

focused on port operations, Jia et al. (2017) uses port call detection for investigating port connectivity.

Another problem is that the data might be transmitted correctly, but for different reasons it is not received or processed properly and is notoriously error prone due to missing and unreliable manual entries. Additionally, missing context, such as port information, communication with other vessels or coastal authorities, environmental factors such as weather or depth of the visited water that might be relevant for a ships behavior are not part of the data set.

This makes prediction of vessel trajectories challenging. A common choice is using deep learning sequence prediction for AIS trajectories. The authors of Ibadurrahman et al. (2021) predict vessel trajectories in a sizable patch of the Indian Ocean consisting only of water area. On the other hand (Volkova et al., 2021) predict ship trajectories in a section of Norwegian river. Even more local analyses are berthing trajectory predictions (Wang et al., 2020). In between are Capobianco et al. (2021), who predict trajectories 3 h ahead in Kattegat, Xiao et al. (2023) predict up to 2 h trajectories at the US west coast, Liang et al. (2024a) use various data sets to predict up to 60 min. However, all of these approaches are using well-defined spatial and temporal boundaries which makes it difficult to apply on large areas. Recent work has also proposed the use of transformers for longer-term predictions on the scale of multiple hours (Yu et al., 2025).

Different similarity based approaches, which are better suited for less well defined data, are used by Cheng et al. (2019), Lu et al. (2020), Xiuju et al. (2021), Chen et al. (2023), Huang et al. (2023), Varlamis et al. (2021) for detecting shipping routes and analyzing maritime traffic. The authors of Cheng et al. (2019) focus on a specific group of routes for oil they term the maritime silk road between east Asia and Europe and Lu et al. (2020), Varlamis et al. (2021) compute route based networks within a given region. Several authors extract vessel routes using density-based clustering (Xiuju et al., 2021; Chen et al., 2023; Huang et al., 2023).

In addition to route-based networks, another commonly used approach is based on automatically computed traffic networks. Filipiak et al. (2020), Rong et al. (2020, 2021), Zhang et al. (2021), Prasad et al. (2021), Onyango et al. (2022), Lee et al. (2022) use automatic detection of waypoints as the basis for spanning a traffic network. Waypoints in this context refers to intersections of possible routes.

Another important research direction is maritime traffic anomaly detection (Riveiro et al., 2018; Ribeiro et al., 2023). Rong et al. (2020), Prasad et al. (2021) propose a traffic network based anomaly detection approach. However, traffic networks are also used to generate input for neural networks (Eljabu et al., 2021). Nguyen et al. (2020), Luo and Zeng (2023), Li et al. (2023) suggest the use of deep learning for modeling maritime behavior and anomaly detection. Liang et al. (2024b) propose a generative adversarial network based deep learning approach to detect anomalies. Xiuju et al. (2021) use the extracted routes for anomaly detection and route prediction.

The sheer amount of data that is produced and therefore needs to be processed makes a real time analysis challenging and heavily restricts the possible approaches when taking computational resources and wall time requirements into account. Real time analysis in this context means that the data needs to be computed as fast, or faster, than it is generated, so 24 h of data needs to be analyzed within 24 h on commercial server hardware.

To tackle these challenges, this work introduces Comprehensive Analytics for Predicting Traffic in mArItime Navigation (CAPTaiN). CAPTaiN is a comprehensive, end-to-end and fully data-driven approach solely based on AIS data for a real time automated predictive analytics system. It is designed to predict individual vessel routes, destinations and discover abnormal behavior without relying on manual intervention, expert knowledge, or additional data sets. All relevant components such as ports, port calls and the traffic network are directly calculated based on AIS transmissions. The resulting system provides continuous prediction and anomaly detection capabilities in an efficient

and transparent way. While previous approaches focus on narrow problem settings, our approach combines several relevant components for the analysis of maritime navigation on a large scale.

The approach is designed with a focus on transparency and efficiency in mind, which ensures explainable, traceable and replicable outputs when compared to more complex black-box models and at the same time allows for targeted modifications and future extensions.

This paper starts with a theoretical description of the developed methodology in Section 2 grouped by the different components. Following the same structure, the computational experiments and their implementation details are described in Section 3. Results and discussions of the experiments based on a hypothetical case study are described in Section 4 followed by concluding remarks about limitations and future work in Section 5.

2. Methods

The goal of CAPTaiN is a continuous predictive analytics monitoring system consisting of several components. Fig. 1 shows a structural overview. The components can be split into two groups, those that are computed in real time and those that are computed once in advance. For ease of reading, first a table of variables used in the subsequent sections as part of the methodology is provided.

Sets and Indices

D	Set of detected ports or destination regions
$D_n \in D$	Port or destination H3 hexagon set n
$G = (V, E)$	Directed maritime traffic graph with set of nodes V and set of edges E
$v_i, v_j, v_k \in V$	Graph nodes i, j, k
$e_{ij} \in E$	Directed edge from node v_i to node v_j
J	Set of extracted voyages
$V_v \in J$	Sequence of graph nodes visited during voyage v
$t \in \{0, \dots, V_v - 1\}$	Index of the currently observed step in a voyage

Vessel Prediction

$H^k \in \mathbb{N}^{ V \times D }$	Frequency matrix observed at node v_k
$H_{in}^k \in \mathbb{N}$	Number of times a vessel moved from v_k to v_i and ended in D_n
(v_j, v_i)	Transition from node v_i to node v_j
$P(v_i v_k)$	Probability of visiting node v_i next, given current node v_k
$P(D_n v_k)$	Probability of ending in destination D_n , given current node v_k
$P(D_n, (v_j, v_i))$	Joint probability of moving from v_i to v_j and ending in D_n
$P(D_n (v_j, v_i))$	Probability of destination D_n after observing transition (v_i, v_j)
$\hat{P}(D_n v_{t+1}, \dots, v_0)$	Interpolated probability of destination D_n after observing latest visited node v_{t+1} of visited node sequence starting in v_0
$P(v_{t+s})$	Predicted probability distribution over nodes at step t after simulation step s
$\text{top}k(\cdot)$	Operator returning the nodes with the largest predicted probabilities

Anomaly Detection

$\tau \in [0, 1]$	Node-specific threshold below which a transition is considered potentially anomalous
$p_l \in [0, 1]$	l -th largest outgoing-edge probability at the current node
$P_{sum} \in (0, 1)$	Cumulative probability parameter used to determine τ
$L \in \mathbb{N}$	Number of outgoing edges considered at the current node

the next step, they can be used to identify port call events, i.e., ships entering a port, which are the basis for automatically extracting voyages from the data.

2.4. Voyage extraction

In order to analyze and predict maritime traffic, ship movements are grouped into individual voyages. Accordingly, the voyage extraction is split into two distinct parts. First, port call recognition and, second, extraction of the relevant voyage data between two subsequent port calls for further use.

A straightforward definition of a port call is given by two properties: a ship is inside of a port, as defined by a polygon in the previous section, and its speed is below a small threshold for a minimum amount of time. The first condition is necessary because a ship cannot dock outside a port. On the other hand, the second condition is a way to mostly identify proper docking that is for example used to load cargo and not movement inside a port.

In more technical terms, a multivariate time series for each vessel is checked for matching these conditions. Every time both conditions are met at the same time, a port call event is created. In a second step, port events are merged if they happen in the same cluster without leaving in between. This procedure results in a sequence of port calls, including the port as well as the duration of the port call, for each ship.

While the port detection algorithm relied on navigational state to identify mooring ships, the port call detection explicitly does not rely on that data field, as it is a manually entered value, and therefore subject to human error. Rather, the port calls are detected solely based on position and speed, two automatically filled fields.

The voyages of each ship are extracted from the complete data set by looking up the data between two subsequent port calls. Additionally, the metadata of the voyage consisting of the starting port call, the ending port call, the duration of the journey and an estimate of the traveled distance are stored for later use.

The port call cluster IDs and duration are directly available from the port call events, the duration of the voyage is calculated as the time difference between the end of the first port call and the beginning of the second. The traveled distance is calculated as the sum of the euclidean distance between subsequent transmitted locations. This is only a lower bound of the travel distance because it presupposes the taken path between each point is the shortest possible, a straight line. However, with longer distances the likelihood of deviation from the shortest possible path increases as well.

The result is a list of all voyages extracted from the data and their metadata and the subset of AIS data associated with each voyage. This split of metadata and subsets of data allows a flexible selection of which voyages are relevant for further processing, for example when only voyages of a specific length are considered or a specific type of ship.

2.5. Graph extraction

The data of voyages and ports is used to model vessel movements to predict destination ports and immediate travel route as well as to identify ships with unusual behavior. The vessel trajectories, i.e., the sequence of discrete observed locations that define its movement during a voyage, require a robust model that compensates for the noisy data and many gaps. The model also needs to be efficient enough to enable continuous data analysis. Therefore, we construct a directed graph $G(V, E)$, with nodes $v \in V$ and edges $e \in E$. Each node v corresponds to a H3 hexagon.

The first step for building the graph is to narrow down the hexagon grid cells by selecting the subset of relevant hexagons. There are no restrictions like streets or rails on open water and ships are incentivized to travel efficiently for economic reasons. The shortest available path is assumed to be a sequence of arc segments. As long as the segment is not too long and the ships are not too far away from the equator,

this is approximately equivalent to a line in GPS coordinates. Hexagons useful for the modeling ship trajectories are therefore those hexagons, at which at least some ships change direction. This subset of hexagons is then used as nodes of the graph, while the remaining hexagons are ignored. The hexagons can be identified using a piecewise linear approximation (PLA) of the actual trajectory, where the connecting points of the PLA are the nodes in the graph. A similar approach to representing shipping routes is used by [Rong et al. \(2020\)](#).

One way to compute a PLA that allows for efficient segmentation of noisy trajectories is based on linear regression, similar to [Izadian et al. \(2020\)](#). In this approach a sliding window of w points is moved over the trajectory and for each window a regression line is fitted. According to [James et al. \(2013\)](#), the slope of this line can be computed as

$$m = \frac{\sum_{i=1}^n (x_i \cdot y_i) - w \cdot \bar{X} \cdot \bar{Y}}{\sum_{i=1}^n x_i^2 - w \cdot \bar{X}^2} \quad (1)$$

and the bias is given by

$$b = \bar{Y} - m \cdot \bar{X} \quad (2)$$

with $x_i \in X$ the latitude of the points in that window, $y_i \in Y$ the longitude of the points in that window and $\bar{X}, \bar{Y}$ denoting the means of X and Y . Afterwards the angle α between lines touching in a single point is calculated using the slopes m_1, m_2 as,

$$\alpha = \arctan \left(\left| \frac{m_1 - m_2}{1 + m_1 m_2} \right| \right) \quad (3)$$

resulting in a series of angles. The local maxima of that series are chosen as the breakpoints. Connecting these breakpoints results in the PLA of the original trajectory. In order to ensure the detected breakpoints are actually representing a change in direction, we introduce a lower bound for a local maximum to be considered as a breakpoint. Furthermore, there can only be one breakpoint per window, i.e. two breakpoints have to be at least w points apart.

[Fig. 3](#) shows an example of this procedure with the original trajectory shown as blue points, with the breakpoints shown in yellow and the approximated trajectory given by the connecting blue lines. The breakpoints are used as the hexagons of interest for building the graph.

The union of all breakpoints calculated and all ports serves as the set of nodes in the graph. To find the edges, each voyage is first mapped to the graph-based on the transmitted locations. At this stage each journey is represented as a sequence of visited nodes. The edges are calculated by introducing an edge $e_{ij} \in E$ between two nodes $v_i, v_j \in V$ exactly if a ship has visited v_j directly after visiting v_i without any other node in between.

The resulting graph represents the motion of the ships in an abstract way that aims to strike a balance between graph size and expressiveness by identifying those points in a voyage where new information is available, either by a ship changing or not changing direction. It is important to note, that each node corresponds to actually observed positions of ships from the historic AIS data, thus the sequence of locations gives a viable travel trajectory, even though the edges cannot be assumed to correspond to the actually chosen trajectory of the ship.

Additionally, this approach automatically corrects for signal coverage, since edge distance is not considered. Even if data from a specific area is spotty, the ships sequence of visited nodes is still consistent when the observed visited nodes are similar between journeys. The resulting graph does not represent the actually traveled trajectory but rather a sequence of traveled positions and represents a lower bound of the traveled distance, which aligns well with the unreliable nature of the AIS data and does not require additional assumptions since the data itself consists of a sequence of traveled positions.

It is however important to note, that the assumption of a straight traveled line does not hold for long lines for two reasons. As the ships are moving on a three dimensional sphere rather than a two dimensional plane, for longer distances this means the shortest path they travel is an arc rather than a straight line, but more importantly

Algorithm 2 Destination Prediction.

Require: V_v
Require: $P(v_j|v_i), P(D|v_j, v_i), P(D, v_i) \forall v_i, v_j \in V$
1: $P_D(D|v_0) \leftarrow P(D, v_0) \cdot 1$ $\triangleright P(v_0) = 1$
2: $v_{t-1} \leftarrow v_0$
3: **for** v_t in $V_v \setminus v_0$ **do**
4: $P_D \leftarrow P_D \cdot \frac{P(D|(v_t, v_{t-1}))}{P(v_t|v_{t-1}) \cdot P(D, v_{t-1}) + 1 \cdot 10^{-6}}$ $\triangleright$ Add small value to avoid division by 0
5: $P_D \leftarrow \frac{P_D}{\|P_D\|}$ $\triangleright$ Discrete distribution requires sum to be 1
6: $v_{t-1} \leftarrow v_t$
7: **end for**
8: **return** P_D

a database. These probabilities will also be used to predict the next visited nodes of a ship.

The iterative updates as presented above consider all nodes visited in a sequence equally important. In order to achieve a prediction that is more dependent on the later steps of the sequence, one possible approach is to introduce an interpolation update. Instead of using the updated probability directly, another adjustment is performed by calculating the average between the latest updated probabilities and the observed probabilities at the current node.

$$\hat{P}(D_n|v_{t+1}, \dots, v_0) = \frac{P(D_n|v_{t+1}, \dots, v_0) + P(D_n|v_{t+1})}{2} \quad (12)$$

In the following, the update calculating $\hat{P}(D_n|v_{t+1}, \dots, v_0)$ is referred to as *interpolation* while the iterative update is simply called *update*. This interpolation is independent of the iterative update. But if applied without the update, it does not change anything, as it calculates the mean of two times the same local probabilities.

2.7. Route prediction

After setting up the iterative procedure for the destination prediction, a natural extension is the estimation of the route that will most likely be traveled. In terms of the underlying graph representation, this is the prediction for the next nodes to be visited.

There are two additional assumptions relevant at this stage. First, it always takes one time step to use a single edge to go from one node to the next, independently of a ships velocity, the distance the edge represents, breaks at sea, actually traveled trajectory or other factors influencing travel time. Furthermore, most edges and most destinations are associated with a very small probability based on the data and can therefore be deemed irrelevant for prediction.

It is straightforward to estimate the next node from the destination prediction procedure, since the probabilities of the destinations as well as the probabilities of the next node given the destination are already calculated as part of the destination prediction.

Algorithm 3 Route Prediction.

Require: P_D $\triangleright$ Current distribution of destinations
Require: $P(v_j|v_i), P(v_j|D, v_i), P(D, v_i) \forall v_i, v_j \in V$
Require: $\text{sim_steps}, \text{sim_width}$
1: $P(v_{t+1}) \leftarrow P(v_{t+1}|D, v_t) \cdot P_D$
2: **for** s in $\{1, \dots, \text{sim_steps}\}$ **do**
3: $p_s \leftarrow 0$
4: **for** p_{vk}, v_k in $\text{topk}(P(v_{t+s}), \text{sim_width})$ **do**
5: $p_s \leftarrow p_s + p_{vk} \cdot P(v_{t+s+1}|D, v_k) \cdot P_D$
6: **end for**
7: $P(v_{t+n}) \leftarrow \frac{p_s}{\|p_s\|}$
8: **end for**

However, in general a n -step estimation for the next $n > 1$ steps can be achieved using a Monte-Carlo simulation based on the updated probabilities and local observations at the current node. For the purposes of the Monte-Carlo simulation, the destination probabilities are fixed at the current update step. In other words, the destination probabilities are based on the updated probabilities described in the previous subsection but do not change during the simulation stage.

Alg. 3 shows the procedure for estimating the next steps on the route of a ship, which is based on weighting the probability distribution of the subsequent nodes by the destination distribution. The prediction depends on two parameters, the `sim_steps` describe how many steps are predicted and the `sim_width` controls how many nodes per step are considered. Each step produces a distribution of nodes possibly to be visited at that step.

Since a higher `sim_width` means more nodes are taken into account during the simulation, the number of nodes with a non-zero predicted probability increases. However, as the probability sums to one, this means the individual node will be predicted with a smaller probability value. On the other hand, if the `sim_width` is small, there are only a few nodes that have a non-zero probability, which means the chance of the actual nodes not being part of the prediction increases.

When evaluating the route prediction, it is important to understand if the actually visited nodes are the same as the ones with the highest probabilities. This means the first evaluation metric is the accuracy, that is the percentage of predictions where the node(s) with the largest predicted probabilities contain the actually visited node.

Another useful metric is the distance between the actually visited node and the closest node in the predicted set. This metric is included, as the exact location of a ship is often not of interest, but rather the trajectory the ship is taking. Since this is not reflected in the accuracy if the predicted node is not exactly the same as the actual visited node, the distance is used to quantify the discrepancy between prediction and ground truth.

2.8. Detection of deviations from expected behavior

The aim in the previous subsections is to model the expected, normal behavior as closely as possible. For some applications it is of interest to identify cases where ships do not behave as expected. In this subsection, an approach utilizing the methods described in previous sections for identifying vessel behavior that might warrant further investigation, i.e., a set of candidate anomalies for manual verification. Importantly, this does not represent a one-stop anomaly detection solution. Rather it is a support for automatically identifying vessels that behave in unexpected ways and therefore selecting a subset of vessels that might be of interest for monitoring purposes.

At every node, there is a distribution for the following nodes that can be used to detect potential anomalies. Whenever a vessel uses an unlikely edge, i.e., is observed to arrive at a node that was predicted to be very unlikely, this is considered an anomaly. By *unlikely* we mean that the predicted probability of the event is smaller than a threshold probability τ . Every event with a smaller probability than τ is considered an anomaly. We set τ using the probabilities p_l , sorted in descending order, of using an outgoing edge at the current node. The value of τ is set to

$$\tau = p_{l^*} \quad \text{where} \quad l^* = \min \left\{ l \in \{0, 1, \dots, L\} \left| \sum_{l'=0}^l p_{l'} > P_{\text{sum}} \right. \right\} \quad (13)$$

with $P_{\text{sum}} \in (0, 1)$ a hyperparameter that is used to tune how many events are considered anomalies and L the total number of outgoing edges at the current node. At each transition it is checked, whether the associated probability of that transition is smaller than τ .

Due to the dynamic nature, this does not result in a global threshold, but different thresholds for different nodes and different voyages, highly dependent on how the observed ships in the training set behave. As an extreme example, consider a node which all ships leave to the

identical next node. The associated probability is 100%, so every transition out of that node with less than 100% expectation is considered unexpected and therefore suspect. In this case, the expectation reflects this as well, and the probability for using a different edge than the one all the other ships used is going to be very low.

The opposite can be true for a hypothetical node with 10,000 uniformly distributed next nodes. The associated probabilities for subsequent node resulting in 0.0001, so only transitions with less than 0.0001 are considered unexpected and therefore suspect. This approach results in a transparent identification of behavior deviation, which can subsequently be independently verified.

The value of τ can be calculated at two different time points. One option is to calculate the threshold in advance, which results in a static threshold for each node. Therefore it does not take dynamic probability updates into account.

Alternatively τ can be computed dynamically at each node when a ship visits it. In this dynamic case, τ is calculated with the edge probabilities modified according to the current estimation of the destination probabilities. This allows for a better estimate of which edge is likely and which is unlikely. However, it is computationally more expensive, as it takes the updated probabilities into account and therefore can only be computed in real time.

Independently of how the threshold probability has been arrived at, each observed node transition can then be analyzed by comparing the probability associated with the observed node transition to τ during the voyage immediately after the latest observation is observed, i.e. as part of the prediction loop if multiple transitions are computed directly in sequence.

Following the assumption that common behavior reflects the motivation of the vast majority of ships acting according to a limited set of motivations, e.g. following a pre-planned route, choosing the most economic or reliable trajectory, ... and any deviations from these behavioral patterns reflect potentially interesting events, depending on the application, but interpreting these is difficult based on AIS data alone and in practice requires additional information or expert knowledge.

The identified node transitions are therefore best considered as anomaly *candidates*, as validation is impossible due to no available data set of anomalies. The identified deviations from expected behavior are however compared to transitions that are expected empirically by comparing the edge structure.

3. Experiments

After describing the ideas behind each part of the analysis in the methods section, the details relevant for implementation are described. The aim of this approach is to create a system that is able to provide *continuous predictive analysis*, which means computational expense is a major factor that needs to be kept in mind. This section is structured similarly to the *Methods* Section 2 and presupposes the ideas described there.

Our experiments are aligned with a case study of a fictive maritime safety organization that is interested in monitoring the Mediterranean Sea and Black Sea. For this case study, we assume that the main goal is to use AIS data arriving in batches in regular intervals to automatically identify unexpected or suspicious events that happened in the last 24 h and improve capabilities to prepare for the next 24 h. The overall parameter choices in the context of this study are set as a reasonable trade-off between resource requirements, computation speed and precision of results, given the amount of data we processed in these experiments and the available hardware.

3.1. Hardware and software stack

The experiments were carried out on a machine with two AMD Epyc 7763 CPUs, each having 64 cores and 128 threads with a maximum clock speed of 3.6 GHz and 1024 GB of shared RAM. There are three NVIDIA A40 GPUs with 48 GB of GDDR6 memory available. The code was written in Python 3.10, with pandas 2.1 ([pandas development team, 2020](#)), numpy 1.26 ([Harris et al., 2020](#)) and H3geo 3.7 ([Uber Technologies, Inc., 2021](#)) as the core packages. PyTorch 2.10 ([Paszke et al., 2017](#)) is used for GPU computation. The data was processed in memory.

3.2. Data and data processing

The analysis was developed using AIS data of the year 2018 of the Mediterranean and Black Seas, with the data set containing more than 2,700,000,000 AIS messages before preprocessing. Evaluation uses data of 2020, with more than 3,800,000,000 AIS messages. The AIS standard calls for different frequency of reports, depending on ship type and status, ranging from 2 s to 10 min ([U.S. Coast Guard, 2026](#)). After preprocessing, data points are associated to a 30 min time window, representing a ships position during that interval. Notably, we do not use databases, e.g. lists of ports and port details or waterways but rather are deducting everything from AIS messages.

In comparison to other data sets, the considered data is based on data points in the Mediterranean and Black Seas but does not make any assumptions on the ships involved, the vessel behavior, trade routes ([Cheng et al., 2019](#); [Ibadurrahman et al., 2021](#)) or vessel and cargo types ([Gouareb et al., 2022](#); [Magnussen et al., 2021](#)), or specific ports or local areas ([Zhang et al., 2021](#); [Liu et al., 2023](#); [Li et al., 2023](#)). We observe 854 Ports, and 45,716 graph nodes and over 250,000 voyages, in a large complex area and do not enforce the voyages to have no gaps or filter by specific behavior.

As some parts of this approach depend on unique identifiers of ships, but MMSI is a manually entered field, we took some steps to obtain a cleaner list of relevant ships. First of all, this means eliminating likely incorrect MMSIs such as 100000000 and lower, or 999999999, and MMSIs which seem to be used by multiple ships. However, to the best of our knowledge, there is no definite way of compiling a list of accurate or inaccurate MMSIs.

3.3. Mooring extraction

The idea of clustering mooring ships that do not move leads to a few requirements for the data. As already described in Section 2.3 only messages that specify navigational status 5 are considered, which means ships at mooring. Additionally the vessel types are restricted to *Passenger ship* (AIS ShipTypes 60–69), *Cargo ship* (70–79), *Tanker* (80–89), *Sailing ship* (36), *Pleasure craft* (37) and *High speed craft* (40–49) at this processing step. The remaining data points are then grouped by MMSI and PositionTime into 30 min windows for each ship. Within each group the speed and position are aggregated by calculating the mean. This value is then used to remove ships that are moving faster than 0.1 knots on average, allowing for small movements due to water movement or transmission errors, but otherwise restricting the data set to non-moving vessels. The speed over ground of a vessel is given in 0.1 kn steps, so the choice of 0.1 represents a small tolerance for ship movements or positioning inaccuracies leading to slight change in transmitted GPS coordinates, thereby reducing the number of erroneously detected voyages.

The locations are then converted to their respective H3 hexagons, with the center coordinates as a proxy for the hexagon location when coordinates are required. Converting to H3 hexagons allows to further reduce the data, since working on a grid means all the points within a single cell are considered equal. As the data set is processed in order to be clustered using DBSCAN, each hexagon at most needs to occur

as many times as the DBSCAN parameter min_pts , which describes the minimum cardinality of a group of points to be considered a cluster and not outliers. In this case, this parameter was set to 20, which means at least 20 occurrences of a ship mooring for 30 min are required for a cluster over the course of a year.

Furthermore the coastline filter is applied. This filter is implemented based on a H3.10 (H3 resolution 10) hexagons, meaning H3 resolution 10. In a first step, the coastline polygons found in the *SeaVox v18* data set (British Oceanographic Data Centre (BODC), United Kingdom, 2021) are converted into a set of H3 hexagons. Based on this initial set, a radius of 62 hexagons, or 8 km is added around each hexagon and the union over all of them is calculated. This provides a set of hexagons which includes all coastal areas and therefore any point used for clustering must be inside of one of these hexagons. The radius of 62 is chosen to achieve an area of around eight kilometers around the coast, which is far enough to include docks that reach into the sea but close enough to filter out waiting areas in international waters. Applying this filter only requires to compute the overlap between hexagons associated with ports and the set of coastline hexagons. It is important to mention, that the coastline hexagon set does not change and therefore can be calculated in advance.

Whether this filter is applied, or applied in this form, depends on the interest regarding waiting areas and anchorages. It is conceivably useful for some applications to not only focus on actual ports, but also large waiting areas, as traffic might be planned with an anchorage in mind and not a port if the destination is not perfectly clear or a waiting period is expected.

At this point the data set consists of points representing a vessel that has docked for 30 min. These points are then clustered based on their H3.8 hexagon centers. In order to set the hyperparameters, we ran the clustering with different configurations (see Table 1) and selected a configuration in agreement with a subject matter expert. The clustering approach as described in Alg. 1 is executed with $eps1 = 0.5$, $eps2 = 0.1$, $min_samples = 20$ and $recluster_threshold = 0.25$. These parameters are chosen to keep distinct ports separate, especially after the second stage of clustering, but allow for spread out ports with distant docks to still be identified reliably. At the same time, the $min_samples$ did not impact the results much beyond a value of 10–15, as at this point the locations where ships were only observed very rarely, and for a short amount of time, so they are deemed not relevant for the purposes of this work.

The set of parameters applied in this case is selected to align with the requirements of the case study. That means, the identified clusters capture the relevant ports in the Mediterranean region and correctly separate distinct ports into two clusters. We accept that small, rarely used ports are mistakenly ignored as outliers, but those cases are of smaller interest in this case of modeling maritime traffic. A port that is used infrequently enough to not clear the threshold in this methodology is assumed to be irrelevant for the overall traffic network. In order to identify even those remote, infrequently used ports, the parameters can be tuned by setting a smaller number of $min_samples$ or higher max speed to achieve a more sensitive detection, keeping more smaller ports in the data set. However, if one is interested in the major traffic only, a higher number of $min_samples$ or a larger allowed port area might be a better choice, as this eliminates smaller ports and potentially irrelevant distinctions between close ports.

3.4. Voyage extraction

The next step is using the identified ports for automatically detecting port calls and subsequently voyages. First, to identify port calls the data is split by ship via the MMSI and sorted by time. This results in a sequence of aggregated data points as described in Section 3.2. For each data point we compute the H3 hexagon of the same resolution that was used for the ports. These hexagons can then be utilized to assign clusters to the data points. Every time a vessel stays in a single cluster

Table 1

Hyperparameter values for the cluster algorithms used for mooring extraction.

Parameter	Values
eps1	0.008, 0.01, 0.02, 0.05
eps2	0.005, 0.008, 0.01, 0.02
recluster threshold	10, 25, 50
2nd cluster algorithm	DBSCAN, Agglomerative Clustering
coastline filtering	True, False

for at least 120 min and moves at most 0.1 kn on average during that interval, a port call event is created.

These values are chosen based on experiments with different cluster configurations and different threshold values. Fig. 4 shows the percentage of extremely short voyages for different clustering results and different minimum port call duration and speed threshold. Overly sensitive values (e.g. < 0.1 , at least 30 min) lead to a large number of voyages that are only short times of movement during a single port call.

The final values are chosen based on feedback from an expert with knowledge of the application, in the form of 120 min minimum duration at port, as usually loading and unloading requires some time, even in the case of ferries, and less than 0.2 average speed over ground, as some water movement or measuring errors can lead to some reported speed, and the smaller threshold produces too many short voyages while the larger values are more sensitive to changes in clustering configuration. These resulting port call events are saved in the form of a separate list containing the ship, the port, as well as starting and ending time of the event.

This setup allows to filter the voyages that need to be analyzed separately after the extraction stage. In this case, the data was used to separate only those voyages that have a minimum length of ten kilometers, as traced by the trajectory of visited H3 hexagon centers and only the ship types *Cargo ship*, *Passenger ship*, *Tanker*, *Fishing vessel*, *Rescue vessel*, *High speed craft*, *Pleasure craft*, *Sailing ship* and *Ship* were considered. Furthermore, the distinction between including and excluding voyages that started and ended in the same cluster is made. As vessel type is a manually entered field, it is unclear how accurate those vessel type claims actually are and how many possibly relevant vessels are missed due to the crew transmitting a different vessel type. However, some vessel types exhibit drastically different behavior, e.g. patrolling law enforcement or port operating pushers or tugs.

Based on these data sets respective graphs are built as described in the next subsection.

3.5. Graph extraction and mapping vessel movement

In order to build the graph, H3.6 hexagons are used as nodes. The relevant hexagons are identified using a PLA of the original trajectories. This procedure can be implemented in a highly parallel way as described in Section 2.5. For the PLA a window size of 10 was used and for the peak detection of the series of angles only local maxima of above 0.2 radians are considered.

Similarly to the AIS transmissions, the movement of each vessel is a sequence of visited positions, in this case nodes on the graph, while the actually traveled trajectory remains unknown. The edges therefore connect two subsequently visited nodes. The edges are therefore added by first mapping the observed vessel locations from the AIS data and then adding an edge between two nodes whenever a ship visits them directly in succession. While a ship can visit the same node multiple times within a single voyage, self loops are not permitted, i.e., no edge e_{ii} from v_i to v_i . This guarantees, that each vessel moves to a new node at every time step.

The resulting graph strikes a good balance between mapping the vessel movements, with enough data per node to be useful, while keeping the size of the graph manageable for further applications in the context of the available hardware and computational requirements.

5.1. Limitations

The presented methodology combines several data-driven approaches in a larger processing pipeline. The results are however depending on the quality and amount of data included in the pipeline. This means first of all, that a limitation of this methodology is the required data itself, and the quality and meaningfulness of the results are depending on it. At various steps, it is necessary to have a large enough sample of data. The port identification, as an example, is based on the assumption that ports have several ships that accurately report their state while mooring, and it is sufficient to use only ships that do accurately report their state as opposed to relying on more complex models, such as sequence classification to identify mooring trajectories of ships in port. This works for larger data sets and especially for larger, more frequented ports but might be unreliable if smaller ports with less observations are of interest.

A second aspect of this is that the different employed methods themselves are based on the intermediate results of previous steps. This is both true for the different pipeline steps, e.g., the reliability of the detection of deviation from expected behavior is presupposing an accurate model of expected behavior and the voyage extraction approach is built on the expectation that the relevant ports are accurately identified.

As AIS data trajectories are often incomplete, a main goal of the proposed approach is that the approach works well even if gaps are present and the reported positions are not equally spaced. In this spirit, iteratively updating the predictions every time there is new relevant information provides the most up-to-date estimation without relying on trying to fill gaps, but it also has the drawback of predictions not being updated if there is no new information and therefore the reliability of predictions decreases with longer gaps as uncertainty about the position of a ship increases. Especially in cases with longer gaps, there might be an interval of hours or days without any updates, which means the predictions for that voyage are likely outdated and can be imprecise. Even worse, in cases where a gap contains an event like a port call, the trajectory expectation can subsequently be off.

Methodologically, the iterative approach is designed to continuously run and incorporate new information. This includes cases, where there is a pattern of gaps, e.g., in cases where receiver coverage in an area is inadequate, where the graph representation allows the expectation of transition using longer edges, which are bridging those gaps. However, it is unclear if there are more complex patterns, that are not accurately captured by the chosen models and can be more adequately captured with large deep learning approaches like transformers or if the accuracy can be improved with data imputation to reduce gaps, as those approaches were not tested.

The data-driven aspect also leads to limitations as a result of the utilized data. This work focuses on decision support solely based on one data source: AIS. However, while this approach makes the methodology and results more transparent, only using AIS data has some downsides as well. First of all, there are some data quality issues that cannot be resolved. It is for example unclear, whether an area has a broken receiver or if there is a different reason for gaps in an area at that time. Similarly, if data points for a ship are missing, we cannot compensate by filling in gaps based on other data sources.

Furthermore, the analyzed data set is limited to the Mediterranean and Black Seas, and only 2018 was used for estimating the traffic behavior. Therefore, seasonality effects or trends over time and very rare events are not modeled well. Additionally we do not consider voyages, that are longer in both time and distance than those found in this data set, like trade routes from China to Hamburg or Amsterdam. This choice of region also affects this limitation, as seasonality effects in this region are much less relevant than, e.g. in the Arctic region with the Northern Sea Route where not all parts are accessible year-round or extreme weather events impacting maritime traffic in the Gulf of Mexico, Atlantic Ocean or Pacific Ocean (Izaguirre et al., 2011; Cardone et al., 2015).

The methodological choice of only relying on a single data source is especially relevant regarding the anomaly detection part. Since we do not consider other aspects such as weather or geopolitical situations, as they are not part of the AIS messages, there can conceivably be periods with many anomalies if, e.g. a frequently used passage is blocked, or sanctions are affecting an important port.

The simplification of not considering travel time or distance but only the sequence of visited nodes on the graph means that there is no estimate of a voyage duration, which can be helpful in some use cases. Similarly, we do not differentiate between different ship types or vessel properties, which potentially might be used to improve some predictions. For both cases the improvement on the predictions remains unclear, and for reasons of data availability and computational requirements, this was not part of this study.

5.2. Future work

The main part of this study demonstrates, how AIS reports can be utilized for a complete pipeline that allows monitoring, analyzing and predicting maritime traffic on a large scale. The approach presented here can be extended in several ways. First, in the case of deploying a system based on the proposed methods for continuous monitoring, updates of the probability distributions and the extracted graph-based on observed vessel behavior might be required for sustained accuracy. This calls for more work on dynamically modifying the underlying graph and probability estimates. At the same time, the training can be rerun regularly to add new ports and graph nodes, especially if the observed routes are continuously saved, since the graph can be built within minutes given the observed ports and vessel trajectories and the port recognition can be run on the scale of minutes as well.

Furthermore, the predictions can possibly be further refined when taking additional information into account. We assume more accurate predictions could be reached if more data or computing power is available, taking regular behavior into account or properties of ships and ports might yield more precise predictions.

Another avenue of investigation that might yield useful results is the estimation of travel time to the predictions as well as to the edges to achieve a more comprehensive analysis of not only where a ship is at the moment and where it will be in the future, but also when it will be there.

CAPTain is designed to incorporate other predictions, as long as they produce a distribution over the destinations or next visited nodes. Currently, the iterative updating procedure is based on the previously visited nodes, but similarly, a different probability estimate can be used to include additional evidence. When staying within the confines of AIS data, one example is weighting the destinations based on the vessel type or dimensions. Furthermore, at this point there could also be other models that make their own predictions of future trajectory and eventual destination. It could therefore be fruitful to investigate other AI approaches, such as Deep Learning or Gradient Boosting, and integrating those predictions with those of CAPTain, creating an ensemble-style prediction.

Similar to adding components to the predictions using only the AIS data, the same idea can be used to incorporate other data sources, or models that combine AIS data with external data to further refine the predicted distributions of the immediate trajectory and destination ports. However, this requires not only alignment of different data sources but may also significantly change the resource requirements and underlying assumptions.

When scaling up the presented method, new problems might arise, but also new opportunities can open up. In this work, we only use H3 hexagons of the same size, but H3 explicitly allows for combining different hexagon sizes. Especially when scaling up to a larger covered area, using a mix of larger hexagons in some areas and smaller ones near coastlines or busy areas might allow for further efficiency improvements.

Another advantage of using the H3 grid is that the nodes in the resulting graph are more robust towards the effects of GPS-coordinates distortion when moving away from the equator. The area each node represents should be reasonably similar, even when moving close to the poles, thereby facilitating application to those areas without requiring extensive modifications.

While this study only considers the Mediterranean and Black Seas, the system can also be used to include larger, or different, parts of the globe. In that case, new challenges arise that need to be investigated. It is reasonable to expect the different sizes of Hexagons to be more useful when analyzing large areas of oceans as well as complex coastlines or bottlenecks. It needs to be investigated how the methodology holds up as the graph size and sequence length increase and the underlying network becomes even more complex.

In general, the methodology provided here, as it does not make assumptions regarding vessel or port type, should generalize well to other applications, but there might be differences in vessel behavior, such as some vessel types mostly using small, even private, marinas rather than large commercial ports. With increasing size, there are assumptions that might no longer be adequate, such as the idea of ships going in approximately straight lines on open water if the distances become too large and thus the linear regression based approach requires to first project the coordinates to a 2D plane to correct distortional effects.

It might also be useful to partition the graph into smaller segments, which are connected by pseudo ports at bottleneck locations, such as the Panama Canal or the Strait of Gibraltar. Segmenting the graph overall can align well with the actual traffic patterns, as a ship might be going towards a bottleneck, which is easy to identified compared to understanding which port it targets afterward. At the same time, this segmentation keeps the sequences limited and therefore more predictable and requires smaller graphs and therefore less computational resources.

On a different note, it can also be interesting to take a closer look at analyzing the underlying graph itself to learn more about the properties of the maritime traffic network. A traffic analysis based on the extracted network can be helpful not only for academic curiosity but also in order to better understand the required infrastructure or traffic patterns and improve the relevant rules and regulations in a data-driven way.

CRediT authorship contribution statement

Matthias Schilling: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Maximilian Moll:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Conceptualization. **Stefan Pickl:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Declaration of Generative AI and AI-assisted technologies in the writing process

ChatGPT Edu was used to create a draft of the notation overview, otherwise generative AI was not used to support the writing process.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Matthias Schilling reports financial support was provided by University of the Bundeswehr Munich. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors acknowledge financial support by the University of the Bundeswehr Munich. We would like to thank Rudy Milani, Sascha-Andre Schuster and Rohan Saxena for support in proof reading this article and Philipp Triebold for support with the graphical abstract.

Data availability

Data cannot be shared due to Licensing.

References

- Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- British Oceanographic Data Centre (BODC), United Kingdom, 2021. Polygon data set of the extent of water bodies from the SeaVoX salt and fresh water body gazetteer (v18). <http://dx.doi.org/10.14284/540>.
- Capobianco, S., Millefiori, L.M., Forti, N., Braca, P., Willett, P., 2021. Deep learning methods for vessel trajectory prediction based on recurrent neural networks. *IEEE Trans. Aerosp. Electron. Syst.* 57 (6), 4329–4346. <http://dx.doi.org/10.1109/TAES.2021.3096873>.
- Cardone, V.J., Callahan, B.T., Chen, H., Cox, A.T., Morrone, M.A., Swail, V.R., 2015. Global distribution and risk to shipping of very extreme sea states (VESS): Global distribution and risk to shipping of very extreme sea states. *Int. J. Clim.* 35 (1), 69–84. <http://dx.doi.org/10.1002/joc.3963>.
- Chen, J., Chen, H., Chen, Q., Song, X., Wang, H., 2023. Vessel sailing route extraction and analysis from satellite-based AIS data using density clustering and probability algorithms. *Ocean Eng.* 280 (114627), 114627. <http://dx.doi.org/10.1016/j.oceaneng.2023.114627>.
- Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, <http://dx.doi.org/10.1145/2939672.2939785>.
- Cheng, L., Yan, Z., Xiao, Y., Chen, Y., Zhang, F., Li, M., 2019. Using big data to track marine oil transportation along the 21st-century maritime silk road. *Sci. China Technol. Sci.* 62 (4), 677–686. <http://dx.doi.org/10.1007/s11431-018-9335-1>.
- Elayam, M.M., Kerhoas, G., Cursay, V.L.d., Ray, C., Ménard, A., 2022. On the interest of hexagonal abstraction of maritime information. In: *OCEANS 2022, Hampton Roads*. <http://dx.doi.org/10.1109/OCEANS47191.2022.9977062>.
- Eljabu, L., Etemad, M., Matwin, S., 2021. Anomaly detection in maritime domain based on spatio-temporal analysis of AIS data using graph neural networks. In: *2021 5th International Conference on Vision, Image and Signal Processing. ICVISIP*, <http://dx.doi.org/10.1109/ICVISIP54630.2021.00033>.
- Ester, M., Kriegel, H., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD* 226–231.
- Filipiak, D., Węcel, K., Stróżyna, M., Michalak, M., Abramowicz, W., 2020. Extracting maritime traffic networks from AIS data using evolutionary algorithm. *Bus. Inf. Syst. Eng.* 62 (5), 435–450. <http://dx.doi.org/10.1007/s12599-020-00661-0>.
- Gouareb, R., Can, F., Ferdowsi, S., Teodoro, D., 2022. Vessel destination prediction using a graph-based machine learning model. In: *Network Science*. In: *Lecture notes in computer science*, Springer International Publishing, Cham, pp. 80–93. http://dx.doi.org/10.1007/978-3-030-97240-0_7.
- Graves, A., Mohamed, A.-R., Hinton, G., 2013. Speech recognition with deep recurrent neural networks. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, <http://dx.doi.org/10.1109/icassp.2013.6638947>.
- Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. *Nature* 585 (7825), 357–362. <http://dx.doi.org/10.1038/s41586-020-2649-2>, URL <https://doi.org/10.1038/s41586-020-2649-2>.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780. <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.
- Huang, C., Qi, X., Zheng, J., Zhu, R., Shen, J., 2023. A maritime traffic route extraction method based on density-based spatial clustering of applications with noise for multi-dimensional data. *Ocean Eng.* 268 (113036), 113036. <http://dx.doi.org/10.1016/j.oceaneng.2022.113036>.
- Ibadurrahman, Hamada, K., Wada, Y., Nanao, J., Watanabe, D., Majima, T., 2021. Long-term ship position prediction using automatic identification system (AIS) data and end-to-end deep learning. *Sensors* 21 (21), <http://dx.doi.org/10.3390/s21217169>.

- Iphar, C., Le Berre, I., Foulquier, É., Napoli, A., 2024. Port call extraction from vessel location data for characterising harbour traffic. *Ocean Eng.* 293 (116771), 116771. <http://dx.doi.org/10.1016/j.oceaneng.2024.116771>.
- Izaguirre, C., Méndez, F.J., Menéndez, M., Losada, I.J., 2011. Global extreme wave height variability based on satellite data: WORLDWIDE EXTREME WAVE HEIGHT. *Geophys. Res. Lett.* 38 (10), <http://dx.doi.org/10.1029/2011gl047302>.
- Izakian, Z., Mesgari, M.S., Weibel, R., 2020. A feature extraction based trajectory segmentation approach based on multiple movement parameters. *Eng. Appl. Artif. Intell.* 88 (103394), 103394. <http://dx.doi.org/10.1016/j.engappai.2019.103394>.
- James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. *An introduction to statistical learning*, 2013 ed. Springer Texts in Statistics, Springer, New York, NY.
- Jia, H., Daas Lampe, O., Solteszova, V., Stranden, S.P., 2017. Norwegian port connectivity and its policy implications. *Marit. Policy Manag.* 44 (8), 956–966. <http://dx.doi.org/10.1080/03088839.2017.1366080>.
- Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. <http://dx.doi.org/10.48550/arXiv.1412.6980>, ArXiv [Cs.LG].
- Lee, J.-S., Lee, H.-T., Cho, I.-S., 2022. Maritime traffic route detection framework based on statistical density analysis from AIS data using a clustering algorithm. *IEEE Access* 10, 23355–23366. <http://dx.doi.org/10.1109/access.2022.3154363>.
- Li, H., Jiao, H., Yang, Z., 2023. AIS data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods. *Transp. Res. Part E: Logist. Trans. Rev.* 175, 103152. <http://dx.doi.org/10.1016/j.tre.2023.103152>.
- Liang, M., Su, J., Liu, R.W., Lam, J.S.L., 2024a. AISClean: AIS data-driven vessel trajectory reconstruction under uncertain conditions. *Ocean Eng.* 306 (117987), 117987. <http://dx.doi.org/10.1016/j.oceaneng.2024.117987>.
- Liang, M., Weng, L., Gao, R., Li, Y., Du, L., 2024b. Unsupervised maritime anomaly detection for intelligent situational awareness using AIS data. *Knowl. Based Syst.* 284 (111313), 111313. <http://dx.doi.org/10.1016/j.knsys.2023.111313>.
- Liu, D., Rong, H., Guedes Soares, C., 2023. Shipping route modelling of AIS maritime traffic data at the approach to ports. *Ocean Eng.* 289 (115868), 115868. <http://dx.doi.org/10.1016/j.oceaneng.2023.115868>.
- Lu, N., Liang, M., Zheng, R., Liu, R.W., 2020. Historical AIS data-driven unsupervised automatic extraction of directional maritime traffic networks. In: 2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics. ICCCBDA, IEEE, pp. 7–12. <http://dx.doi.org/10.1109/icccbda49378.2020.9095596>.
- Luo, S., Zeng, W., 2023. Vessel trajectory similarity computation based on heterogeneous graph neural network. *J. Mar. Sci. Eng.* 11 (7), 1318. <http://dx.doi.org/10.3390/jmse11071318>.
- Magnussen, B.B., Bläser, N., Jensen, R.M., Ylänen, K., 2021. Destination prediction of oil tankers using graph abstractions and recurrent neural networks. In: *Lecture Notes in Computer Science*. In: Lecture notes in computer science, Springer International Publishing, Cham, pp. 51–65. http://dx.doi.org/10.1007/978-3-030-87672-2_4.
- Meyers, S.D., Azevedo, L., Luther, M.E., 2021. A scopus-based bibliometric study of maritime research involving the automatic identification system. *Transp. Res. Interdiscip. Perspect.* 10 (100387), 100387. <http://dx.doi.org/10.1016/j.trip.2021.100387>.
- do Nascimento, V.D., Alves, T.A.O., de Farias, C.M., Dutra, D.L.C., 2024. A hybrid framework for maritime surveillance: Detecting illegal activities through vessel behaviors and expert rules fusion. *Sensors (Basel)* 24 (17), 5623. <http://dx.doi.org/10.3390/s24175623>.
- Nguyen, D., Simonin, M., Hajdich, G., Vadaine, R., Tedeschi, C., Fablet, R., 2020. Detection of abnormal vessel behaviours from AIS data using GeoTrackNet: from the laboratory to the ocean. In: 2020 21st IEEE International Conference on Mobile Data Management. MDM, pp. 264–268. <http://dx.doi.org/10.1109/MDM48529.2020.00061>.
- Oliveira, S., 2022. Data extracts. (Accessed 8 May 2025), <https://documentation.spire.com/data-extracts/>.
- Onyango, S.O., Owiredo, S.A., Kim, K.-I., Yoo, S.-L., 2022. A quasi-intelligent maritime route extraction from AIS data. *Sensors (Basel)* 22 (22), 8639. <http://dx.doi.org/10.3390/s22228639>.
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch. In: *NIPS-W*.
- Prasad, P., Vatsal, V., Roy Chowdhury, R., 2021. Maritime vessel route extraction and automatic information system (AIS) spoofing detection. In: 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies. ICAECT, IEEE, pp. 1–11. <http://dx.doi.org/10.1109/icaect49130.2021.9392536>.
- Ribeiro, C.V., Paes, A., Oliveira, D.d., 2023. AIS-based maritime anomaly traffic detection: A review. *Expert Syst. Appl.* 231, 120561. <http://dx.doi.org/10.1016/j.eswa.2023.120561>.
- Riveiro, M., Pallotta, G., Vespe, M., 2018. Maritime anomaly detection: A review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 8 (5), e1266. <http://dx.doi.org/10.1002/widm.1266>.
- Rong, H., Teixeira, A.P., Guedes Soares, C., 2020. Data mining approach to shipping route characterization and anomaly detection based on AIS data. *Ocean Eng.* 198 (106936), 106936. <http://dx.doi.org/10.1016/j.oceaneng.2020.106936>.
- Rong, H., Teixeira, A.P., Soares, C.G., 2021. Maritime traffic network extraction and application based on AIS data. In: 2021 6th International Conference on Transportation Information and Safety. ICTIS, IEEE, pp. 1244–1252. <http://dx.doi.org/10.1109/ictis54573.2021.9798507>.
- Şengül, B., Yılmaz, F., Uğurlu, Ö., 2023. Safety–security analysis of maritime surveillance systems in critical marine areas. *Sustainability* 15 (23), 16381. <http://dx.doi.org/10.3390/su152316381>.
- Steenari, J., Lwakatere, L.E., Nurminen, J.K., Talonen, J., Manderbacka, T., 2022. Mining port operation information from AIS data. *epubli*, <http://dx.doi.org/10.15480/882.4705>.
- pandas development team, T., 2020. Pandas-dev/pandas: Pandas. <http://dx.doi.org/10.5281/zenodo.3509134>.
- Uber Technologies, Inc., 2021. H3: A Hexagonal Hierarchical Geospatial Indexing System. URL <https://h3geo.org>.
- U.S. Coast Guard, 2026. Class a AIS position report (messages 1, 2, and 3). no date, (Accessed 16 January 2026), <https://www.navcen.uscg.gov/ais-class-a-reports>.
- Varlamis, I., Kontopoulos, L., Tserpes, K., Etamad, M., Soares, A., Matwin, S., 2021. Building navigation networks from multi-vessel trajectory data. *Geoinformatica* 25 (1), 69–97. <http://dx.doi.org/10.1007/s10707-020-00421-y>.
- Volkova, T.A., Balykina, Y.E., Bespalov, A., 2021. Predicting ship trajectory based on neural networks using AIS data. *J. Mar. Sci. Eng.* 9 (3), 254. <http://dx.doi.org/10.3390/jmse9030254>.
- Wang, C., Ren, H., Li, H., 2020. Vessel trajectory prediction based on AIS data and bidirectional GRU. In: 2020 International Conference on Computer Vision, Image and Deep Learning. CVIDL, pp. 260–264. <http://dx.doi.org/10.1109/CVIDL51233.2020.00-89>.
- Wen, Y., Lu, Z., Huang, L., Wen, Y., Zhang, J., Wan, C., 2021. A storage method of ship trajectory data based on ClickHouse database. In: 2021 6th International Conference on Transportation Information and Safety. ICTIS, IEEE, pp. 772–779. <http://dx.doi.org/10.1109/ICTIS54573.2021.9798612>.
- Wielen, N.v.d., McGurk, J., Barrett, L., 2024. Optimising port arrival statistics: Enhancing timeliness through automatic identification system (AIS) data. *Stat. J. IAOS* 40 (2), 421–434. <http://dx.doi.org/10.3233/sji-230100>.
- Wu, L., Xu, Y., Wang, F., 2020. Identifying port calls of ships by uncertain reasoning with trajectory data. *ISPRS Int. J. Geoinf.* 9 (12), 756. <http://dx.doi.org/10.3390/ijgi9120756>.
- Xiao, Y., Li, X., Yao, W., Chen, J., Hu, Y., 2023. Bidirectional data-driven trajectory prediction for intelligent maritime traffic. *IEEE Trans. Intell. Transp. Syst.* 24 (2), 1773–1785. <http://dx.doi.org/10.1109/TITS.2022.3219998>.
- Xiuju, F., Zhe, X., Haiyan, X., Vasundhara, J., Nasri Bin, O., Chye Poh, C., Mikael, L., 2021. AIS data analytics for intelligent maritime surveillance systems. In: *Progress in IS*. Springer International Publishing, Cham, pp. 393–411. http://dx.doi.org/10.1007/978-3-030-50892-0_23.
- Yan, Z., Cheng, L., He, R., Yang, H., 2022. Extracting ship stopping information from AIS data. *Ocean Eng.* 250 (111004), 111004. <http://dx.doi.org/10.1016/j.oceaneng.2022.111004>.
- Yu, Q., Yin, X., Geng, X., Chen, S., Yang, J., 2025. AISFormer for long-term vessel trajectory prediction. *Ocean Eng.* 340 (122098), 122098. <http://dx.doi.org/10.1016/j.oceaneng.2025.122098>.
- Zhang, X., Fu, X., Xiao, Z., Xu, H., Qin, Z., 2022. Vessel trajectory prediction in maritime transportation: Current approaches and beyond. *IEEE Trans. Intell. Transp. Syst.* 23 (11), 19980–19998. <http://dx.doi.org/10.1109/TITS.2022.3192574>.
- Zhang, D., Zhang, Y., Zhang, C., 2021. Data mining approach for automatic ship-route design for coastal seas using AIS trajectory clustering analysis. *Ocean Eng.* 236 (109535), 109535. <http://dx.doi.org/10.1016/j.oceaneng.2021.109535>.