the detected waypoints stabilise leading to the nominal traffic representation.

3.2 Red Sea and Gulf of Aden

The approach has also been tested where the AIS information is highly intermittent as a result of gaps in coverage or between satellite overpasses. This allows measuring the consistency of the methodology in generating instances of waypoints under conditions where vessel objects are not persistently observed. To this aim, the area of Red Sea and Gulf of Aden has been analysed using Satellite AIS data. Gaps in the data ranging from a few seconds to up to tens of minutes can be observed depending on the AIS messaging update interval and lag between consecutive satellite overpasses. In Figure 4 the unsupervised detection of waypoints is illustrated. The International Recommended Transit Corridor (IRTC) is successfully mapped into a series of turning points and port areas from the data, demonstrating the learning capability of the implemented methodology also when using non-persistent data.

![Figure 4: Turning points (red) and ports (green) automatically detected over the Red Sea and the Gulf of Aden using one month satellite constellation AIS data.](image)

4 Conclusions

The paper has presented a framework aiming at automatically learning AIS maritime traffic patterns using an unsupervised approach that can work on real time systems. The proposed methodology proves effective both with data collected by terrestrial networks of AIS receivers and where the information is highly disrupted, e.g. as a result of spatial gaps in coverage or significant temporal intermittence due to the satellite revisit time.

The approach can be used as a basis to deliver real time low-likelihood behaviour detection using, for instance, Bayesian inference. Moreover, the tool can support other surveillance technologies, route planning and vessel position prediction.

References

[1] Safety of Life at Sea (SOLAS) convention Chapter V. Regulation 19.