Vessel Trajectory Prediction Using Robust AIS Preprocessing and Dual-Self-Attention GRU

Marilena Sinni[1,*,†], Dimitris M. Kyriazanos[1,‡]

[1]National Centre of Scientific Research "Demokritos", Institute of Informatics and Telecommunications (IIT), Patr. Gregoriou E & 27 Neapoleos, Athens, Greece

Abstract

Precise forecasting of vessel movements is crucial for effective maritime traffic monitoring and management. Automatic Identification System (AIS) data offer rich information for trajectory prediction but suffer from irregular sampling, missing values, noise, and transmission errors that complicate modeling. To address these challenges, we propose a two-stage forecasting framework: first, a comprehensive preprocessing pipeline performs data cleaning, feature extraction, anomaly correction, and interpolation to produce consistent trajectories; second, an enhanced GRU network augmented with a dual-head self-attention layer jointly predicts future vessel positions, speed, and course up to 7 hours ahead. We evaluate our method on real AIS data from the port of Brest, France, and demonstrate that it outperforms LSTM, standard GRU, and single-head attention baselines in both loss and error metrics. These results highlight the effectiveness of combining rigorous data denoising with a tailored deep-learning architecture for accurate multi-step vessel trajectory forecasting.

Keywords

AIS data preprocessing, Future trajectory forecasting, Gated Recurrent Units, Self-attention

1. Introduction

Accurate forecasting of vessel trajectories is fundamental to safe and efficient maritime traffic management. With global shipping carrying over 80% of world trade by volume, port authorities and coastal surveillance systems depend on reliable predictions of vessel movements to prevent collisions, optimize traffic flow, and ensure environmental compliance.

The AIS provides the primary data source for these forecasts. The International Maritime Organization (IMO) mandates AIS, which uses VHF transmissions to relay dynamic data (e.g., position, speed, course) and static data (e.g., MMSI, vessel type, voyage info) [1]. Despite its richness, AIS suffers from irregular sampling intervals (from seconds to hours), missing or corrupted records (e.g., spurious land-based "jumps" or implausible speeds), and heterogeneous vessel behaviors (cargo ships, tankers, fishing boats, etc.), all of which pose significant challenges for machine learning models. Many works have analyzed the AIS data errors and proposed preprocessing methods, such as correcting AIS errors [2], simplifying [3] and compressing the vessel trajectory data [4].

Future trajectory forecasting is an active research domain, with a variety of techniques explored for vessel trajectory prediction [5]. Existing methods utilize Recurrent Neural Networks (RNN) [6], bidirectional Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) [7, 8], Transformer [9], and Attention-based architectures [10] for AIS-based future location forecasting, as well as interval-based future location prediction [11], each employing its own preprocessing strategy.

The key challenge is to bridge the gap between the complexities of AIS data preprocessing and the need for accurate future trajectory forecasting. To address this, we propose a rigorous data analysis pipeline alongside a prediction model that estimates both vessel position and kinematic variables -

DARES 2025: AI-driven Data Engineering and Reusability for Earth and Space Sciences, Workshop at the 28th European Conference on Artificial Intelligence (ECAI 2025), 25 October 2025, Bologna, Italy

*Corresponding author

†Main writer of the manuscript.

‡Scientific responsible for the project.

masinni@iit.demokritos.gr (M. Sinni); dkyri@iit.demokritos.gr (D. M. Kyriazanos)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

speed and course - using a GRU architecture enhanced with a dual-head self-attention layer. The main contributions of this paper are summarized as follows:

- A comprehensive preprocessing pipeline for AIS trajectories, incorporating noise filtering, interpolation of missing values, and normalization ensuring high-quality, analysis-ready sequences.
- A robust sequence forecasting model, combining a single-layer GRU with a dual-head self-attention mechanism that enhances prediction accuracy.
- A multi-task output design to jointly predict future vessel position, Speed Over Ground (SOG), and Course Over Ground (COG) up to 7 hours ahead, significantly outperforming LSTM and standard GRU baselines.

1.1. Overview

The remainder of this paper is organized as follows. Section 2 discusses the related work. In section 3 the GRU+self-attention forecasting architecture is described. Experimental setup, evaluation metrics, and results are presented in Section 4. Finally, Section 5 concludes the study and discusses directions for future research.

2. Related work

2.1. AIS data preprocessing

Several works exist in AIS data preprocessing addressing the aforementioned errors. In [4], the authors propose a preprocessing model consisting of data cleaning, trajectory extraction, and compression stages, where the Douglas–Peucker (DP) algorithm is used to reduce data size while preserving key trajectory information. This work keeps the sparse sampling of the dataset that is created after the compression and prepares the dataset for trajectory clustering.

Building on the identification and categorization of AIS data errors, Zhao et al. [2] develop a systematic framework that addresses inaccuracies in three dimensions: physical integrity, spatial logical integrity and time accuracy. For physical integrity, they filter out implausible values, such as latitudes beyond ±90°, unrealistically high speeds over ground, and tracks with too few points or missing static information and correct or discard those records. To improve spatial logical integrity, they detect and eliminate sudden, illogical jumps in vessel position as well as conflicts in MMSI assignment that lead to overlapping tracks or land-crossing anomalies. In the time domain, they address timestamp delays and random resets using kernel density estimation and apply a time-gap threshold (e.g., 10 minutes) to reconstruct consistent trajectories. Through this combination of threshold-based partitioning, statistical correction and association filtering, their method retains over 97% of the original AIS records while markedly improving the quality and usability of trajectory data for maritime traffic monitoring and navigation applications.

Concerning that our goal is to prepare the AIS dataset for tasks such as trajectory prediction or anomaly detection, we reviewed several studies that address both tasks. These approaches typically follow a pipeline that begins with AIS data preprocessing to remove outliers and prepare clean inputs, followed by trajectory prediction of the vessel's next expected position, speed and course, and finally anomaly detection based on deviations from the predicted trajectory. Studies consistently emphasize sanitizing raw inputs by removing invalid records like positions on land, impossible speeds (e.g., <0 knots or >30 knots), incorrect MMSI formats, and null values [12, 13, 14]. Following this core validation, trajectory structuring is applied to define coherent voyages; this involves segmenting data streams using maximum time-gap thresholds (typically 2-4 hours) and discarding tracks deemed too short for meaningful analysis (e.g., <4 hours or fewer than 36 points) [13, 14]. To handle irregular reporting intervals and missing data, temporal regularization is then performed through resampling at fixed intervals (ranging from 1 to 10 minutes) using interpolation methods like inverse distance weighting [12] or linear interpolation [15]. Beyond these foundational steps, some approaches incorporate advanced filtering, such as excluding stationary points (SOG <1 knot), removing statistical outliers based on trajectory variance [12], or repairing implausible kinematic sequences (e.g., unrealistic accelerations or sudden position jumps) [14]. The studies that perform the most analytical preprocessing [12, 14], correcting the most inherent errors of the data, are the works that achieve better results in the prediction step.

2.2. Future location prediction

In recent years, several advancements have been made in the trajectory forecasting problem using NN-based prediction techniques. The authors in [16] propose an extended sequence-to-sequence model based on GRU networks for short-term vessel trajectory prediction using AIS data, demonstrating improved stability over LSTM and standard GRU baselines when forecasting future positions on Yangzi River routes, especially in multi-step scenarios where traditional models tend to lose accuracy. Similarly, Capobianco et al. [6] leverage sequence-to-sequence deep learning models with encoder–decoder RNNs and attention mechanisms to predict vessel trajectories from historical AIS data. Further innovations include Bi-LSTM approaches that combine trajectory denoising with standardized time-series formatting, achieving notable improvements in prediction accuracy [7], and transformer-based models like TrAISformer, which enable long-term forecasting with errors under 10 nautical miles over 10-hour horizons through high-dimensional AIS representations [9]. Another study explores trajectory prediction of sea surface targets using LSTM enhancing prediction accuracy by integrating attention mechanisms to better capture temporal dependencies [10]. In congested port waters, Wang et al. [8] propose a Bi-GRU-based model specifically designed for predicting vessel berthing trajectories. Trained on AIS data from Tianjin Port, their approach demonstrates higher accuracy and lower prediction error compared to traditional RNN models such as LSTM and GRU, highlighting the effectiveness of Bi-GRU in dense and complex maritime environments. Although these works focus solely on position (latitude/longitude) prediction, our interest in this work also includes predicting SOG and COG. Qi et al. in [12] developed a CNN-LSTM model that specifically predicts the vessel's next position, SOG and COG, as a precursor to anomaly detection task, demonstrating the value of multi-feature prediction in maritime applications. Similarly, Zhang et al. [14] proposed a bi-GRU model that predicts position, SOG and COG to support anomaly detection tasks.

Unlike threshold-based filtering and simple moving average (MA) correction, [7], our preprocessing pipeline enforces kinematic constraints (e.g., acceleration limits) and iteratively refines trajectories through physics-aware outlier correction (Section 3.1.3), preserving realistic vessel behavior. While there are recent works that predict SOG/COG along with position, [12, 14], our unified output design forecasts longer horizons (7 hours) and handles COG's circular nature via cosine similarity loss. Additionally, several studies have addressed the trajectory prediction problem using GRU [8], biGRU [8], and attention mechanisms [10]. However, to the best of our knowledge, no prior work has leveraged a GRU equipped with two self-attention heads.

3. Methodology

This study follows a step-by-step methodology to create a solid dataset, clean of outliers and irregularly sampled vessel trajectories and perform efficient next-point trajectory prediction. Our forecasting model is a GRU encoder augmented with a dual-head self-attention layer, trained to predict vessel's next geographic position (latitude and longitude), SOG and COG. This section details both the preprocessing steps applied to the raw AIS data and the architecture of the prediction framework.

3.1. Data preprocessing

Data preprocessing addresses AIS flaws, mentioned in 2, in terms of physical errors, spatial inconsistencies, and temporal inaccuracies.

3.1.1. Data cleaning

Specifically, for each vessel, this work leverages some standard data cleansing operations including: removing duplicate records (i.e., entries with identical timestamps or timestamps less than one second apart), elimination of rows with column data for SOG with a value of 0. Additionally, all points with SOG greater than 30 (outside of normal speed values) and COG outside 0-360 were removed. Also, in order to exploit the static information of the dataset, vessels with ship type out of the official boundaries (20-90), according to [17], were filtered out.

3.1.2. Trajectory extraction

Trajectory extraction groups AIS data by MMSI to form vessel-specific trajectories [4]. Since a single vessel can make multiple passes, it is essential to segment its trajectory accordingly. First, each vessel's points are chronologically sorted, then the time difference between successive records is computed and new trajectory is started whenever this gap exceeds 10 minutes. We also split tracks exceeding 24 hours or containing more than our 24-point minimum, discarding shorter ones as uninformative. Finally, we filter out trajectories with an average SOG below 1 knot to remove stationary vessels, retaining only actively moving tracks.

3.1.3. Speed and position anomaly correction

To ensure the quality of the extracted trajectories, we implement two anomaly correction methods:

Speed outlier correction method prevents unrealistic jumps in SOG. Physically implausible accelerations or decelerations in vessel speed were detected and corrected. Given a maximum allowed acceleration ($a_{\max}$) and deceleration ($d_{\max}$), we enforce the following constraints in each trajectory:

$$\Delta\mathrm{SOG}_i = |\mathrm{SOG}_i - \mathrm{SOG}_{i-1}| \tag{1}$$

$$\Delta\mathrm{SOG}_i \leq \begin{cases} a_{\max} \cdot \Delta t_i & \text{if accelerating} \\ d_{\max} \cdot \Delta t_i & \text{if decelerating} \end{cases} \tag{2}$$

where ($a_{\max}$) and ($d_{\max}$) were fixed in 0.15 knots/s (this value is approximately the average of acceleration for all ships) and $\Delta t_i = t_i - t_{i-1}$ is the time difference in seconds. So, if a violation is detected, the erroneous SOG value is replaced by time-weighted average of its neighbors:

$$\mathrm{SOG}_i^{\text{corrected}} = \frac{\Delta t_i \cdot \mathrm{SOG}_{i-1} + \Delta t_{i-1} \cdot \mathrm{SOG}_{i+1}}{\Delta t_{i-1} + \Delta t_i}$$

Position outlier correction detects unrealistic displacements using kinematic constraints (maximum allowed movement per timestep). The maximum allowed displacement Δd_{max} from point i-1 to i is derived from:

$$\Delta d_{\max} = v_{i-1} \cdot \Delta t_i + \frac{1}{2} a_{\max} \cdot (\Delta t_i)^2 \tag{3}$$

where $v_{i-1} = \mathrm{SOG}_{i-1} \cdot 0.514$ (knots $\rightarrow$ m/s), and $a_{\max} = 0.15$ knots/s $\cdot\, 0.514$ ($\rightarrow$ m/s²).

The actual displacement Δd_i is computed in UTM coordinates (meters):

$$\Delta d_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{4}$$

If $\Delta d_i > \Delta d_{max}$, we further check the next segment $i \rightarrow i+1$ to avoid overcorrection. If both segments violate constraints, the point is flagged for correction. Once an outlier is identified, an interpolation-based method is applied to perform the correction.

$$x_i^{\text{corrected}} = x_{i-1} + \left(\frac{t_i - t_{i-1}}{t_{i+1} - t_{i-1}} \right) (x_{i+1} - x_{i-1}) \tag{5}$$

$$y_i^{\text{corrected}} = y_{i-1} + \left(\frac{t_i - t_{i-1}}{t_{i+1} - t_{i-1}}\right)(y_{i+1} - y_{i-1}) \tag{6}$$

The algorithm operates iteratively to progressively refine data corrections, as single-pass approaches often introduce new anomalies while resolving existing ones. Based on empirical evaluation on the selected dataset, we determined that three iterations provide a balanced trade-off between correction accuracy and computational efficiency.

3.1.4. Resampling

The final preprocessing step involves resampling trajectory points to uniform 1-minute intervals. This converts irregularly sampled data into temporally consistent sequences, helping models learn clearer time-based patterns and reducing noise from variable sampling rates. After segmentation (Section 3.1.2), which limits gaps between points to under 10 minutes, we chose a 1-minute interval, close to the mean sampling rate, to balance data smoothing with movement fidelity. Missing SOG values are recalculated using distance over time between interpolated positions. For COG, angular unwrapping is used to handle 0°/360° discontinuities, allowing smooth linear interpolation before rewrapping to the [0°, 360°) range.

3.2. Trajectory Prediction Framework

Architecture and hyperparameters

This work employs a GRU-based neural network with multi-head attention to predict subsequent trajectory points. The architecture features an input layer, a recurrent layer (GRU) and a dual-head attention block. During training, the model processes normalized sliding windows of vessel trajectories, with inputs containing five temporal features (normalized position, SOG, and COG represented as sine-cosine components) and outputs predicting the subsequent position, speed, and course. The model structure is illustrated in Figure 1.

Input Layer

The input layer ingests fixed-length sequences of normalized trajectory points. Each raw trajectory point $p_t = (x_t, y_t, v_t, \theta_t)$ where x, y are longitude/latitude, v is SOG, and θ is COG, is transformed into a feature vector $p'_t = (x_t, y_t, v_t,\ \cos\psi_t,\ \sin\psi_t)$, where position coordinates (x_t, y_t) are converted to UTM (meters), and along with speed are z-score normalized. Course θ is converted to radians $\psi = \frac{\pi}{180}\theta$ and then mapped to sine-cosine components to avoid discontinuities at $0°/360°$.

Sequential patterns are extracted via overlapping sliding windows of length $T = 5$ over each vessel's trajectory:

$$X_i = \{\tilde{p}_i, \tilde{p}_{i+1}, \tilde{p}_{i+2}, \tilde{p}_{i+3}, \tilde{p}_{i+4}\}, \quad y_i = \tilde{p}_{i+5},$$

with $i = 0, \ldots, M - 6$, where M is the length of the trajectory. Each window shares four points with the previous one, ensuring smooth transitions and maximal use of the trajectory data for predicting the immediate next point.

GRU Layer

The recurrent base consists of a single-layer GRU with hidden size $h = 256$. At each time step t, the GRU updates its hidden state h_t by combining the current input $\tilde{p}_t$ with the previous state h_{t-1}:

$$h_t = \mathrm{GRU}(\tilde{p}_t,\ h_{t-1})$$

It generates full sequences of hidden states $H = \{h_1, h_2, ..., h_T\} \in R^{T\times 256}$, where T = window length. The GRU [18] uses gating mechanisms (update/reset gates) to control information flow, addressing vanishing gradients in standard RNNs. This enables effective modeling of temporal dependencies in vessel trajectories.

Self-Attention Layer

Following the GRU encoder, a dual-head self-attention mechanism [19] refines the sequence representation through contextual aggregation. It operates on the entire GRU output sequence $H \in \mathbb{R}^{T\times 256}$, enabling each timestep to dynamically attend to all others. The mechanism first projects H into combined query, key, and value representations $Q = HW_q,\ K = HW_k,\ V = HW_v,\ W_* \in \mathbb{R}^{256\times 256}$. These projections are split into $k = 2$ heads by reshaping the 256-dimensional features into pairs of 128-dimensional heads, producing $Q^{(k)}, K^{(k)}, V^{(k)} \in R^{T\times 128}$ for each head k. Scaled dot-product attention is computed as $\text{head}^{(k)} = \text{softmax}\left(\frac{Q^{(k)}(K^{(k)})^\top}{\sqrt{d_k}}\right) V^{(k)}$ with $d_k = 128$ for the two heads. The head outputs are then concatenated $[\text{head}^{(1)};\ \text{head}^{(2)}] \in \mathbb{R}^{T\times 256}$ and projected with $W_o \in \mathbb{R}^{256\times 256}$. This output combines with the original GRU states through residual connection and layer normalization:

$$H_{\text{out}} = \text{LayerNorm}\left(H + [\text{head}^{(1)}; \text{head}^{(2)}]W_o^\top + b_o\right),$$

where $b_o \in \mathbb{R}^{256}$ is a bias term, preserving temporal features while enhancing contextual awareness.

Output Heads

The two head outputs are concatenated to restore dimension $h = 256$. The fused representation is fed into three specialized output heads predicting: (1) Position, two-dimensional coordinates in normalized space, (2) Speed (SOG) which is a scalar value, (3) Course (COG) a sine-cosine pair.

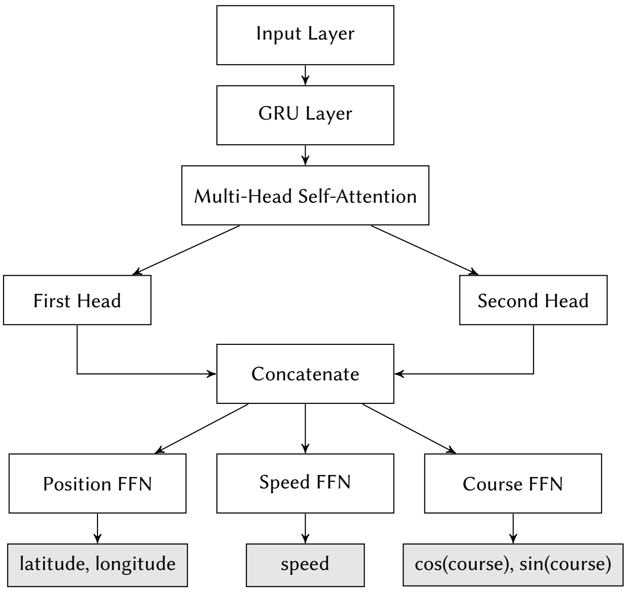


Figure 1: GRU + Multi-Head Attention architecture.

Loss function

The model employs a Mean Squared Error (MSE) loss function, with a modification for the COG component. Specifically, cosine similarity loss is used for COG to account for its circular nature, avoiding discontinuities at the 0°/360° boundary. This approach ensures more stable and accurate predictions of directional angles. The final loss is given by:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (v_i - \hat{v}_i)^2 + \left(1 - (\cos\theta_i \cos\hat{\theta}_i + \sin\theta_i \sin\hat{\theta}_i)\right)\right] \tag{7}$$

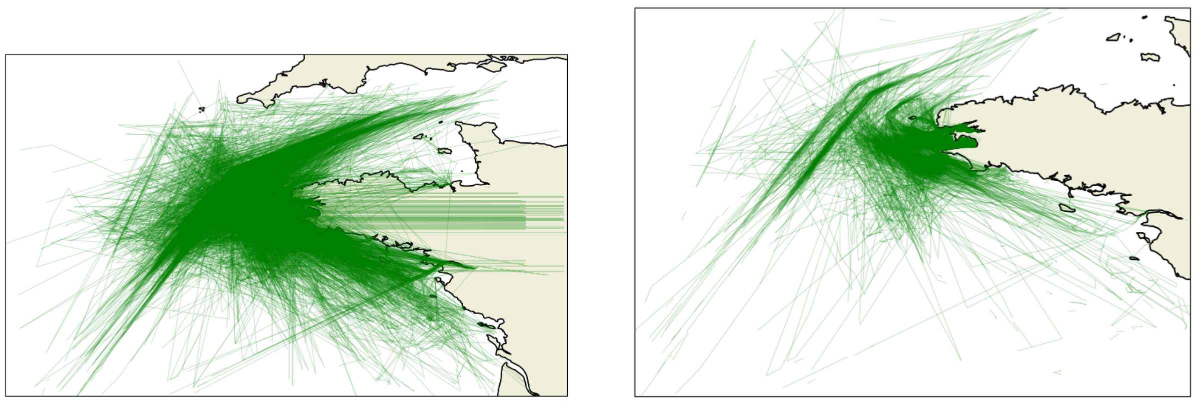
4. Experimental evaluation

4.1. Dataset

For the evaluation of our method, we chose the Brest dataset [17] because it is a publicly available dataset and is the most rich in terms of complexity. The dataset includes both static and dynamic information for various types of vessels, collected by an AIS receiver located in Brest, France, between October 1, 2015, and March 31, 2016 (a six-month period) and are illustrated in Fig. 2a. It comprises 18,657,858 AIS records from 5,041 vessels. The number of signals recorded per vessel ranges from 1 to 1,065,741. The sampling rate varies from less than one second to several days, with an average of approximately 15 minutes and a median of 10 seconds. The Brest dataset contains all types of ships like cargo, tanker, passenger, fishing boats, pleasure crafts, towing and tug, military, and other. The special thing about this dataset is that it contains rare vessel types, such as military and 'other' categories, which are often absent from other AIS datasets and also has extreme sampling irregularity that makes it particularly challenging for comprehensive data preprocessing and evaluating model robustness.

4.2. AIS data preprocessing

In Section 3.1, we outlined our end-to-end preprocessing pipeline covering data cleaning, trajectory segmentation, and temporal resampling. As shown in Figure 2, spatial integrity issues, such as land crossings and illogical position jumps, have been effectively corrected. Table 1 summarizes the results of speed and position anomaly corrections aimed at reinforcing kinematic consistency, while Table 2 details the impact of each cleaning stage on dataset size, trajectory count, and vessel coverage. Trajectory extraction enhances the dataset by partitioning continuous AIS tracks into consistent sub-sequences, thereby increasing the diversity of vessel behavior patterns.



(a) Raw BREST dataset

(b) Preprocessed dataset

Figure 2: Trajectory data before and after preprocessing

The key insights from the table highlight two major stages where a significant number of points were removed. The first stage is during the speed filtering step, which eliminates vessels with either zero SOG or unrealistic values above 30 knots. Most of the removed points had SOG = 0, indicating many stationary vessels or erroneous readings that could have negatively impacted the prediction model. In the second stationary trajectory filtering applied after segmentation, the rest of route segments where vessels remain static were also removed. These steps ensure that the final dataset primarily contains meaningful kinematic information suitable for trajectory prediction. The second stage is at the temporal resampling which uniformly down-samples the data to fixed intervals, smoothing out irregular sampling and improving the stability of downstream prediction models. Another observation from the table is that the extraction procedure resulted in a significant reduction in the number of vessels, while the total number of trajectory points decreased far less proportionally. This suggests that the 10-minute time-gap threshold led to the creation of many short trajectories either single point or fewer than 24 points which were subsequently discarded. As a result, many vessels with dense but uneven sampling were removed from the dataset.

Table 1
Anomaly correction statistics on extracted trajectories

Correction Type	Points Updated	Updated (%)
Speed anomaly correction	1 106 345	13.56%
Position anomaly correction	1 620 114	19.85%

Table 2
Data-cleaning and extraction summary with removal rates

Stage	Points	Removed (%)	Vessels / Trajectories
Initial	19 035 630	—	5 055 vessels
After Deduplication	18 026 004	2.93%	5 055 vessels
After COG Filtering	18 025 812	0.00%	5 055 vessels
After SOG Filtering	9 389 527	45.37%	4 982 vessels
After Ship-Type Filtering	8 617 489	4.06%	4 891 vessels
Extracted Trajectories	8 159 865	5.31%	31 136 traj./ 1 681 vessels
After Stationary Removal	6 947 719	14.86%	25 710 trajectories
After 1-min Resampling	1 737 625	75.00%	25 660 traj./ 1 661 vessels

4.3. Evaluation Setup - Parameter Selection

This work implements and trains the model using the PyTorch framework over 200 training epochs, with early stopping (patience=20 epochs) to prevent overfitting. The model optimization employs the Adam optimizer with a learning rate of 0.001 and a step decay scheduler (gamma=0.1 at every 10 epochs). This configuration promotes stable convergence during training while mitigating gradient-related issues such as explosion or vanishing gradients. We employed a batch size of 128 and the dataset was split into 70% for training, 10% for validation, and 20% for testing and finally, k-fold cross-validation was used to ensure the robustness and generalizability of the model across different subsets of the data.

To evaluate our model's predictive performance, we apply it to the test set, which is preprocessed in the same way as the training data. For each trajectory, we slide a window of length w over the sequence: at each step i (where $i = w + 1, \ldots, n$), the model observes the previous w points and predicts the i-th point. We then compare this prediction to the true value at time i to obtain an instantaneous loss L_i. By sliding the window across all n points, we accumulate these losses and compute the average prediction error as follows:

$$L = \frac{1}{n - w} \sum_{i=w+1}^{n} L_i$$

where n is the total number of points in the trajectory and w is the window size.

To evaluate the selection of GRU with dual-head self-attention layer as the best model, we experimented with other methods that are usually used to solve such time-series forecasting tasks. The models tested are LSTM, GRU without attention layer and also GRU with a single-head attention layer to justify the usefulness of the second head.

4.4. Results

In this section, we present the results of the prediction framework. Table 3 shows a detailed comparison of different models to identify the best-performing one, and on Table 4 the rest of the performance measurements of the best performing model are presented. While the enhanced MSE loss was used during training, it proved insufficient on its own for evaluating model performance. Therefore, we also report additional metrics, including the mean, median, and maximum distance errors across the entire dataset, to provide a more comprehensive evaluation.

Table 3
Comparison of model performance based on prediction loss and distance errors

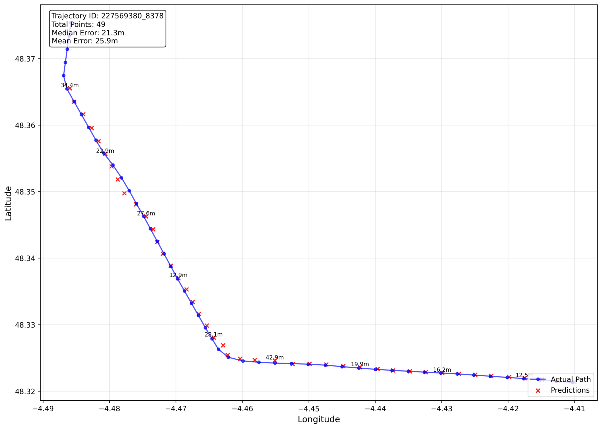
Model	Total Loss	Median Error (m)	Avg Error (m)	Max Error (m)
LSTM	0.0158	98.5	154.0	7341.6
GRU	0.0157	86.3	131.8	8796.3
GRU+Single-head Attention Layer	0.0157	63.1	97.4	7198.9
GRU+Dual-head Attention Layer	**0.0156**	**58.5**	**90.4**	**6327.0**

Table 4
SOG and COG Performance Metrics of the Best Model (GRU+Dual-head Attention Layer)

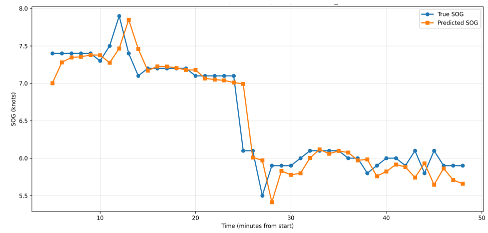
Metric	Value
Mean SOG Error (knots)	0.6
Mean COG Error (degrees)	10.8

Self-attention layer allows the model to dynamically re-weight past time steps, enabling it to capture long range dependencies and abrupt maneuvering behaviors that purely recurrent layers may overlook. By computing pairwise affinities across the entire sequence, self-attention can emphasize critical context—such as an earlier course change or speed adjustment when predicting the next point. In our ablation (Table 3), adding a single attention head already yields a substantial drop in median error (from 86.3 m to 63.1 m) by focusing on the most informative time steps. Splitting into two parallel heads further boosts performance reducing interference between these distinct modalities. This dual-head design leads to the lowest overall loss and best error statistics (median 58.5 m, average 90.4 m), demonstrating that two parallel attention heads significantly improve trajectory prediction accuracy.

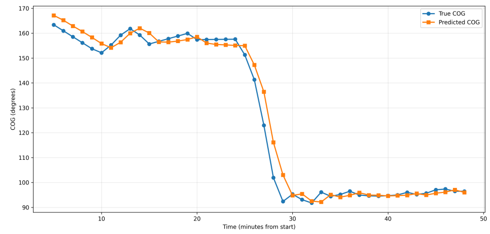
For further evaluation, we select two vessels with different total prediction errors as examples. Figure 3 shows the experimental results for one vessel with an average distance error of 25.9 m. Specifically, Figure 3a shows the ground truth represented by a blue line and the predicted points marked as red crosses, and the other two figures 3b, 3c illustrate the SOG and COG predictions, compared with the ground truth. Also, Figure 4 shows the comparison charts for another vessel with an average error of 26.2 m. This vessel is particularly interesting because, from the position and SOG plots, we observe that it remains stationary for 155 minutes; despite this, the prediction accuracy remains strong for up to 208 minutes, with the longest successful prediction extending to 7 hours. An analytical error distribution for the two vessels is shown in Figures 5 and 6, respectively.



(a) Latitude and longitude position prediction

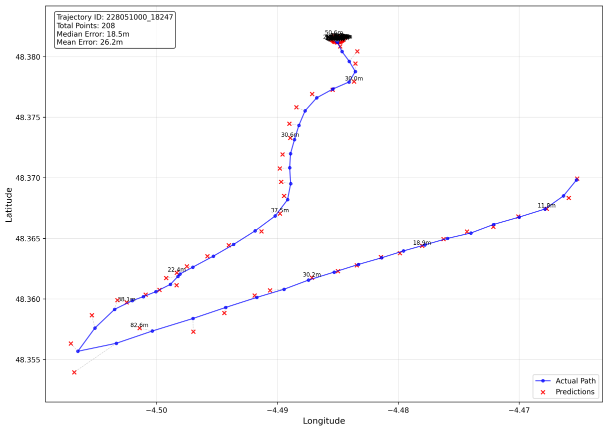


(b) SOG prediction

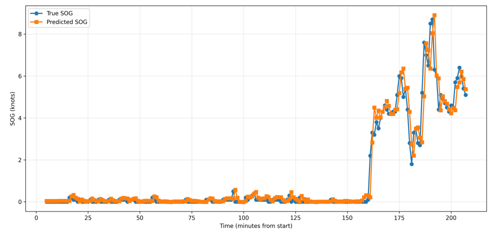


(c) COG prediction

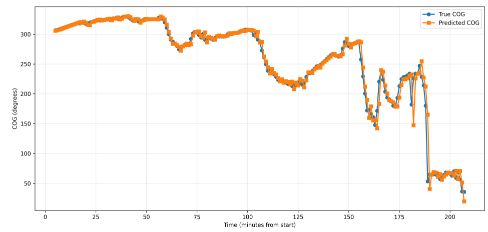
Figure 3: Cargo vessel trajectory with 49 points length



(a) Latitude and longitude position prediction



(b) SOG prediction



(c) COG prediction

Figure 4: Tug vessel trajectory with 208 points

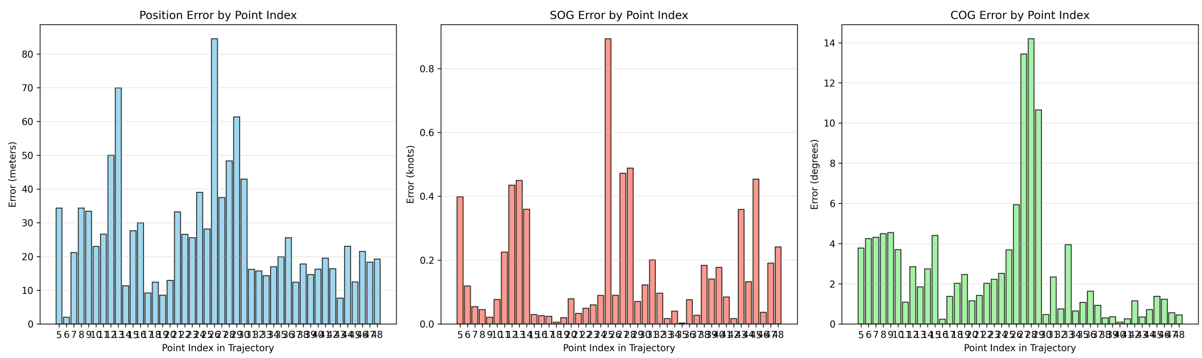


Figure 5: Position, SOG, COG prediction error for the cargo vessel

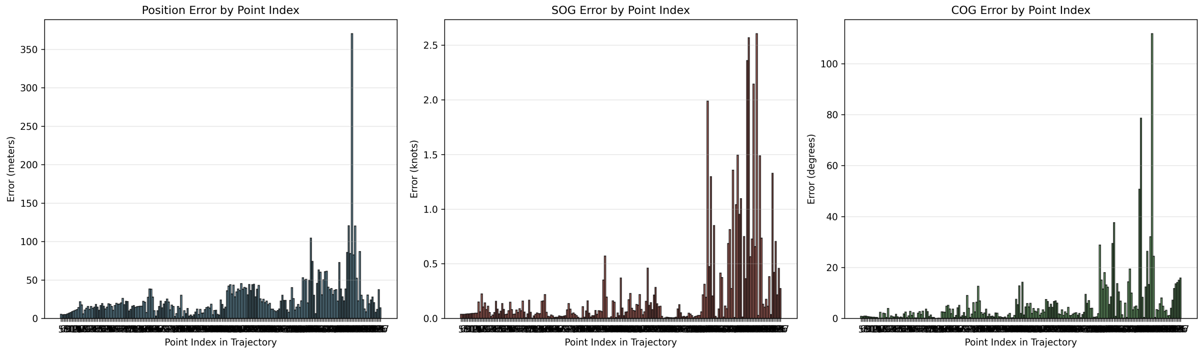


Figure 6: Position, SOG, COG prediction error for the tug vessel

5. Conclusion and Future Work

In this study, we presented a vessel trajectory prediction framework that combines robust preprocessing of AIS data with a GRU-based model augmented by a dual-head self-attention layer. Using historical AIS records from the port of Brest, France, our experiments demonstrate that:

- The proposed preprocessing pipeline successfully handles a heterogeneous dataset containing diverse vessel types. By filtering spurious points, interpolating gaps, and normalizing features, we reduce noise and produce smooth, analysis-ready trajectories.
- The GRU + dual-head self-attention architecture reliably forecasts future positions up to more than 7 hours ahead. In addition to spatial coordinates, it jointly predicts SOG and COG with strong accuracy, as evidenced by consistently low median and average error metrics.
- Incorporating two parallel attention heads, yields the best overall performance, reducing interference between modalities and improving both loss and error statistics relative to single-head and purely recurrent baselines.

In future work, our goal is to validate the generality of our approach on additional AIS datasets from different regions and traffic conditions to prove its robustness. Another goal is to compare our forecasting performance against state-of-the-art methods in trajectory prediction benchmarks, and specifically bidirectional RNNs and transformers, to quantify relative gains. Furthermore, we intend to use this model as a foundation to extend to anomaly detection: by comparing live AIS streams against our predicted trajectories, speeds, and headings, we aim to identify unusual vessel behavior in real time.

Acknowledgments

This work has received funding from the European Defence Fund programme under grant agreement No 101103386, FaRADAI project. The views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

Special thanks to the IIT group MagCIL and in particular research associates Christos Nikou and Christos Sgouropoulos for their scientific support and comments.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to: Grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Z. Yan, Y. Xiao, L. Cheng, R. He, X. Ruan, X. Zhou, M. Li, R. Bin, Exploring ais data for intelligent maritime routes extraction, Applied Ocean Research 101 (2020) 102271.

[2] L. Zhao, G. Shi, J. Yang, Ship trajectories pre-processing based on ais data, The Journal of Navigation 71 (2018) 1210–1230.

[3] Z. Wei, X. Xie, X. Zhang, Ais trajectory simplification algorithm considering ship behaviours, Ocean Engineering 216 (2020) 108086.

[4] I. P. N. Hartawan, I. M. O. Widyantara, A. Karyawati, N. I. Er, K. B. Artana, N. P. Sastra, Ais data pre-processing for trajectory clustering data preparation, in: 2021 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES), IEEE, 2021, pp. 1–5.

[5] X. Zhang, X. Fu, Z. Xiao, H. Xu, Z. Qin, Vessel trajectory prediction in maritime transportation: Current approaches and beyond, IEEE Transactions on Intelligent Transportation Systems 23 (2022) 19980–19998.

[6] S. Capobianco, L. M. Millefiori, N. Forti, P. Braca, P. Willett, Deep learning methods for vessel trajectory prediction based on recurrent neural networks, IEEE Transactions on Aerospace and Electronic Systems 57 (2021) 4329–4346. doi:10.1109/TAES.2021.3096873.

[7] C.-H. Yang, C.-H. Wu, J.-C. Shao, Y.-C. Wang, C.-M. Hsieh, Ais-based intelligent vessel trajectory prediction using bi-lstm, IEEE Access 10 (2022) 24302–24315. doi:10.1109/ACCESS.2022.3154812.

[8] C. Wang, H. Ren, H. Li, Vessel trajectory prediction based on ais data and bidirectional gru, in: 2020 International conference on computer vision, image and deep learning (CVIDL), IEEE, 2020, pp. 260–264.

[9] D. Nguyen, R. Fablet, A transformer network with sparse augmented data representation and cross entropy loss for ais-based vessel trajectory prediction, IEEE Access 12 (2024) 21596–21609. doi:10.1109/ACCESS.2024.3349957.

[10] T. Yu, Y. Zhang, S. Zhao, J. Yang, W. Li, W. Guo, Vessel trajectory prediction based on modified lstm with attention mechanism, in: 2024 4th International Conference on Neural Networks, Information and Communication Engineering (NNICE), IEEE, 2024, pp. 912–918.

[11] E. Chondrodima, N. Pelekis, A. Pikrakis, Y. Theodoridis, An efficient lstm neural network-based framework for vessel location forecasting, IEEE transactions on intelligent transportation systems 24 (2023) 4872–4888.

[12] Y. Qi, J. Yang, A. Bai, J. Liu, A method for real-time detection of vessel abnormal behavior based on cnn-lstm, Expert Systems with Applications (2025) 128303.

[13] Z. Xie, X. Bai, X. Xu, Y. Xiao, An anomaly detection method based on ship behavior trajectory, Ocean Engineering 293 (2024) 116640.

[14] B. Zhang, K. Hirayama, H. Ren, D. Wang, H. Li, Ship anomalous behavior detection using clustering and deep recurrent neural network, Journal of Marine Science and Engineering 11 (2023) 763.

[15] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, R. Fablet, Geotracknet—a maritime anomaly detector using probabilistic neural network representation of ais tracks and a contrario detection, IEEE Transactions on Intelligent Transportation Systems 23 (2021) 5655–5667.

[16] L. You, S. Xiao, Q. Peng, C. Claramunt, X. Han, Z. Guan, J. Zhang, St-seq2seq: A spatio-temporal feature-optimized seq2seq model for short-term vessel trajectory prediction, IEEE Access 8 (2020) 218565–218574. doi:10.1109/ACCESS.2020.3041762.

[17] C. Ray, R. Dréo, E. Camossi, A.-L. Jousselme, C. Iphar, Heterogeneous integrated dataset for maritime intelligence, surveillance, and reconnaissance, Data in brief 25 (2019) 104141.

[18] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).