Data Mining and Knowledge Discovery (2026) 40:56
https://doi.org/10.1007/s10618-026-01221-y

Anomaly detection for evolving maritime trajectories with continual learning

Jack Julian[1] · Yun Sing Koh[1] · Albert Bifet[2,3]

Received: 22 September 2025 / Accepted: 10 May 2026 / Published online: 9 June 2026

Abstract
Anomaly detection in live trajectory data is a critical task for ensuring safety, security, and legality in global transport. Traditional anomaly detection methods often struggle with dynamic and evolving trajectory patterns, especially as systems must adapt to new scenarios over time due to increased traffic, geopolitical events, or global warming. We propose a continual learning approach to detect anomalous activity in moving vessels. Unlike conventional static models, our method leverages continual learning to enable the model to learn from new data continuously and recognise specific behaviours dependent on position and recent movements. We implement an adapter-based framework, Continual Learning for AIS Anomalies (CLAISA), that adapts to shifting behavioural environments in transportation, ensuring the system can identify novel and evolving patterns of anomalies, such as deviations from expected routes, irregular speed changes, or unusual local movements. Evaluations on synthetic maritime trajectory datasets spanning sparsely populated waters and heavily trafficked shipping lanes demonstrate that CLAISA achieves up to a 32% decrease in error for trajectory forecasting and consistently outperforms benchmark methods in anomaly detection on synthetically generated datasets.

Keywords Continual learning · Anomaly detection · Trajectory forecasting · Deep learning · Adapters

✉ Jack Julian
jack.julian@auckland.ac.nz

Yun Sing Koh
y.koh@auckland.ac.nz

Albert Bifet
albert.bifet@waikato.ac.nz

1 School of Computer Science, University of Auckland, Auckland, New Zealand

2 AI Institute, University of Waikato, Hamilton, New Zealand

3 LTCI, Télécom Paris, IP Paris, Paris, France

1 Introduction

Maritime traffic currently accounts for approximately 90% of global trade, with steady growth projected in coming years (Wan et al. 2016). With this influx of traffic, anomaly detection is critical for maritime activity to ensure safety and security across global waters. Unusual vessel behaviour may indicate illegal activities such as smuggling or unauthorised fishing, deviations from planned routes, or safety risks due to mechanical failure or weather conditions. The detection of such anomalies allows for timely intervention, reducing the risk of accidents and enabling more effective maritime oversight.

Maritime vessel behaviour is dynamic, shaped by shifting routes, port operations, seasonal patterns, and global events. These fluctuations challenge anomaly detection systems, which must continuously adapt to redefine what constitutes "normal" behaviour. Static deep learning models, though effective on benchmark datasets (Li et al. 2024; D'afflisio et al. 2018), degrade over time when applied to evolving real-world data. Transport monitoring systems now produce massive volumes of geospatial data (March et al. 2021), offering the potential for continuous learning, but requiring models that can handle this dynamic context. For instance, maritime vessel traffic dropped by 51% during the initial COVID-19 lockdowns in March 2020 (March et al. 2021), a shift that static models fail to accommodate without retraining. Continual learning (CL) (Ho et al. 2023; Smith et al. 2023; Madaan et al. 2023; Pham et al. 2021; Buzzega et al. 2020) offers a solution by enabling models to incrementally incorporate new behavioural patterns. However, existing CL methods focus on global data distribution shifts (Zhi and Sun 2023), and struggle to adapt to fine-grained domain-specific contexts, such as distinct vessel behaviours near ports versus in open waters (Alomari and Andó 2024). This leads to overly generalised updates that miss localised behavioural changes (Bao et al. 2023). Additionally, most CL systems prioritise retention of all past knowledge (Aljundi and Kelchtermans 2019), but maritime monitoring requires selective forgetting of transient patterns, such as those caused by temporary weather disruptions, highlighting the need for mechanisms that differentiate between long-term trends and short-term anomalies.

To address these challenges, we propose Continual Learning for AIS Anomalies (CLAISA): a novel approach that enhances anomaly detection by integrating CL techniques within a transformer architecture that embeds motion and positional features separately. CLAISA introduces an adapter layer that allows the model to adapt to new information using small feed-forward layers with dynamic weights suitable for understanding short-term behaviours while retaining long-term knowledge in a frozen transformer architecture. Different behaviours are mapped to distinct adapters, allowing for behaviours to be captured without cross-domain interference. By incorporating this adapter layer, CLAISA can handle the complexity of maritime and aerospace trajectories, adjusting to new, unseen behaviours in real time without requiring retraining from scratch. The proposed transformer architecture is designed to separately process position and motion, thereby enhancing the model's ability to capture distinct patterns within trajectory data. Position embeddings capture the spatial information, while motion embeddings represent the temporal and velocity-related aspects of the trajectories. By treating these features independently within the transformer, CLAISA can recognise specific behaviours more accurately, improving its ability to detect anomalies that may arise from unusual movement patterns or deviations from expected locations.

We provide the following contributions: First, the CLAISA framework and implementation. We apply an adapter-based CL approach to detect anomalies early in AIS trajectories while accounting for changing behaviours. Second, we provide a set of definitions describing anomalies in AIS data. Unlike existing definitions for anomalies in trajectories, our definition applies to all ocean regions. Finally, we present a method of generating synthetic trajectories within a two-dimensional space with anomalous values to assist the evaluation of trajectory anomaly detection methods.

2 Related work

Recent advancements in maritime anomaly detection using Automatic Identification System (AIS) (Pallotta and Vespe 2013) data have led to more studies addressing this problem. Notable works in this area include (Laxhammar 2008; Ristic et al. 2008; Mascaro et al. 2014; D'afflisio et al. 2018; Kawaguchi 2018; Forti et al. 2019; Varlamis et al. 2019). These studies can be categorised into two broad groups: rule-based anomaly detection and learning-based anomaly detection. The rule-based approaches explicitly define abnormal behaviours through a set of predefined rules. A comprehensive list of such rules can be found in Kazemi et al. (2013). While rule-based methods are often interpretable, they have limitations, including difficulty defining abnormal behaviours. Furthermore, some terms (e.g., “fast" or “slow") are relative and challenging to implement operationally, which limits the practical applicability of these systems. On the other hand, learning-based anomaly detection methods leverage historical data to learn detection rules implicitly. Due to the absence of reliable ground truth data for maritime anomaly detection, supervised methods cannot be directly applied (Song et al. 2018; Ma et al. 2018; Bouritsas et al. 2019). Thus, unsupervised learning methods are favoured (Pallotta and Vespe 2013; Arguedas 2017; Zhao 2019). These learning-based approaches offer greater flexibility and scalability than rule-based systems, overcoming the challenge of defining exhaustive normal/abnormal behaviours. The lack of labelled anomalous data further supports the use of unsupervised techniques, which have emerged as the dominant approach in maritime anomaly detection (Pallotta and Vespe 2013; Laxhammar 2008; Forti et al. 2019).

Unsupervised learning-based methods typically involve two main stages: learning a normalcy model and detecting deviations from the learned normalcy. In the first stage, density-based spatial clustering algorithms, such as DBSCAN (Ester et al. 1996), have been widely adopted (Pallotta and Vespe 2013; Varlamis et al. 2019; d'Afflisio et al. 2018). DBSCAN is used to cluster critical points from AIS trajectories into entry, exit, and stop points within a region of interest. These factors are then used to construct a graph, with nodes representing the points and edges corresponding to maritime routes. A normalcy model is fitted to each edge using probabilistic techniques, such as Kernel Density Estimation (Pallotta and Vespe 2013), Gaussian Mixture Models (Laxhammar 2008), or multiple Ornstein-Uhlenbeck processes (Forti et al. 2019). In the second stage, anomaly detection is performed by measuring the likelihood of a new AIS track relative to the normalcy model. Detection can be done by applying thresholds on the distance to the centroid of a route (Varlamis et al. 2019), the probability of an AIS track given the normalcy model (Pallotta and Vespe 2013), or through adaptive hybrid filters, such as the Bernoulli filter (Forti et al. 2019).

Trajectory mining research has also investigated the explicit modelling of motion and positional dynamics in AIS data. Several works focus on extracting structured spatio-temporal movement patterns, for example, through tensor decomposition and spatial–temporal data mining approaches (Yang et al. 2024; Qiang et al. 2025). These methods identify latent traffic corridors and behavioural regimes directly from positional and kinematic features, providing a complementary perspective to route-based graph modelling. More recent studies employ deep spatio-temporal architectures to jointly capture motion characteristics and spatial dependencies in AIS trajectories (Ma et al. 2024). These approaches explicitly model velocity, heading changes, and interaction-aware dynamics, showing the importance of disentangling positional and motion information.

Probabilistic and reconstruction-based anomaly detection methods have been explored for maritime trajectories. Autoencoder and sequence-to-sequence models detect anomalies through reconstruction error (Nguyen et al. 2021), while recurrent neural networks and predictive likelihood models identify abnormal behaviour via forecasting deviation (Song et al. 2018; Ma et al. 2018). Isolation-based and graph-based online anomaly detectors have also been proposed for streaming trajectory data (Chen et al. 2013; Varlamis et al. 2019).

Online and streaming anomaly detection has received increasing attention in dynamic environments where data distributions evolve over time. Unlike batch learning approaches, online methods update models incrementally as new observations arrive. In trajectory analysis, isolation-based techniques such as iBOAT (Chen et al. 2013) perform incremental anomaly scoring without requiring full retraining. Adaptive filtering and recursive probabilistic estimators have been used for real-time maritime anomaly detection (Forti et al. 2019). Concept drift adaptation and incremental clustering methods have been extensively studied to address non-stationarity (Gama et al. 2014; Lu et al. 2018). Surveys on streaming anomaly detection further highlight the importance of windowing mechanisms, density updates, and incremental model refinement in evolving systems (Ahmed et al. 2016; Pimentel et al. 2014).

3 Anomaly detection for AIS trajectories

The Automatic Identification System (AIS) is a shipboard broadcast system responsible for transmitting and receiving data pertaining to a ship's current position, velocity and status, as well as additional information about the vessel and its identifiers. As the International Maritime Organisation requires AIS to be fitted aboard all international voyaging ships with 300 or more gross tonnage and all passenger ships regardless of size, AIS datasets capture global maritime behaviours and are often used in machine learning approaches for trajectory prediction (Hexeberg et al. 2017; Li et al. 2024; Wu et al. 2022).

We aim to use AIS data points to describe trajectories and define anomalies. A vessel point $x_t = (p_t, m_t, d)$ represents the AIS signal from a vessel at time *t*, which consists of the physical position $p_t \in \mathbb{R}^2$ (i.e. latitude/longitude), motion features $m_t \in \mathbb{R}^2$ (i.e. speed/heading), and the descriptive feature *d* which contains the vessel type (i.e. cargo or fishing vessel). Other features of an AIS signal are discarded. Given a sequence of vessel points, we define a vessel trajectory as follows:

Definition 1 *(Vessel Trajectory)* A vessel trajectory is a chronologically ordered sequence of vessel points generated by a single vessel traversing a geospatial area between times t_0 and t_n: $\mathrm{x} = \{x_{t_0}, \cdots, x_{t_n}\}$. A *subtrajectory* is the sequence of vessel points between time $t - \Delta t(T_p + 1)$ and *t*:

$$\mathrm{x}_{t-\Delta t(T_p+1)\to t} = \{x_{t-\Delta t(T_p+1)}, \cdots, x_{t-\Delta t}, x_t\},$$

where T_p is the number of vessel points in the trajectory and Δt is the timestep between vessel points. Because we assume Δt to be constant, we omit Δt and set the start time $t_s = t - \Delta t(T_p + 1)$ for more simplistic notation: $\mathrm{x}_{t_s \to t} = \{x_{t_s}, \cdots, x_{t-1}, x_t\}$.
Position features p_t and motion features m_t exist in a continuous domain $\mathbb{R}^2$, which may hinder interpretability and model robustness in deep learning applications (Nguyen et al. 2021; Chen et al. 2013). To address this, we discretise the continuous space into a finite grid G of size $H \times W$ and represent each position and motion feature as a one-hot encoded vector corresponding to a specific grid cell. Let $\mathscr{E} : G \to 0, 1^{H\times W}$ be a bijective encoding function that maps a grid cell $g = (i, j) \in G$ to a one-hot vector e_{ij}, where the (i, j)-th component is one and all others are 0. The full one-hot encoded trajectory is denoted as $\bar{\mathrm{x}} = (\mathscr{E}(g_1), \ldots, \mathscr{E}(g_n))$, where each $\mathscr{E}(g_i)$ encodes the grid cell g_i corresponding to the original point x_i.

To map continuous features to grid cells, we define two quantisation functions: $\rho_p : \mathbb{R}^2 \to G$ maps position features p_t to the corresponding grid cell, and $\rho_m : \mathbb{R}^2 \to G$ maps motion features m_t to the corresponding grid cell. These functions partition the continuous space into $H \times W$ non-overlapping rectangular bins, and assign each point to the cell in which it falls. The mapped trajectory is a discrete sequence $g_1, \ldots, g_n$, where $g_i = \rho_p(x_i)$, and the one-hot encoded version is $\bar{\mathrm{x}} = (\mathscr{E}(g_1), \ldots, \mathscr{E}(g_n))$.

As we will deal only with mapped trajectories from now on, we remove the mapped qualifier and refer to a mapped trajectory $\bar{\mathrm{x}}$ as x. We adapt definitions from Chen et al. (2013) to form the foundation for anomalies, and extend them to account for anomalies in AIS trajectories. Before we define an anomalous trajectory, we define functions used to index and reference grid cells in a trajectory x.

First, we define a function to retrieve all neighbouring grid cells. Let $\eta \in \mathbb{N}$ be a parameter that controls the radius of the neighbourhood. We define the neighbourhood function $N_\eta : G \to \mathscr{P}(G)$, where G is the set of all grid cells and $\mathscr{P}(G)$ denotes its power set. For a given grid cell $g = (i, j) \in G$, the set $N_\eta(g)$ contains all grid cells $(i', j') \in G$ such that:

$$|i - i'| \leq \eta \quad \text{and} \quad |j - j'| \leq \eta$$

This defines a square neighbourhood of side length $2\eta + 1$ centred at g. When $\eta = 1$, $N_1(g)$ includes the eight adjacent neighbours and g itself (if g is not on the grid's border). Although similar to a kernel function, N_η does not incorporate weighted aggregation or convolution, and is purely used as a proximity constraint used to define approximate path equivalence. For a trajectory x, we write $N_\eta(g) \cap \mathrm{x} \neq \emptyset$ to denote that at least one of the neighbours of g is in the trajectory x.

Second, we define $\mathrm{pos} : \mathbb{T} \times \mathscr{P}(G) \rightarrow \mathbb{N}$ which, given a trajectory x and set of cells G, returns the *cell index* of some $g \in G$ in x:

$$\mathrm{pos}(\mathrm{x}, G) = \begin{cases} \mathrm{argmin}_{i \in \mathbb{N}}(x_i \in G), & \mathrm{if} G \cap \mathrm{x} \neq \emptyset \\ \infty, & \mathrm{otherwise.} \end{cases}$$

For example, if $\mathrm{x} = \{g_1, g_2, g_3\}$, then $\mathrm{pos}(\mathrm{x}, \{g_3\}) = 3$ and $\mathrm{pos}(\mathrm{x}, \{g_5\}) = \infty$.

An example of the encoding of x into a grid structure G is shown in Fig. 1. We are now ready to define anomalous trajectories. Definitions in Chen et al. (2013) do not capture anomalies based on vessel types, for instance, a cargo ship entering an anomalous area. To address this problem, we introduce the following definitions:

Definition 2 *(hasPath)* We define the function $\mathrm{hasPath} : \mathscr{P}(\mathbb{T}) \rightarrow \mathscr{P}(\mathbb{T})$ that returns the set of trajectories with the same vessel type that contains all of the points in x in the correct order. The points do not need to be sequential; it suffices that they appear in the same order:

$$\mathrm{hasPath}(X, \mathrm{x}) = \left\{ \mathrm{x}' \in X \;\middle|\; \begin{array}{l} \forall 1 \leq i \leq n, N_\eta(g_i) \cap \mathrm{x}' \neq \emptyset, \\ \forall 1 \leq i \leq j \leq n \\ \mathrm{pos}(\mathrm{x}', N_\eta(g_i)) \leq \mathrm{pos}(\mathrm{x}', N_\eta(g_j)) \\ \forall x \in \mathrm{x}, x' \in \mathrm{x}', d \in x, d' \in x', d = d' \end{array} \right\}$$

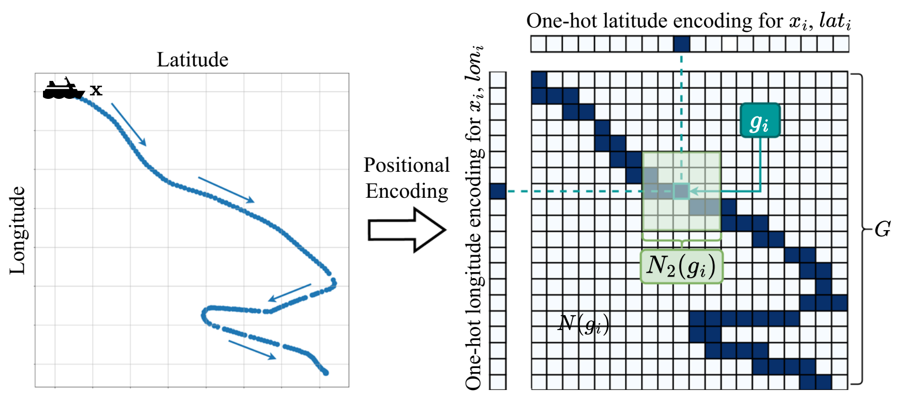


Fig. 1 Example of a trajectory encoded into grid matrix G

We define a binary variable $M_{x'} = 1_{\text{hasPath}(X,x)}(x')$ for each trajectory $x' \in X$. These definitions can determine whether two trajectories are identical or follow the same path within neighbouring bounds, or whether a new trajectory is *θ-anomalous*:

Definition 3 (*θ-Anomalous Trajectories*) Given threshold $0 < \theta \leq 1$, trajectory x is *θ-anomalous* with respect to a set of observed trajectories X if:

$$\text{anom}_\theta(X, x) = \frac{1}{|X|} \sum_{x' \in X} M_{x'} \leq \theta.$$

These definitions rely heavily on X, which is difficult to sample in practice. In large regions, low-traffic ports may falsely report more θ-anomalous trajectories due to fewer comparable examples. During events like storms, X must quickly adapt to reflect changing vessel behaviour, then reset once conditions normalise. Thus, the following sections focus on a deep learning approach for detecting θ-anomalous trajectories without relying on a manually curated X.

4 Continual learning for anomaly detection

Our main objective is to predict the trajectory of a target vessel, assuming a free range of movement on a two-dimensional plane, and determine whether the difference between the predicted and actual trajectory should be considered anomalous. Specifically, we aim to accurately model $p(x_{t+1:t+l}|x_{0:t})$ and acquire $q(x_{t+1:t+l}|x_{0:t})$ to calculate the KL divergence between distributions. Furthermore, we expect that for $x_{t+1:t+l} = x_{u+1:u+l}$ at times t and u, $p(x_{t+1:t+l}|x_{0:t}) \neq p(x_{u+1:u+l}|x_{0:u})$. The predictive model must, therefore, adapt to changes over a sequence of trajectories.

We introduce CLAISA, an adaptive method for trajectory prediction and anomaly detection in dynamic maritime environments. CLAISA builds on a frozen pre-trained transformer backbone, augmented with trainable adapters that learn spatio-temporal patterns specific to evolving behaviours. CLAISA consists of two key components. First, CLAISA separates its frozen weights into positional and motion embeddings to disentangle different aspects of trajectory behaviour, detailed in Sect. 4.1, and continually trains adapters to produce a predicted trajectory, detailed in Sect. 4.2. Second, it includes an anomaly detection module designed to identify abnormal patterns within a dynamic environment, described in Sect. 4.3.

4.1 Knowledge embedding

Here we discuss the knowledge embedding step of the CLAISA framework. We approach the problem of trajectory forecasting by addressing the challenge of effectively capturing the distinct relationship between motion dynamics and spatial positioning. Separately embedding motion and positional features improves the model's ability to handle varying behaviours based on movement patterns and spatial coordinates. Furthermore, we approach the problem of adjusting to changing behaviours in real-world trajectories by introducing a dynamic adapter to enable CL. First, we describe the motion embedding step.

Motion Embedding. For any trajectory dataset with positional features, additional motion features may be extracted, improving predictive performance and exhibiting high Shapley values (Li et al. 2023). The most important features we consider in this work are speed and heading, which represent a vessel's velocity. However, previous works indicate that it is difficult to encode complex behaviours through this feature space (Nguyen et al. 2021), and so we embed each motion vector m^t into a higher-dimensional vector space:

$$e_t^m = \sigma(\mathrm{W}m_t + b) \in \mathbb{R}^{d_m}. \tag{1}$$

We use a multi-layer perceptron to embed the vector in a d-dimensional space, shown to effectively learn feature mappings (Vahdat et al. 2020).

Position Embedding. Positional data is represented by the geospatial features p_t, usually latitude and longitude. We use a similar mapping to Eq. 1 for positional features, substituting m_t for p_t and separate weights to acquire the vector $e_t^p \in \mathbb{R}^{d_p}$. Next, we describe the transformer architecture and detail how positional and motion features are separately extracted. CNNs are commonly used to capture spatial dependencies, however previous work has indicated that a finer granularity is required to represent spatial information in positional trajectories (Wang et al. 2018).

4.2 Trajectory forecasting

The core of our proposed method is based on the transformer architecture, which has proven highly effective for sequence-to-sequence tasks, particularly when capturing long-term dependencies in time series data (Wilbur et al. 2021; Li et al. 2024; Shi et al. 2023; Yang et al. 2024), making it ideal for trajectory forecasting. Transformers consist of a series of stacked attention layers, and each attention layer performs auto-regressive modelling using a multi-head self-attention mechanism. The query, key, and value matrices are obtained by linearly projecting the input sequence e_t^m or e_t^p in the first block for motion and positional encoding respectively and the output of the previous layer for subsequent blocks. Specifically, for the first layer, the input sequence depends on the embedding step described previously, and for subsequent layers, the sequence corresponds to the previous layer's output. The scalar d corresponds to the dimension of the input vector.

We use descriptive features d as the base vessel representation and introduce motion and positional features through multi-head attention via the motion and positional attention extractors. The motion attention extractor consists of two encoder layers that learn interactions between motion embeddings e_t^m and descriptive features d, which consist of vessel type and status in a maritime setting. As shown in Fig. 2, the first layer computes multi-head attention with e_t^m as value and key, and d as query, followed by residual connection and layer normalisation. The second layer applies multi-head self-attention and outputs v_t^m. The positional attention extractor, also visualised in Fig. 2, has a similar structure. It uses v_t^m as the value and key, and positional embeddings e_t^s as query in the first layer, then applies residual connection

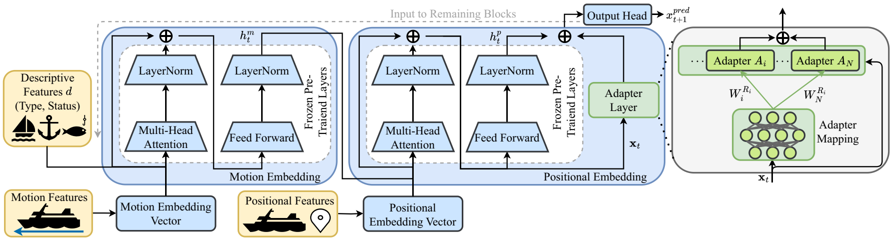


Fig. 2 Overview of the CLAISA method producing a predicted AIS signal. Adapter layers are inserted after multi-head attention layers in a transformer architecture. Weights within the adapter layer are updated while all other parameters remain frozen within the architecture. A router weights each adapter output based on its importance to the input

and normalisation. The second layer performs self-attention and outputs v_t^p. Each extractor spans one layer, and the full architecture uses L stacked layers.

Classification-based forecasting has demonstrated promising results versus regression in transformer models for multimodal distributions common to maritime data (Li et al. 2024). We adopt this approach by modelling $p(e_{t+l}|e_{0:t+l-1})$ as four categorical distributions over latitude, longitude, speed, and heading. At the output of the final attention layer, the model produces a vector l_t with the same dimensionality as the input h_t, which is subsequently used to model the distribution $p(h_{t+l}|e_{0:t+l-1})$, as detailed in the following sections.

Adapters. We propose an adapter layer alongside the positional attention extractor to facilitate CL while avoiding interference between positions and distinct behaviours. Weights within the adapters change with every new input, while other parameters within the transformer architecture remain frozen after a training period. CLAISA consists of multiple adapters, denoted as $\mathrm{A} = \{A_i\}_{i=1}^N$, where N is the number of predefined adapters. A feed-forward mapping of input vectors to adapters determines the appropriate outputs of these adapters by utilising a gated averaging mechanism weighted by adapters trained on specific vessel types. An *adapter* refers to a lightweight, task-specific module inserted into a pre-trained deep learning model to allow it to learn new behaviours without overwriting or forgetting previously learned information. Specifically, we adopt the adapter layer proposed by Houlsby et al. (2019), where each adapter, denoted as A_i, operates by down-projecting and up-projecting an input h:

$$A_i(\mathrm{h}) = \mathrm{h} + f(\mathrm{h}\theta_{\mathrm{down}})\theta_{\mathrm{up}},$$

where $f(\cdot)$ represents a nonlinear activation function such as ReLU, and the weights $\theta_{\mathrm{down}} \in \mathbb{R}^{d\times r}$ and $\theta_{\mathrm{up}} \in \mathbb{R}^{r\times d}$ are used to project the input to a lower-dimensional subspace and then back to the original dimension, respectively. Here, r is the reduced dimensionality introduced by the adapter layer. For each input, adapter outputs are weighted by a mapping layer designed to prioritise adapters that produce the most accurate outputs. Alongside the adapters, a feed-forward layer with parameters θ_{map} is initialized to take in the same input h as the adapters and output a vector w:

$$\mathrm{w} = \sigma(\mathrm{h}\theta_{map} + b_{map}) \in \mathbb{R}^N,$$

where $\sigma(\cdot)$ is the softmax function and b_{map} is the bias for the mapping. The output w contains one item per adapter, $\mathrm{w} = [w_1, \cdots, w_N]$, where w_i is the weighting for adapter A_i. Specifically, the combined output of the adapter layer is computed as:

$$\mathrm{y} = \sum_{i=1}^{N} w_i A_i(\mathrm{h}),$$

where the output y matches the dimension of input h, and y is added to v_t^p to create the final output from the block $\mathrm{d}_t = v_t^p + \mathrm{y}$, as seen in Fig. 2. Parameters θ_{map} are updated depending on the adapter with the lowest error on the final trajectory. The loss function for the mapping is given by:

$$\mathscr{L}_{map}(\mathrm{y}) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{2}$$

where $y_i \in \mathrm{y}$ and $\hat{y}_i \in \hat{\mathbf{y}}$. The vector $\hat{\mathbf{y}} \in \{0,1\}^N$ is a one-hot vector where each $\hat{y}_i = 1$ if $A_i(\mathrm{h})$ has the lowest trajectory error and all other $\hat{y}_j = 0$. We now detail how CLAISA update its weights and models a probability distribution for future trajectories.

Modelling the conditional distribution $p(\mathrm{x}_{t+1:t+l} \mid \mathrm{x}_{0:t})$ directly is challenging due to the continuous nature of the target space. To simplify this problem, we discretise the four continuous variables- latitude, longitude, Speed over Ground (SOG), and Course over Ground (COG)- into grid cells and reformulate the prediction task as a multi-head classification problem. Let h_t denote the discretised representation of the continuous trajectory x_t. For each predicted time step, the model outputs four sets of logits corresponding to the discretised grids of the four features. Given a logit vector $\mathrm{z}^{(v)} \in \mathbb{R}^K$ for variable $v \in \{\mathrm{lat}, \mathrm{lon}, \mathrm{SOG}, \mathrm{COG}\}$, where G_v is set of grid cells corresponding to v, we compute the probability distribution over the grid using the softmax function:

$$p(\mathrm{h}_{t+1}^{(v)} = g \mid \mathrm{x}_{0:t}) = \frac{\exp(z_k^{(v)})}{\sum_{j=1}^{K} \exp(z_j^{(v)})} \quad \text{for } g \in G_v.$$

This yields a categorical distribution over the discretised space for each variable, approximating the original continuous conditional distribution. Because $\mathrm{h}_t \to \mathrm{x}_t$ is one-to-one, $p(\mathrm{h}_{t+1:t+l} | \mathrm{x}_{0:t}) = p(\mathrm{x}_{t+1:t+l} | \mathrm{x}_{0:t})$. The loss function for transformer parameters is:

$$\mathscr{L}(\mathrm{h}_t, p_t) = -\sum_{l=1}^{L} CE(\mathrm{h}_{t+l}, p_{t+l}), \tag{3}$$

where $p_t = p(\mathrm{h}_{t+1:t+l}|\mathrm{x}_{0:t})$ and $CE(\cdot)$ is the cross-entropy function. The categorical cross-entropy better captures multimodal behaviours (Li et al. 2024), such as a point when a ship exits a port, where multiple directions may be taken.

During continual learning, trajectories are processed sequentially, and model parameters are updated incrementally without revisiting previous trajectories. Gradients of the forecasting loss in Eq. 3 are propagated only through the adapter weights $\theta_{\text{down}}, \theta_{\text{up}}$ and the mapping parameters θ_{map}, b_{map}, while all other transformer parameters remain fixed. Because the adapter output is computed as a weighted combination $\mathrm{y} = \sum_{i=1}^{N} w_i A_i(\mathrm{h})$, gradients from $\mathscr{L}$ flow through all adapters proportionally to their routing weights. The mapping loss in Eq. 2 updates θ_{map} by encouraging a higher weight on the adapter that yields the lowest trajectory error over the prediction horizon. No replay buffer or explicit regularisation is used, and continual adaptation arises from residual updates within the adapters and routing-based behavioural specialisation, similarly to existing dynamic architecture approaches (Yu et al. 2024).

4.3 Anomaly detection

Here, we detail the anomaly detection module of CLAISA. Although Definition 3 defines anomalous trajectories with respect to an experienced set *X*, the choice of *X* may not represent current behaviours. So far, we have described how CLAISA models $p(\mathrm{x}_{t+1:t+l}|\mathrm{x}_{0:t})$. We now define how an anomaly is identified using this prediction.

Definition 4 *(Trajectory Anomaly)* Given the conditional probability distribution of the future trajectory *l* points ahead of time *t* given past trajectories $p(\mathrm{x}_{t+1:t+l}|\mathrm{x}_{0:t})$ and the actual probability distribution of the trajectory $q(\mathrm{x}_{t+1:t+l}|\mathrm{x}_{0:t})$, a trajectory is anomalous if the KL divergence is equal to or greater than a threshold δ:

$$KL(q(\mathrm{x}_{t+1:t+l}|\mathrm{x}_{0:t}) \parallel p(\mathrm{x}_{t+1:t+l}|\mathrm{x}_{0:t})) \geq \delta.$$

We choose KL divergence for its theoretical connection to the model loss detailed in Eq. 3, following known relative entropy relationships (Cover and Thomas xxxx):

Theorem 1 *For all p_t, q_1 and q_2 where $\mathscr{L}(q_1, p_t) > \mathscr{L}(q_2, p_t)$, then $KL(q_1 \parallel p_t) > KL(q_2 \parallel p_t)$.*

Proof The loss function $\mathscr{L}(q, p_t)$ is related to the KL divergence by the formula $CE(p_t, q) = KL(q \parallel p_t) + H(p_t)$, where $H(p_t)$ is the entropy of p_t . Since $H(p_t)$ is constant for all distributions q, $\mathscr{L}(q_1, p_t) > \mathscr{L}(q_2, p_t)$ implies $KL(q_1 \parallel p_t) > KL(q_2 \parallel p_t)$. □

By Theorem 1, the loss of any newly encountered point h_t for a modelled $p(\mathrm{h}_t|\mathrm{x}_{0:t})$ increases with $KL(q(\mathrm{h}_t|\mathrm{x}_{0:t}) \parallel p(\mathrm{h}_t, \mathrm{x}_{0:t}))$, the distance of the model prediction from the observed event. Therefore, a trajectory with a sufficiently high loss will be identified as anomalous. Through the training scheme proposed above, we acquire $p(\mathrm{h}_{t+1:t+l}|\mathrm{x}_{0:t})$. To determine whether a trajectory is anomalous by Definition 4, we require the actual probability distribution of the trajectory $q(\mathrm{x}_{t+1:t+l}|\mathrm{x}_{0:t})$. Because $\mathrm{x}_{t+1:t+l}$ is known during anomaly detection, we set:

$$q(\mathrm{x}_{t+1:t+l}|\mathrm{x}_{0:t}) = \delta_a(\mathrm{x}_{t+1:t+l}) = \frac{1}{|a|\sqrt{\pi}} e^{-(\mathrm{x}_{t+1:t+l}/a)^2}, \tag{4}$$

where $\delta_a(\mathrm{x}_{t+1:t+l})$ is the dirac distribution on $\mathrm{x}_{t+1:t+l}$ with a as a near-zero value, which we set as 0.001. This acts as an impulse on the actual predicted trajectory, and an anomaly is classified following Definition 4.

5 Evaluation of CLAISA

We evaluate CLAISA using six real-world AIS datasets and a synthetic anomaly dataset. After detailing the experimental hyperparameters, we compare CLAISA's trajectory forecasting error with state-of-the-art methods and its anomaly detection performance with benchmark algorithms. We also investigate the sensitivity of CLAISA by varying key hyperparameters N and δ.

5.1 Experiment setup

Datasets. We conduct experiments using six publicly available AIS datasets to evaluate the predictive accuracy of CLAISA and a synthetic dataset to assess its anomaly detection capabilities. The AIS datasets consist of real-world vessel trajectory data, constrained by specific geographic boundaries (latitude and longitude) from distinct regions and defined temporal intervals. The regions are the Gulf of Mexico (Oceanic 2024), the Kattegat Strait (Authority et al. 2024), the North Sea in Denmark (Authority et al. 2024), the USA East Coast (Oceanic 2024), the port of Piraeus (Tritsarolis and Kontoulis 2021), and Ushant, France (Gloaguen et al. 2023). These datasets capture a wide range of behaviours, spanning sparsely populated waters, busy shipping routes, fishing zones, and port areas.

We also generate a synthetic dataset with labelled anomalies to evaluate detection methods. A grid G is defined with start and end points $g_s, g_f \in G$. A vessel moves from $g_i = (x_i, y_i)$ to a neighboring cell $g_{i+1} \in N_1(g_i)$. Using A* Search with a heuristic $H(x_i, y_i)$ as the L2 distance to g_f, we obtain the shortest path from g_s to g_f. We then modify this heuristic as follows:

$$H(x_i, y_i) \sim \mathcal{N}\left(\sqrt{(x_i - x_f)^2 + (y_i - y_f)^2}, \sigma^2\right),$$

with trajectory noise σ^2. With this heuristic, we relate η and σ to $P\left(\mathrm{anom}_\theta(X, \mathrm{x}) < \theta\right)$, the probability that a generated trajectory is θ-anomalous. We assume all $\mathrm{x}' \in X$ are i.i.d. trajectories generated by the stochastic A* algorithm with noise σ. Then each $M_{\mathrm{x}'}$ is an independent Bernoulli variable with:

$$P(M_{\mathrm{x}'} = 1) = p_{\mathrm{match}}(\eta, \sigma),$$

where $p_{\mathrm{match}}(\eta, \sigma)$ is the probability that a generated trajectory matches x under $\mathrm{hasPath}$. This probability depends on η, where the larger the spatial tolerance of the neighbourhood, the higher the probability, and σ, where a larger σ induces more deviation in the path and lowers the match probability. Hence,

$$\sum_{\mathrm{x}' \in X} M_{\mathrm{x}'} \sim \mathscr{B}(|X|, p_{\mathrm{match}}(\eta, \sigma)),$$

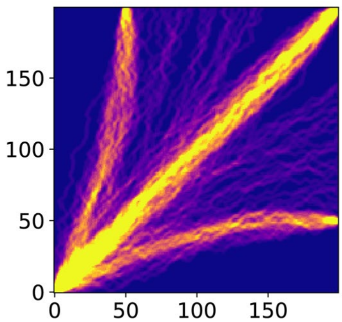
and so:

$$P(\mathrm{anom}_{\theta}(X, \mathrm{x}) < \theta) = F_{\mathscr{B}}(\lfloor \theta |X| \rfloor - 1;\ |X|,\ p_{\mathrm{match}}(\eta, \sigma)),$$

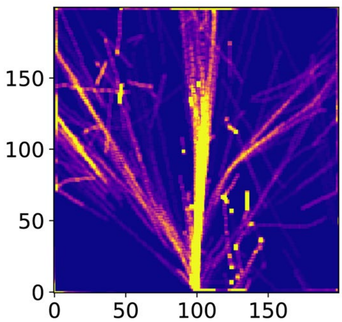
where $F_{\mathscr{B}}(k; n, p)$ is the CDF of the binomial distribution. An example of synthetic trajectories is provided in a heatmap in Fig. 3.

Preprocessing. CLAISA is designed to process streams of signals in real time. Erroneous position and speed messages were removed, and the SOG values for AIS datasets were truncated to a maximum of 30 knots to ensure realistic readings (Nguyen et al. 2021). Discontiguous trajectories with a gap of more than 2 h between consecutive signals were divided into two contiguous trajectories. Additionally, all trajectories were resampled to a 10-minute resolution using linear interpolation.

Hyperparameters. Our results were obtained using the following hyperparameters, unless otherwise stated in sensitivity or ablation experiments. The resolution of the grid mapping is 0.01^{circ} for latitudes and longitudes, 1 knot for SOG, and 5^{circ} for COG. When applied to the corresponding regions on our datasets, the dimensions of the position grid are 256×256, and the dimensions of the motion grid are 128×128. The input trajectory length was set to 3 hours, and the prediction horizon was 2 hours. Except for sensitivity tests, we use $N = 10$ adapters during training and an anomaly threshold $\delta = 1.0$. Experiments were conducted on an NVIDIA A100



(a) Synthetic.



(b) Real.

Fig. 3 Example of synthetic trajectory generation on a 200×200 grid, and real trajectories mapped to the same space

GPU. Python code implementing CLAISA and additional hyperparameters can be found here: https://github.com/jjul482/CLAISA.

Baselines. We compared the trajectory prediction performance of our approach against eight deep learning models for AIS forecasting: Long Short-Term Memory (LSTM) (Forti et al. 2020), Clustered LSTM (C-LSTM) (Capobianco et al. 2021), Bidirectional Gated Recurrent Unit (Bi-GRU) (Wang et al. 2020), and Bidirectional LSTM (Bi-LSTM) (Wu et al. 2022). Additionally, we assess advanced methods such as Masked Autoencoders for Trajectory Prediction (Traj-MAE) (Chen et al. 2023), a Transformer-based approach TrAISformer (Li et al. 2024), Gated Spatio-Temporal Graph Aggregation Network (G-STGAN) (Zhang et al. 2023), and Continual Learning-based Trajectory Prediction with Memory Augmented Networks (CLTP-MAN) (Yang et al. 2022). Finally, we compare CLAISA's anomaly detection capabilities with existing anomaly detection methods designed for AIS signals, such as Bayesian Networks (Mascaro et al. 2014), Kinetic Interpolation (K-Interpolation) (Guo et al. 2021), Multi-Task Artificial Neural Networks (ANN) (Singh and Heymann 2020), and the *a-contrario* detector GeoTrackNet (Nguyen et al. 2021).

Evaluation Metrics. To evaluate the predictive performance of AIS forecasting methods, we average the haversine distance d_t between the predicted and actual position at each time step t:

$$d_t = 2R \arcsin\left(\sqrt{\sin^2(\bar{lat}) + \cos(lat_{pred})\cos(lat_{true})\sin^2(\bar{lon})}\right),$$

where R is the radius of the Earth, $\bar{lat} = \frac{1}{2}(lat_{true} - lat_{pred})$ and $\bar{lon} = \frac{1}{2}(lon_{true} - lon_{pred})$. The average displacement error (ADE) is the average d_t for all time steps t for all trajectories.

We evaluate anomaly detection using the area under the ROC curve (AUC) and introduce Early Anomaly Detection (EAD), a metric that quantifies how soon anomalies are correctly identified. While standard metrics like precision, recall, F1-score, and AUC measure detection accuracy, they ignore detection timing. In time-sensitive settings such as monitoring ship movements, early detection can be critical for preventative action. EAD addresses this by explicitly rewarding early identification:

$$EAD = \frac{2e^{s(a-t)}}{1 + e^{s(a-t)}},$$

where a is the time of detection, t is the true anomaly onset time, and s is the severity of the anomaly. We set $s = 1 - \theta$ for θ-anomalies. The change in EAD with different s is shown in Fig. 6a, where we observe that EAD is bounded between 0 and 1. A higher EAD indicates that the anomaly is detected earlier.

Warm-up Phase. Though CLAISA is a continual learning method, the frozen backbone must first be trained to acquire an underlying understanding of trajectory dynamics. In practice, we expect this will be done by training the transformer layers on historical AIS data. In our experiments, the earliest 50% of trajectories are used to train the full transformer backbone using the categorical forecasting objective defined in Eq. 3. After this warm-up phase, backbone parameters are frozen. The remaining 50% of the data is processed sequentially, where only adapter and routing parameters are updated to enable continual adaptation.

5.2 Experimental results

Predictive Performance. Table 1 compares the ADE of baseline trajectory prediction models with CLAISA. While CLTP-MAN and CLAISA are both CL approaches, all other methods are deep-learning algorithms that typically stop training after experiencing a pre-defined training dataset. To allow for a fairer comparison, we perform batch retraining on non-CL methods, where models are retrained on the latest one-month period of data. For a qualitative analysis of our method, Fig. 4 plots the predicted and actual trajectories of the CLAISA model on maps corresponding to four of our benchmark datasets. We find that CLAISA captures a variety of behaviours exhibited by a range of vessels within constricted port areas, open seas, fishing areas and shipping lanes.

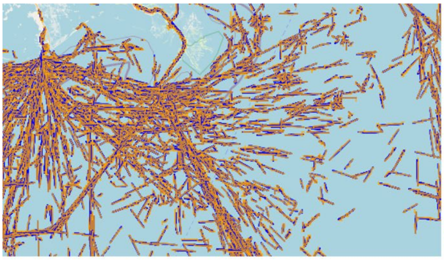
Anomaly Detection Performance. We evaluate the anomaly detection capabilities of CLAISA and existing baselines for anomaly detection for maritime trajectories on our synthetic dataset. We use two baselines to demonstrate effectiveness: AUC and EAD. Figure 5ashows the AUC with varying θ of baseline methods. A higher

Table 1 Average ADE for trajectory prediction in nautical miles

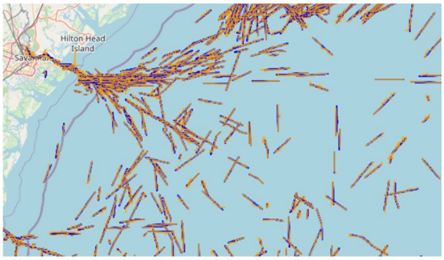
Methods	Open sea		
	Gulf of Mexico	Denmark	East Coast
LSTM (Forti et al. 2020) (2020)	1.21 ± 0.16	1.30 ± 0.20	1.31 ± 0.21
C-LSTM (Capobianco et al. 2021) (2021)	1.01 ± 0.14	1.03 ± 0.11	1.04 ± 0.16
Bi-LSTM (Wu et al. 2022) (2022)	0.98 ± 0.08	1.11 ± 0.08	1.09 ± 0.07
Bi-GRU (Wang et al. 2020) (2020)	0.89 ± 0.05	0.98 ± 0.07	0.97 ± 0.09
Traj-MAE (Chen et al. 2023) (2023)	0.70 ± 0.04	0.81 ± 0.03	0.78 ± 0.03
TrAISformer (Li et al. 2024) (2024)	0.66 ± 0.02	0.74 ± 0.04	0.70 ± 0.04
G-STGAN (Zhang et al. 2023) (2023)	0.72 ± 0.05	0.92 ± 0.06	0.94 ± 0.05
CLTP-MAN (Yang et al. 2022) (2022)	0.65 ± 0.03	0.75 ± 0.05	0.77 ± 0.04
CLAISA	$\mathbf{0.58 \pm 0.03}$	$\mathbf{0.65 \pm 0.02}$	$\mathbf{0.62 \pm 0.03}$
Methods	Port areas		
	Kattegat	Piraeus	Ushant
LSTM (Forti et al. 2020) (2020)	1.32 ± 0.19	1.33 ± 0.22	1.27 ± 0.25
C-LSTM (Capobianco et al. 2021) (2021)	1.12 ± 0.19	1.14 ± 0.11	1.10 ± 0.19
Bi-LSTM (Wu et al. 2022) (2022)	1.03 ± 0.13	1.07 ± 0.10	0.97 ± 0.23
Bi-GRU (Wang et al. 2020) (2020)	0.97 ± 0.07	1.05 ± 0.11	0.95 ± 0.13
Traj-MAE (Chen et al. 2023) (2023)	0.74 ± 0.05	0.87 ± 0.05	0.82 ± 0.07
TrAISformer (Li et al. 2024) (2024)	0.70 ± 0.03	0.83 ± 0.04	0.77 ± 0.03
G-STGAN (Zhang et al. 2023) (2023)	0.89 ± 0.06	0.84 ± 0.05	0.82 ± 0.05
CLTP-MAN (Yang et al. 2022) (2022)	0.86 ± 0.04	0.82 ± 0.03	0.79 ± 0.11
CLAISA	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.72 \pm 0.05}$	$\mathbf{0.70 \pm 0.06}$

Non-CL methods are retrained on one-month batches. Gulf of Mexico, Denmark and East Coast are datasets that represent open sea areas, and Kattegat, Piraeus and Ushant are datasets that represent port areas to capture a variety of different behaviours

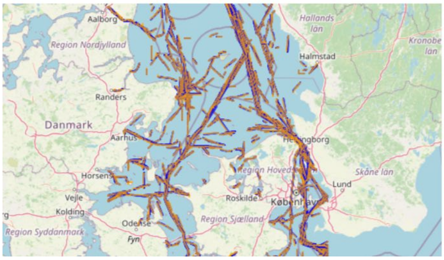
The lowest ADE for each dataset is bolded



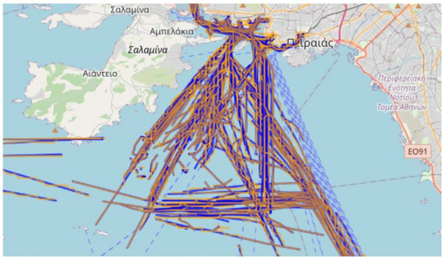
(a) Gulf of Mexico.



(b) East Coast.

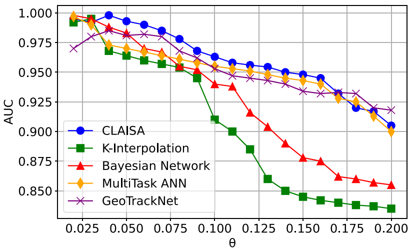


(c) Kattegat.

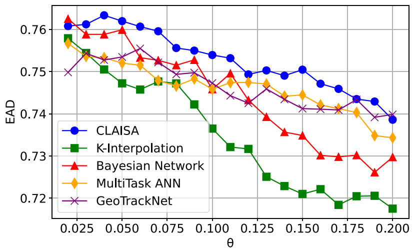


(d) Piraeus.

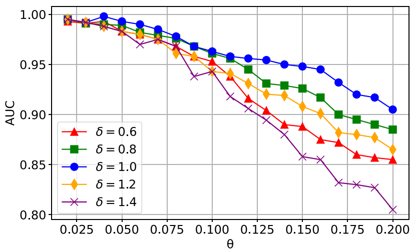
Fig. 4 Visualisation of CLAISA trajectory predictions on benchmark datasets. Actual trajectory is given in blue, overlayed with CLAISA predictions in orange



(a) AUC values for each anomaly detection model under varying θ.

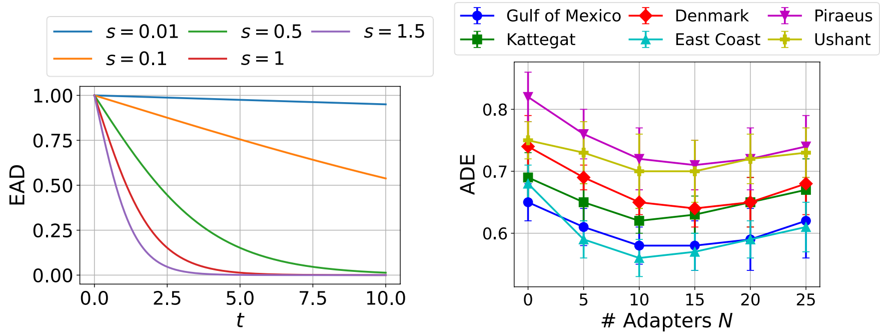


(b) EAD values for each anomaly detection model under varying θ.



(c) AUC values for a range of anomaly detection thresholds δ under varying θ.

Fig. 5 Anomaly detection results on our synthetic trajectory dataset



(a) Variation in EAD with time t for severities s.

(b) Average ADE with varying number of adapters N.

Fig. 6 EAD for different severities, and ADE of CLASIA with varying adapters

Table 2 Ablation study on embedding strategy

Methods	Open sea		
	Gulf of Mexico	Denmark	East Coast
Single Embedding	0.65 ± 0.03	0.73 ± 0.05	0.68 ± 0.05
CLAISA	$\mathbf{0.58 \pm 0.03}$	$\mathbf{0.65 \pm 0.02}$	$\mathbf{0.62 \pm 0.03}$
Methods	Port areas		
	Kattegat	Piraeus	Ushant
Single Embedding	0.69 ± 0.05	0.81 ± 0.05	0.78 ± 0.04
CLAISA	$\mathbf{0.62 \pm 0.02}$	$\mathbf{0.72 \pm 0.05}$	$\mathbf{0.70 \pm 0.06}$

We compare CLAISA's dual motion–positional embedding design against a single unified embedding across open sea and port datasets. Results are given as average ADE for trajectory prediction in nautical miles

The lowest ADE for each dataset is bolded

AUC indicates a higher true positive rate and a lower false positive rate. Anomaly detection becomes harder as θ increases, as anomalies become more common in the dataset, resulting in a lower AUC. Figure 5bshows the EAD with varying θ. Because $s = 1 - \theta$, the EAD shows a downward trend with higher θ.

Sensitivity and Ablation. We assess the importance of CLAISA's adapter component and its sensitivity to the number of parameters N and the anomaly threshold δ. Figure 6b shows the ADE for varying N between 0 and 25 on all benchmark datasets. With $N = 0$, no adapters are used, serving as an ablation of the adapter component. The lowest ADE occurs at $N = 10$, while $N = 0$ increases ADE by up to 20%. For $N > 10$, ADE rises due to challenges in mapping adapters and isolated knowledge hindering generalisation. Figure 5c shows the AUC for CLAISA's anomaly detection on our synthetic dataset. A low δ results in more false positives, while a high δ leads to more false negatives.

We also include an ablation of CLAISA's dual motion and positional embedding components in Table 2. Here, instead of embedding motion and position separately, we combine features into a single embedding vector and feed this into a unified set of transformer layers. In this setup, the adapter component remains unchanged. The

Table 3 Sensitivity to positional discretisation granularity

Resolution	Open sea		
	Gulf of Mexico	Denmark	East Coast
$0.02°$	0.75 ± 0.10	0.83 ± 0.13	0.84 ± 0.11
$0.01°$	0.58 ± 0.03	0.65 ± 0.02	0.62 ± 0.03
$0.005°$	0.64 ± 0.05	0.71 ± 0.04	0.68 ± 0.03
Resolution	Port areas		
	Kattegat	Piraeus	Ushant
$0.02°$	0.78 ± 0.11	0.86 ± 0.08	0.87 ± 0.10
$0.01°$	0.62 ± 0.02	0.72 ± 0.05	0.70 ± 0.06
$0.005°$	0.64 ± 0.03	0.73 ± 0.03	0.72 ± 0.05

We vary the grid resolution (degrees) used for mapping latitude/longitude to grid cells. Results report average ADE

results demonstrate that separating motion and positional embeddings consistently improves forecasting accuracy across all datasets.

Finally, we assess CLAISA's sensitivity to the positional discretisation resolution (latitude and longitude) in Table 3. All other settings are held fixed, and thresholds and any neighbourhood tolerances are scaled to keep the same physical tolerance across resolutions. We find that coarser resolution results in higher error, as the model is trained to accept broader error bounds. This is particularly prominent in port areas, where manoeuvres tend to be finer. A finer resolution, on the other hand, results in a higher entropy with more targets. While this may be desirable in port areas to accurately predict finer movements, it tends to harm performance in deep-sea areas. Future work may investigate varied resolution mappings depending on regional behaviours.

6 Conclusion

Our approach effectively captures maritime behaviours and demonstrates accurate trajectory forecasting with CL, allowing for heightened anomaly detection capabilities. By implementing CL strategies, we find that a trajectory prediction model may better capture evolving behaviours compared to static deep learning approaches. CLAISA performs CL by leveraging a pre-trained transformer architecture with an adapter-based module to capture a variety of positional behaviours. Finally, we demonstrate CLAISA's effectiveness against existing baselines in an evolving maritime environment.

Acknowledgements This research is supported by MBIE Strategic Science Investment Fund (SSIF) Data Science platform - Time-Evolving Data Science / Artificial Intelligence for Advanced Open Environmental Science (UOWX1910).

Author contributions Jack Julian designed and implemented the proposed framework, conducted the experiments, and led the manuscript preparation. Yun Sing Koh and Albert Bifet supervised the research, provided guidance on methodology and analysis, and contributed to the manuscript writing and revisions. All authors reviewed and approved the final version of the manuscript.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Data availability The Automatic Identification System (AIS) datasets used in this study are publicly available from the following sources: Gulf of Mexico and USA East Coast data from NOAA's Marine Cadastre, Danish AIS data from the Danish Maritime Authority, Piraeus dataset from Zenodo, and Ushant dataset. The synthetic dataset generated for anomaly detection benchmarking, along with the implementation code for CLAISA, is openly available on GitHub at https://github.com/jjul482/CLAISA.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Ahmed M, Mahmood AN, Hu J (2016) A survey of network anomaly detection techniques. J Netw Comput Appl 60:19–31. https://doi.org/10.1016/j.jnca.2015.11.016

Aljundi R, Kelchtermans K, Tuytelaars T (2019) Task-free continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11254–11263 . https://doi.org/10.48550/arXiv.1812.03596

Alomari Y (2024) Andó M SHAP-based insights for aerospace PHM: Temporal feature importance, dependencies, robustness, and interaction analysis. Results in Engineering 21:101834. https://doi.org/10.1016/j.rineng.2024.101834

Arguedas VF, Pallotta G, Vespe M (2017) Maritime traffic networks: From historical positioning data to unsupervised maritime traffic monitoring. IEEE Trans Intell Transp Syst 19(3):722–732. https://doi.org/10.1109/TITS.2017.2699635

Authority DM (2024) Danish AIS System. https://www.dma.dk/safety-at-sea/navigational-information/ais-data

Bao P, Chen Z, Wang J, Dai D, Zhao H (2023) Lifelong vehicle trajectory prediction framework based on generative replay. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2023.3300545

Bouritsas G, Daveas S, Danelakis A, Thomopoulos SC (2019) Automated real-time anomaly detection in human trajectories using sequence to sequence networks. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–8 .https://doi.org/10.1109/AVSS.2019.8909844 . IEEE

Buzzega P, Boschini M, Porrello A, Abati D, Calderara S (2020) Dark experience for general continual learning: a strong, simple baseline. Adv Neural Inf Process Syst 33:15920–15930. https://doi.org/10.48550/arXiv.2004.07211

Capobianco S, Millefiori LM, 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. https://doi.org/10.1109/TAES.2021.3096873

Chen C, Zhang D, Castro PS, Li N, Sun L, Li S, Wang Z (2013) iBOAT: Isolation-based online anomalous trajectory detection. IEEE Trans Intell Transp Syst 14(2):806–818. https://doi.org/10.1109/TITS.2013.2238531

Chen H, Wang J, Shao K, Liu F, Hao J, Guan C, Chen G, Heng P-A (2023) Traj-MAE: Masked autoencoders for trajectory prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8351–8362 . https://doi.org/10.1109/ICCV51070.2023.00767

Cover T.M, Thomas J.A Elements of Information Theory. https://doi.org/10.1002/047174882X d'Afflisio E, Braca P, Millefiori LM, Willett P (2018) Detecting anomalous deviations from standard maritime routes using the ornstein-uhlenbeck process. IEEE Trans Signal Process 66(24):6474–6487. https://doi.org/10.1109/TSP.2018.2875887

d'Afflisio E, Braca P, Millefiori LM, Willett P (2018) Maritime anomaly detection based on mean-reverting stochastic processes applied to a real-world scenario. In: 2018 21st International Conference on Information Fusion (FUSION), pp. 1171–1177 . https://doi.org/10.23919/ICIF.2018.8455854 . IEEE

Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. Knowledge Discovery and Data Mining 96:226–231

Forti N, Millefiori LM, Braca P, Willett P (2019) Anomaly detection and tracking based on mean–reverting processes with unknown parameters. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8449–8453. https://doi.org/10.1109/ICASSP.2019.8683428 . IEEE

Forti N, Millefiori LM, Braca P, Willett P (2020) Prediction of vessel trajectories from AIS data via sequence-to-sequence recurrent neural networks. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8936–8940. https://doi.org/10.1109/ICASSP40776.2020.9054421 . IEEE

Gama J, Žliobaitė I, Bifet A, Pechenizkiy M, Bouchachia A (2014) A survey on concept drift adaptation. ACM computing surveys (CSUR) 46(4):1–37. https://doi.org/10.1145/2523813

Gloaguen P, Chapel L, Friguet C (2023) Tavenard R Scalable clustering of segmented trajectories within a continuous time framework: application to maritime traffic data. Mach Learn 112(6):1975–2001. https://doi.org/10.1007/s10994-021-06004-8

Guo S, Mou J, Chen L (2021) Chen P An anomaly detection method for AIS trajectory based on kinematic interpolation. Journal of Marine Science and Engineering 9(6):609. https://doi.org/10.3390/jmse9060609

Hexeberg S, Flåten A.L, Brekke EF, et al (2017) AIS-based vessel trajectory prediction. In: 2017 20th International Conference on Information Fusion (Fusion), pp. 1–8. https://doi.org/10.23919/ICIF.2017.8009762 . IEEE

Ho S, Liu M, Du L, Gao L, Xiang Y (2023) Prototype-guided memory replay for continual learning. IEEE Transactions on Neural Networks and Learning Systems. https://doi.org/10.1109/TNNLS.2023.3246049

Houlsby N, Giurgiu A, Jastrzebski S, Morrone B, De Laroussilhe Q, Gesmundo A, Attariyan M, Gelly S (2019) Parameter-efficient transfer learning for NLP. In: International Conference on Machine Learning, pp. 2790–2799. https://doi.org/10.48550/arXiv.1902.00751 . PMLR

Kawaguchi Y (2018) Anomaly detection based on feature reconstruction from subsampled audio signals. 26th European Signal Processing Conference (EUSIPCO). IEEE, pp 2524–2528. https://doi.org/10.23919/EUSIPCO.2018.8553480

Kazemi S, Abghari S, Lavesson N, Johnson H, Ryman P (2013) Open data for anomaly detection in maritime surveillance. Expert Syst Appl 40(14):5719–5729. https://doi.org/10.1016/j.eswa.2013.04.029

Laxhammar R (2008) Anomaly detection for sea surveillance. In: 2008 11th International Conference on Information Fusion, pp. 1–8. IEEE

Li M, Wang Y, Sun H, Cui Z, Huang Y, Chen H (2023) Explaining a machine-learning lane change model with maximum entropy shapley values. IEEE Transactions on Intelligent Vehicles 8(6):3620–3628. https://doi.org/10.1109/TIV.2023.3266196

Li Y, Wang J, Li T, Fu Z (2024) Traisformer: Spatio-temporal ship trajectory prediction based on transformer. In: 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), pp. 1099–1104 . https://doi.org/10.1109/AINIT61980.2024.10581516 . IEEE

Lu J, Liu A, Dong F, Gu F, Gama J, Zhang G (2018) Learning under concept drift: A review. IEEE Trans Knowl Data Eng 31(12):2346–2363. https://doi.org/10.1109/TKDE.2018.2876857

Ma Q, Du X, Zhang M, Wang H, Lang X, Mao W (2024) A spatial-temporal attention method for the prediction of multi ship time headways using AIS data. Ocean Eng 311:118927. https://doi.org/10.1016/j.oceaneng.2024.118927

Madaan D, Yin H, Byeon W, Kautz J, Molchanov P (2023) Heterogeneous continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15985–15995. https://doi.org/10.48550/arXiv.2306.08593

Ma C, Miao Z, Li M, Song S, Yang M-H (2018) Detecting anomalous trajectories via recurrent neural networks. In: Asian Conference on Computer Vision, pp. 370–382.https://doi.org/10.1007/978-3-030-20870-7_23 . Springer

March D, Metcalfe K, Tintoré J, Godley BJ (2021) Tracking the global reduction of marine traffic during the COVID-19 pandemic. Nat Commun 12(1):1–12. https://doi.org/10.1038/s41467-021-22423-6

Mascaro S, Nicholso AE, Korb KB (2014) Anomaly detection in vessel tracks using bayesian networks. Int J Approximate Reasoning 55(1):84–98. https://doi.org/10.1016/j.ijar.2013.03.012

Nguyen D, Vadaine R, Hajduch G, Garello R, Fablet R (2021) GeoTrackNet-a maritime anomaly detector using probabilistic neural network representation of AIS tracks and a contrario detection. IEEE Trans Intell Transp Syst 23(6):5655–5667. https://doi.org/10.1109/TITS.2021.3055614

Oceanic N, Administration A (2024) Vessel Traffic AIS Data. https://marinecadastre.gov/ais/

Pallotta G, Vespe M, Bryan K (2013) Vessel pattern knowledge discovery from ais data: A framework for anomaly detection and route prediction. Entropy 15(6):2218–2245. https://doi.org/10.3390/e15062218

Pham Q, Liu C, Sahoo D, Steven H (2021) Contextual transformation networks for online continual learning. International Conference on Learning Representations

Pimentel MA, Clifton DA, Clifton L, Tarassenko L (2014) A review of novelty detection. Signal Process 99:215–249. https://doi.org/10.1016/j.sigpro.2013.12.026

Qiang H, Guo Z, Chu Z, Xie S, Peng X (2025) Motion-inspired spatial-temporal transformer for accurate vessel trajectory prediction. Eng Appl Artif Intell 148:110391. https://doi.org/10.1016/j.engappai.2025.110391

Ristic B, Scala L, Morelande B, Gordon M N (2008) Statistical analysis of motion patterns in ais data: Anomaly detection and motion prediction. 11th International Conference on Information Fusion. IEEE, pp 1–7

Shi L, Wang L, Zhou S, Hua G (2023) Trajectory unified transformer for pedestrian trajectory prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9675–9684.https://doi.org/10.1109/ICCV51070.2023.00887

Singh SK, Heymann F (2020) Machine learning-assisted anomaly detection in maritime navigation using AIS data. In: 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), pp. 832–838 . https://doi.org/10.1109/PLANS46316.2020.9109806 . IEEE

Smith JS, Tian J, Halbe S, Hsu Y-C, Kira Z (2023) A closer look at rehearsal-free continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2409–2419. https://doi.org/10.48550/arXiv.2203.17269

Song L, Wang R, Xiao D, Han X, Cai Y, Shi C (2018) Anomalous trajectory detection using recurrent neural network. In: Advanced Data Mining and Applications: 14th International Conference, ADMA 2018, Nanjing, China, November 16–18, 2018, Proceedings 14, pp. 263–277. https://doi.org/10.1007/978-3-030-05090-0_23 . Springer

Tritsarolis A, Kontoulis Y (2021). Theodoridis Y The Piraeus AIS dataset for large-scale maritime data analytics. https://doi.org/10.5281/zenodo.6323416

Vahdat A, Kautz JNVAE (2020) A deep hierarchical variational autoencoder. Adv Neural Inf Process Syst 33:19667–19679

Varlamis I, Tserpes K, Etemad M, Júnior AS (2019) Matwin S A network abstraction of multi-vessel trajectory data for detecting anomalies. EDBT/ICDT Workshops, vol 2019

Wan Z, Chen J, Makhloufi AE, Sperling D, Chen Y (2016) Four routes to better maritime governance. Nature 540(7631):27–29. https://doi.org/10.1038/540027a

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). https://doi.org/10.1109/CVIDL51233.2020.00-89 . IEEE

Wang D, Zhang J, Cao W, Li J, Zheng Y (2018) When will you arrive? Estimating travel time based on deep neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32. https://doi.org/10.1609/aaai.v32i1.11877

Wilbur M, Mukhopadhyay A, Vazirizade S, Pugliese P, Laszka A, Dubey A (2021) Energy and emission prediction for mixed-vehicle transit fleets using multi-task and inductive transfer learning. In: Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part IV 21, pp. 502–517. https://doi.org/10.1007/978-3-030-86514-6_31 . Springer

Yang C-H, Wu C-H, Shao J-C, Wang Y-C, Hsieh C-M (2022)AIS-based intelligent vessel trajectory prediction using bi-LSTM. IEEE Access https://doi.org/10.1109/ACCESS.2022.3154812

Yang B, Fan F, Ni R, Li J, Kiong L, Liu X (2022) Continual learning-based trajectory prediction with memory augmented networks. Knowl-Based Syst 258:110022. https://doi.org/10.1016/j.knosys.2022.110022

Yang J, Bian X, Qi Y, Wang X, Yang Z, Liu J (2024) A spatial-temporal data mining method for the extraction of vessel traffic patterns using AIS data. Ocean Eng 293:116454. https://doi.org/10.1016/j.oceaneng.2023.116454

Yang B, Fan F, Ni R, Wang H, Jafaripournimchahi A, Hu H (2024) A multi-task learning network with a collision-aware graph transformer for traffic-agents trajectory prediction. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2023.3345296

Yu J, Zhuge Y, Zhang L, Hu P, Wang D, Lu H, He Y (2024) Boosting continual learning of vision-language models via mixture-of-experts adapters. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 23219–23230. https://doi.org/10.48550/arXiv.2403.11549

Zhang X, Liu J, Gong P, Chen C, Han B, Wu Z (2023) Trajectory prediction of seagoing ships in dynamic traffic scenes via a gated spatio-temporal graph aggregation network. Ocean Eng 287:115886. https://doi.org/10.1016/j.oceaneng.2023.115886

Zhao L, Shi G (2019) Maritime anomaly detection using density-based clustering and recurrent neural network. The Journal of Navigation 72(4):894–916. https://doi.org/10.1017/S0373463319000031

Zhi C, Sun H, Xu T (2023) Adaptive trajectory prediction without catastrophic forgetting. J Supercomput 79(14):15579–15596. https://doi.org/10.1007/s11227-023-05241-z