Measurement Science
and Technology

ACCEPTED MANUSCRIPT

LCTA-Net: A Deep Fusion Model for Multi-Scenario Vessel Trajectory Prediction

To cite this article before publication: Baiheng Wang *et al* 2025 *Meas. Sci. Technol.* in press https://doi.org/10.1088/1361-6501/ae08d2

LCTA-Net: A Deep Fusion Model for Multi-Scenario Vessel Trajectory Prediction

Baiheng Wang[1], Zhenkai Zhang[*,1], Boon-Chong Seet[2], Fuqiang Zeng[1] and Jielong Lu[1]

1: Ocean College, Jiangsu University of Science and Technology, Zhenjiang, 212003, China

2: Department of Electrical and Electronic Engineering, Auckland University of Technology, Auckland 1010, New Zealand

*Corresponding author(s). E-mail(s): zhangzhenkai@just.edu.cn

Other author(s). E-mail(s): 1461697359@qq.com

boon-chong.seet@aut.ac.nz

3197418739@qq.com

1429096527@qq.com

Abstract: Vessel trajectory prediction tasks currently face several challenges, including the diversity of behavioral patterns, interference from anomalous data, and limited generalization capabilities of existing models. These challenges are particularly pronounced when dealing with multi-pattern trajectory characteristics, where achieving both high prediction accuracy and robust cross-scenario adaptability remains difficult. To address these issues, this paper proposes a high-precision vessel trajectory prediction framework. Firstly, the Local Outlier Factor (LOF) algorithm is applied to eliminate anomalies from raw Automatic Identification System (AIS) data, thereby improving data quality. The Douglas–Peucker simplification and the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) trajectory clustering algorithm are then used to construct datasets characterized by different turning behaviors. Subsequently, a prediction model called Local-Causal Temporal Attention Network (LCTA-Net) is developed, based on a pruned Transformer encoder. LCTA-Net incorporates a temporal residual encoding module and a causal feature-augmented attention mechanism. These components jointly improve the model's capacity to model temporal dependencies within trajectory sequences. In addition, a fixed-window local neighborhood search strategy is introduced to improve the spatial continuity and physical feasibility of the predicted trajectories. Finally, experimental results on two constructed trajectory datasets demonstrate that the proposed method significantly outperforms several state-of-the-art models across multiple error metrics, confirming its superior prediction accuracy and strong cross-scenario generalization capability.

Keywords: Vessel trajectory prediction; AIS; trajectory clustering; LCTA-Net; local neighborhood search

1 Introduction

Maritime transportation plays a pivotal role in global trade, accounting for nearly 90% of the world's cargo volume [1]. To enhance navigational safety, the International Maritime Organization (IMO) introduced the Automatic Identification System (AIS) in the 1990s, mandating its installation on all types of vessels to facilitate real-time acquisition of positional information [2]. The AIS system has since accumulated a vast amount of historical vessel trajectory data. The openness and availability of these data have fueled research progress in various domains of maritime transportation, including vessel trajectory classification [3], abnormal navigation behavior detection [4], and vessel collision risk assessment [5]. Of particular relevance to this study is the great potential of AIS data in vessel trajectory prediction [6][7], as it provides up-to-date dynamic measurements such as position, speed, heading angle, and course over ground. In earlier studies on vessel trajectory prediction, machine learning approaches have demonstrated distinctive advantages in modeling such dynamic measurements in a data-driven manner. Jiang L *et al*. [8] proposed the DGSTAN model, which integrates graph convolutional networks with spatiotemporal attention mechanisms, enabling accurate modeling of long-range dependencies in traffic flow prediction tasks. This approach validates the significant contribution of graph structures and dynamic weight allocation in enhancing the generalization capability of trajectory modeling. Maskooki et al. [9] employed the k-nearest neighbors (k-NN) algorithm in combination with dynamic time warping (DTW) to extrapolate future trajectory states by matching the spatiotemporal features of historical trajectory segments, thereby completing missing dynamic position measurements. Abebe M *et al.* [10] utilized a Random Forest (RF) ensemble approach to capture nonlinear relationships through a multi-tree voting mechanism, incorporating static vessel attributes such as draft and gross tonnage along with real-time ocean current data to enhance speed estimation accuracy in complex voyage segments. Bei T *et al.* [11] proposed the VIG model, a multimodal trajectory prediction framework based on VectorNet and generative adversarial networks (GAN). By introducing an interaction modeling mechanism among vehicles, lanes, and maps, the model effectively enhances the realism and diversity of generated trajectories, offering a feasible paradigm for multi-agent behavior modeling in complex navigation environments. Liu J *et al.* [12] enhanced Support Vector Regression (SVR) by developing a multi-output least squares SVR (LSSVR) model, which enabled real-time trajectory prediction through kernel-based mapping and a sliding window update mechanism.

With the rapid advancement of deep learning technology, deep neural networks have emerged as a prominent research direction due to their powerful capability to capture complex spatiotemporal features. Yang C H *et al.* [13] proposed a method that integrates data denoising with a bidirectional long short-term memory (Bi-LSTM) network. The AIS data were preprocessed through trajectory separation, outlier removal, and normalization, and the experimental results demonstrated that the proposed method achieved lower prediction error compared to traditional models such as exponential smoothing and support vector regression. Wang L *et al.* [14] proposed the CNN-LSTM-Attention model, which further demonstrated in tide displacement prediction tasks that the attention mechanism can significantly enhance the model’s adaptability to non-stationary time series and strengthen the dynamic correlation modeling capability between historical states and future trajectories. Zhang Y *et al.* [15] introduced the PESO model, which is based on a sequence-to-sequence (Seq2Seq) framework. It enhances spatial correlation modeling by employing parallel encoders—including a position encoder and a voyage state encoder—and incorporating semantic position vectors to guide the decoding process. In recent years, the Transformer architecture has been increasingly applied to prediction tasks due to its advantage in capturing long-range dependencies [16][17]. Jiang D *et al.* [18] proposed the TRFM-LS model, which embeds LSTM module within the Transformer framework. By combining the temporal modeling strength of LSTM with the global dependency modeling capacity of the Transformer's self-attention mechanism, the model significantly reduces prediction error—by over 40% compared to traditional methods. Huang F *et al.* [19] proposed the Dual Path Spatio-Temporal Attention Network (DualSTMA), which separates dynamic and static attributes for independent processing. A gating mechanism is employed to adjust the contribution of neighboring vessels' features. This method achieves superior prediction accuracy in multi-vessel interaction scenarios compared to existing approaches. Collectively, these studies have advanced the application of deep learning to vessel trajectory prediction, each from a distinct technical perspective.

Clustering combined with prediction has proven effective in enhancing model performance by providing more targeted inputs through pattern recognition and classification of historical trajectories. This approach improves both the accuracy of trajectory estimation and measurement stability under diverse navigational behaviors. For instance, Murray B *et al.* [20] developed a behavior-aware prediction framework by applying Gaussian Mixture Models (GMM) in conjunction with Kalman filtering to segment and cluster vessel trajectories. In an experiment conducted in the Singapore Strait, their method achieved a 28% improvement in prediction accuracy compared to traditional approaches. Zhu M *et al.* [21] incorporated domain knowledge from nautical experts to classify encounter scenarios using a probabilistic model and a hybrid predictor, resulting in a 19.3% reduction in 10-minute prediction error relative to a single LSTM baseline in the Yangtze River Estuary. Park J *et al.* [23] introduced spectral clustering to capture nonlinear trajectory patterns and optimized a Bi-LSTM model based on the longest common subsequence (LCSS) distance. Their method reduced positional error by 35% compared to conventional LSTM models in multi-vessel interaction scenarios at Busan Port. These studies optimize feature inputs using clustering priors, significantly enhancing the adaptability of prediction models to complex navigational scenarios and providing trajectory prediction solutions with scenario adaptability for large-scale spatiotemporal measurement systems.

Compared to trajectory prediction methods that rely solely on deep models, the synergistic integration of clustering and prediction demonstrates stronger generalization capabilities in constructing physical constraints, adapting to complex navigational dynamics, and enhancing robustness against anomalous data and distributional shifts. However, existing methods still exhibit significant limitations when confronted with challenges such as behavioral diversity, frequent noise interference, and cross-scenario adaptability. On the one hand, most clustering approaches rely on static heuristics or handcrafted features, making them unstable in handling scenarios where multiple trajectory types coexist in dynamic environments. On the other hand, the decoupled design of clustering and prediction models lacks a trainable coordination mechanism, limiting the effective utilization of contextual information. Moreover, existing studies often overlook the modeling of continuous dynamic features such as speed over ground and course over ground, thus failing to fully capture the vessel's motion evolution. These issues become particularly prominent in real-world maritime environments characterized by frequent multi-vessel interactions and high environmental variability, highlighting the urgent need for more efficient and robust prediction frameworks. To address these challenges, this study proposes a trajectory prediction framework that integrates clustering techniques with deep neural networks. The main contributions of this work are as follows:

To address these challenges, this paper proposes a trajectory prediction framework that integrates clustering techniques with deep neural networks. Specifically, behaviorally consistent trajectory subsets are extracted through structural simplification and morphological clustering. Based on these subsets, a deep prediction network is constructed that combines local causal attention with temporal modeling capabilities. This design enhances prediction accuracy and stability across diverse scenarios and behavioral patterns. The main contributions of this work are as follows:

1) To address the issue of anomalous noise in AIS data, the Local Outlier Factor (LOF) algorithm is employed to remove outlier points. In addition, the Douglas–Peucker (DP) simplification and HDBSCAN clustering algorithms are applied to perform trajectory structure simplification and morphological clustering, resulting in two distinct trajectory datasets characterized by different turning behaviors.

2) A novel integrated prediction model, LCTA-Net (Local-Causal Temporal Attention Network), is proposed. Built upon a pruned Transformer encoder, the model incorporates a temporal residual encoding module and a causal feature-augmented attention module. This architecture combines global attention modeling with local temporal dynamics perception, enabling more effective capture of spatiotemporal evolution patterns and behavioral dependencies within trajectories.

3) To enhance the spatial coherence and local consistency of the generated trajectories, a local neighborhood search constraint mechanism is introduced. This strategy effectively mitigates common issues such as jump points and outliers in high-dimensional classification-based trajectory outputs.

This study targets representative measurement scenarios such as maritime safety and surveillance, focusing on key challenges in complex dynamic systems including uncertainty suppression, structural behavior modeling, and physically consistent prediction. An end-to-end measurement modeling framework is developed, integrating data cleansing and predictive inference into a unified process. The proposed clustering-enhanced prediction paradigm is not only applicable to vessel trajectory forecasting but also offers generalizable insights for other high-dynamic measurement systems such as traffic flow analysis and Industrial Internet-of-Things (IIoT) monitoring. Furthermore, this approach breaks away from the conventional paradigm of "static modeling + independent inference" commonly found in traditional measurement workflows. By integrating clustering-driven data structure reconstruction with a trainable, physically-aware predictive model, the method provides a unified framework for dynamic measurement tasks. This fusion yields a generalizable and interpretable solution with significant theoretical value and practical potential for engineering applications.

The rest of this paper is organized as follows. Section 2 presents the proposed trajectory prediction framework, including the data preprocessing procedures, the architecture of the trajectory prediction model, and the local neighborhood search mechanism. Section 3 presents the simulation experiments and analyzes the results of the proposed framework. Section 4 concludes the paper by summarizing the main contributions and outlining some directions for future work.

2 Proposed Trajectory Prediction Framework

2.1 Framework Overview

This study aims to develop a high-precision vessel trajectory prediction method tailored to complex water environments, leveraging the spatiotemporal characteristics of vessel movements. As illustrated in Fig. 1, the overall framework consists of four main stages: trajectory preprocessing, turning angle-based data partitioning, trajectory prediction, and comparative experiments. In the preprocessing stage, the raw trajectories are first denoised and simplified using outlier detection and the Douglas–Peucker algorithm, thereby improving data quality while preserving key movement features. Subsequently, the HDBSCAN density-based clustering algorithm is applied to automatically group trajectories with similar motion patterns, resulting in structurally consistent trajectory clusters. Building upon this, the dataset is further divided into two subsets according to the turning angle variation patterns of vessel trajectories. This partitioning enables the model to perform behavior-specific learning for different navigation modalities, thereby significantly enhancing prediction accuracy and cross-scenario adaptability.

In the prediction stage, trajectory data undergo feature transformation through three key steps: discrete index conversion, trajectory token embedding, and positional encoding. These steps respectively achieve spatial structuring, semantic mapping, and the incorporation of temporal information, providing a unified spatiotemporal representation for downstream prediction.

The core prediction model adopts the proposed LCTA-Net architecture, which integrates the temporal modeling capabilities of LSTM, a local-context-aware QKV encoder based on one-dimensional causal convolution, and a multi-head attention mechanism. This design enables simultaneous modeling of both local dynamic variations and global spatiotemporal dependencies in vessel trajectories. To further refine prediction accuracy, a local neighborhood search strategy is introduced during the decoding phase for fine-grained optimization of predicted positions. Specifically, a set of candidate positions is first generated based on the current prediction, followed by the construction of a scoring function that incorporates historical trajectory features and vessel motion constraints. The optimal candidate is then selected as the corrected prediction result. To evaluate the effectiveness of the proposed model, multiple state-of-the-art deep learning and temporal modeling approaches are selected as baselines for comparative experiments, focusing on prediction accuracy. This allows for a comprehensive assessment of model performance under complex trajectory prediction scenarios. Finally, to enhance the interpretability and visual presentation of the results, the predicted trajectories are converted into image representations, facilitating an intuitive evaluation of their continuity and accuracy.

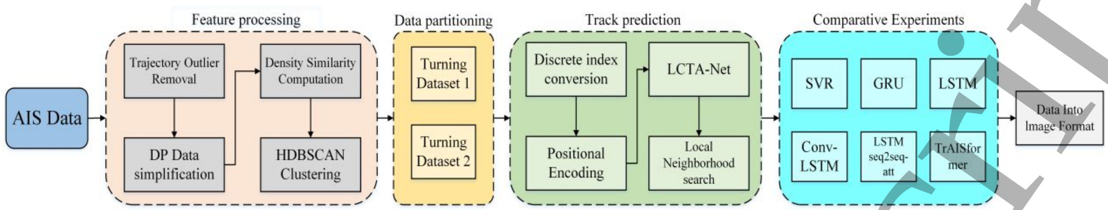


Fig. 1. Flowchart of the vessel trajectory prediction method

2.2 Feature Preprocessing

2.2.1 LOF-Based Outlier Elimination

AIS data inevitably contain noise and outlier points during the collection process. If left unprocessed, these anomalous trajectories may interfere with clustering performance by blurring cluster boundaries, thereby reducing the accuracy of trajectory prediction. To address this issue, the Local Outlier Factor (LOF) algorithm [22] is employed in this study for outlier detection. LOF is a density-based anomaly detection method. Its core principle is that if a point's local density is significantly lower than that of its neighboring points, it is more likely to be an outlier. Let the raw trajectory be denoted as $\mathrm{T}=\{x_1, x_2, ..., x_\mathrm{n}\}$ .The LOF score for each point x_i is calculated using the following formula:

$$LOF_k(x_i) = \frac{1}{|N_k(x_i)|} \sum_{x_j \in N_k(x_i)} \frac{lrd_k(x_j)}{lrd_k(x_i)} \tag{1}$$

Here, $N_k(x_i)$ denotes the *k*-nearest neighbor set of trajectory point x_i and $lrd_k(\cdot)$ represents its local reachability density. A point is identified as an outlier and removed if its LOF score exceeds 1.5, indicating that it lies in a relatively low-density region and deviates from the normal trajectory clusters. This process effectively cleans and optimizes the trajectory dataset.

2.2.2 Vessel Trajectory Clustering and Pattern Extraction Method

Trajectory clustering analysis is an effective means of identifying vessel motion patterns, which facilitates neural networks in extracting and learning features from similar trajectory behaviors. In this study, the Douglas–Peucker (DP) algorithm is first applied to simplify raw trajectories. By recursively retaining critical points that deviate significantly from simplified line segments, the algorithm effectively removes redundant points and achieves trajectory simplification. Specifically, the DP algorithm constructs a baseline between the start and end points of a trajectory and calculates the perpendicular distance from each intermediate point to this baseline. If the maximum distance exceeds a predefined threshold, the corresponding point is retained, and the trajectory is segmented and processed recursively. Otherwise, only the endpoints are preserved. To balance shape preservation and simplification efficiency, the predefined threshold is set to 0.01, corresponding to a maximum allowable spatial deviation of approximately 1.1 kilometers, which ensures that the essential shape of the trajectory remains intact. On the basis of the simplified trajectories, the HDBSCAN algorithm is employed for density-based clustering of trajectory points. HDBSCAN computes density relationships between data points using the concepts of core distance and mutual reachability distance. For any given point, its core distance is defined as the distance to its *k*-th nearest neighbor, formulated as:

$$core_k(p) = d(p, p^{(k)}) \tag{2}$$

where, $p^{(k)}$ denotes the *k*-th nearest neighbor of point p ,and $d(\cdot)$ represents the Euclidean distance.

The mutual reachability distance between any two points a and b is defined as:

$$d_{mreach}(a,b) = \max\{core_k(a), core_k(b), d(a,b)\} \tag{3}$$

This distance effectively reduces the influence of isolated points and locally sparse regions on the clustering results. Based on the mutual reachability distance, HDBSCAN constructs a weighted complete graph over the set of points and extracts the Minimum Spanning Tree (MST) using Kruskal's algorithm. On the resulting MST, HDBSCAN progressively removes edges with higher weights to generate a bottom-up hierarchical clustering structure. Within this hierarchy, the *stability* of each candidate cluster is defined by its persistence across varying density scales. The stability is calculated as follows:

$$Stability(C) = \int_{\lambda_{birth}}^{\lambda_{death}} |C_\lambda| \, d\lambda \tag{4}$$

Here, $\lambda = \frac{1}{d_{mreach}}$, C_λ denotes the set of points contained in cluster C at density scale λ . Based on the principle of maximum stability, HDBSCAN ultimately selects the optimal clustering result and automatically identifies noise points. The algorithm involves two key parameters: minimum cluster size and minimum samples. The minimum cluster size defines the smallest allowable cluster size—larger values tend to preserve denser and more representative clusters, while smaller values may result in more points being labeled as noise. The minimum samples parameter specifies the density criterion for core points, indicating the minimum number of neighboring samples required within a local region. In this study, we set the minimum cluster size to 6 and minimum samples to 4. The detailed parameter selection process and corresponding analysis for both the HDBSCAN and DP algorithms are provided in Section 3.1.2.

2.3 Tokenized Spatiotemporal Representation of Vessel Motion

In vessel trajectory prediction tasks, it is challenging to represent complex dynamic patterns directly within the low-dimensional raw feature space. To address this, this study adopts a sparse vector representation method based on discrete binning [24], implementing a Four-Hot Vector encoding scheme. For each AIS observation point $x_t \in \mathbb{R}^4$ (Longitude, Latitude, SOG, COG), the continuous variables are discretized into N_{lon} (Longitude), N_{lat} (Latitude), N_{sog} (SOG), and N_{cog} (COG) bins, respectively, and independently encoded using one-hot encoding. The resulting discretized representation vector is defined as:

$$h_t \triangleq \left[1_t^{lat}, 1_t^{lon}, 1_t^{sog}, 1_t^{cog}\right]^T \in \{0,1\}^{N_{lat}+N_{lon}+N_{sog}+N_{cog}} \tag{5}$$

Here, 1_t^{α} denotes the one-hot encoding of feature $\alpha \in \{lat, lon, sog, cog\}$ after discrete binning.

Each one-hot vector is mapped to a high-dimensional dense vector space and represented as an embedding vector $e_t^{\alpha} \in \mathbb{R}^{d_\alpha}$, which is retrieved from an embedding matrix via a lookup operation. All embedding vectors are then concatenated to form the final unified trajectory representation:

$$e_t = e_t^{lat} \| e_t^{lon} \| e_t^{sog} \| e_t^{cog} \tag{6}$$

This representation maps the original four-dimensional input into a high-dimensional embedding space $\mathbb{R}^d$, where the total embedding dimension is given by $d = d_{lat} + d_{lon} + d_{sog} + d_{cog}$, To prevent overfitting due to the high-dimensional embedding, the value space of e_t is constrained by fixing the number of discrete bins, resulting in a discrete combinatorial structure. This constraint not only improves the expressiveness of the feature representation but also reinforces regularization.

2.4 Transformer-Based Multi-Stage Prediction Model

The proposed LCTA-Net architecture is illustrated in Fig. 2. It consists of four main components: an input embedding layer, a positional encoding module, a stack of encoder blocks, and a prediction layer. The model takes the discrete four-dimensional tuple features of AIS trajectories as input. These are first encoded through independent embedding layers that map each discrete variable into a unified high-dimensional vector space. Learnable positional encodings are then added to incorporate temporal information. The embedded feature sequence is subsequently fed into a stack of three LCTA blocks, which progressively extract spatiotemporal dynamics from the trajectory. Unlike the original Transformer architecture, this model removes the standard decoder module. Instead, a linear classification head is applied directly to the encoder output to predict the next-step trajectory state. This design is motivated by two key considerations: (1) In trajectory prediction, both the input and output belong to the same modality space, so there is no need for the encoder-decoder alignment mechanism used in traditional Transformers; (2) By incorporating causal convolution-based attention, the model structurally enforces a unidirectional information flow, relying solely on current and past information for representation and prediction, thereby aligning with the temporal causality inherent in trajectory evolution.

The internal structure of each LCTA block incorporates two key modifications to the standard Transformer encoder. First, a unidirectional Long Short-Term Memory (LSTM) network is introduced to process the input sequence, enabling the extraction of dynamic features along the temporal dimension. The LSTM output is fused with the original input via a residual connection and serves as the input to the subsequent attention mechanism. In the attention computation stage, the model employs three separate one-dimensional causal convolution (Causal Conv1D) operations to generate the Query, Key, and Value representations. This replaces the conventional linear projections used in standard Multi-Head Self-Attention, ensuring that each position's representation depends only on the current and preceding states. A multi-head self-attention mechanism is then used to aggregate contextual features across multiple subspaces, enhancing the model's capacity to capture long-range dependencies within the sequence. The attention output is further transformed through layer normalization and a feed-forward network to produce the final feature representation. By integrating temporal modeling, causal control, and global interaction, this architecture significantly improves the model's ability to represent the evolutionary patterns of trajectory sequences.

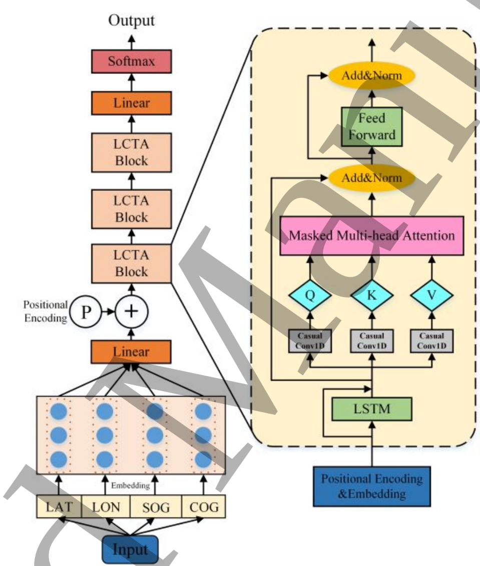


Fig. 2. Structure of LCTA-Net

2.4.1 Positional Encoding

Since the Transformer architecture does not inherently encode sequence order, it cannot distinguish the relative structure between time steps unless positional information is explicitly injected. This issue becomes especially pronounced after discrete representations—such as Four-Hot vectors—are applied, as the resulting input tokens lack inherent temporal order. To ensure the model's ability to learn sequential dependencies, an external positional encoding mechanism is required to provide temporal context. To address this, a learnable positional embedding scheme is introduced at the input stage. Each time step is assigned a unique, trainable vector to explicitly encode its position within the trajectory sequence. Let the trajectory length be T, and let each embedded trajectory point at time t be denoted by $e_t \in \mathbb{R}^d$. Then, a unique positional vector $p_t \in \mathbb{R}^d$ is assigned to each position $t \in [0, T-1]$, where:

$$p_t = \text{Embedding}(t) = P[t] \tag{7}$$

Here, $P \in \mathbb{R}^{T\times d}$ denotes a trainable parameter matrix that is updated jointly with the main network parameters during model training. The final input sequence representation is obtained by performing element-wise addition of the trajectory embeddings and their corresponding positional embeddings at each time step:

$$\%= e_t + p_t \tag{8}$$

Compared to the fixed sinusoidal positional encoding method used in the original Transformer architecture [18], this study adopts a learnable positional encoding scheme. This approach allows the model to automatically adapt to the temporal patterns of different trajectory segments during training, enabling more flexible and effective capture of both local and global positional features within the trajectory sequence.

2.4.2 Temporal Residual Encoding Module

After the discrete encoding and embedding of AIS trajectory data, a high-dimensional representation sequence is obtained that integrates spatial and motion-related information such as longitude, latitude, speed, and course. Following positional encoding, this sequence possesses an explicit temporal order, making it suitable for input into a temporal modeling module. To effectively capture dynamic evolution patterns and behavioral trends between trajectory points, this study designs a temporal module based on the LSTM network. On top of this, cross-layer residual connections are introduced to enhance the model's ability to retain high-dimensional trajectory information. This design improves the stability and representational capacity of trajectory prediction in complex maritime environments. The structure is illustrated in Fig. 3.

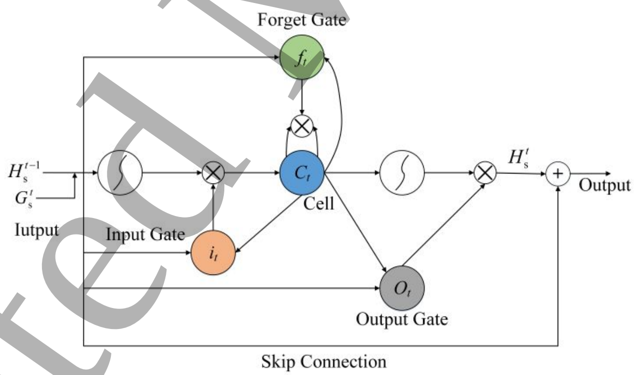


Fig. 3. Structure of the Temporal Residual Encoding Module

LSTM is a type of recurrent neural network capable of modeling long-term dependencies. Its core innovation lies in the use of gating mechanisms, which regulate the flow and retention of information across time steps, making it well-suited for handling non-stationary sequence modeling tasks. Let $G_s^t \in \mathbb{R}^d$ denote the embedded representation of the trajectory sequence at time step t and $H_s^{t-1} \in \mathbb{R}^h$ represent the hidden state from the previous time step. The state update process of the LSTM is defined as follows:

$$Z_s^t = G_s^t \oplus H_s^{t-1} \tag{9}$$

$$f_t = \sigma(W_f Z_s^t + b_f) \tag{10}$$

$$i_t = \sigma(W_i Z_s^t + b_i) \tag{11}$$

$$o_t = \sigma(W_o Z_s^t + b_o) \tag{12}$$

$$\overline{c}_t = \tanh(W_c Z_s^t + b_c) \tag{13}$$

$$C_t^s = f_t \odot C_s^{t-1} + i_t \odot \tilde{c}_t \tag{14}$$

$$H_s^t = o_t \odot \tanh(C_s^t) \tag{15}$$

In the above computations, $\sigma(\cdot)$ denotes the sigmoid activation function. The operator $\odot$ denotes element-wise (Hadamard) multiplication, whereas $\oplus$ represents the concatenation operation, which merges the current input and the previous hidden state along the feature dimension. The learnable weight matrices associated with the gating units are $W_f, W_i, W_o, W_c \in \mathbb{R}^{h\times(d+h)}$, and the corresponding bias vectors are $b_f, b_i, b_o, b_c \in \mathbb{R}^{h}$. In conventional LSTM architectures, the output is entirely generated through internal state transitions. However, when processing high-dimensional discrete trajectory embeddings, such models may suffer from excessive compression of input information and degradation of temporal dependencies across time steps. This issue is particularly pronounced when handling complex navigation segments or abrupt behavioral transitions, where critical features may be lost. To mitigate these limitations, a structural enhancement mechanism is introduced. Specifically, a direct skip connection is added between the input layer and the output of each LSTM unit. This enables cross-layer fusion by combining the input feature Z_s^t at each time step with its corresponding output state, resulting in the final representation:

$$\tilde{H}_s^t = H_s^t + Z_s^t \tag{16}$$

This structure preserves the dynamic features modeled by the LSTM while introducing a static feature path constructed from the original embedding information. By enabling cross-temporal fusion, it helps to maintain the structural integrity of the trajectory input.

2.4.3 Causal Feature-Augmented Attention

In standard multi-head self-attention mechanisms, the Query (Q), Key (K), and Value (V) vectors are typically obtained by applying linear projections independently to each time step of the input sequence. However, this pointwise projection approach fails to capture temporal context, making it difficult to effectively model the continuity and local dynamics between adjacent points in trajectory data. This limitation becomes especially problematic when trajectory points are densely distributed or subject to small perturbations, as the resulting attention distributions can be easily disrupted, thereby reducing prediction accuracy. To address this issue, a causal convolutional structure is introduced within the multi-head attention mechanism to replace the conventional linear projections used to generate Q, K, and V representations. This design explicitly incorporates local temporal structure and causal constraints during the feature generation stage. The improved architecture is illustrated in Fig. 2. Let the input sequence be denoted as $X = [x_1, x_2, ..., x_T] \in \mathbb{R}^{\{T\times d\}}$, where T is the sequence length and d is the feature dimension at each time step. The model applies three separate one-dimensional convolutional kernels—W_Q 、 W_K and W_V —to map the input sequence X and obtain the corresponding Q , K and V representations:

$$Q=\text{Conv1D}_{\text{causal}}(X, W_Q) \tag{17}$$

$$K=\text{Conv1D}_{\text{causal}}(X, W_K) \tag{18}$$

$$V=\text{Conv1D}_{\text{causal}}(X, W_V) \tag{19}$$

Here, $Conv1D_{causal}$ denotes a convolution operation with causal padding, which ensures that the output at each time step depends only on the current and preceding time points. This enforces a unidirectional flow of information and prevents any leakage of future information.

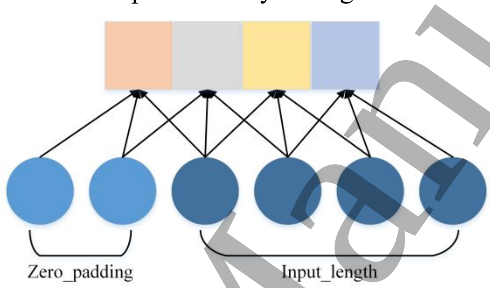


Fig. 4. Structure of Causal Convolution

The receptive field of the convolution operation can be adjusted by configuring the kernel size and the depth of the stacked structure, thereby enhancing the model's ability to capture short- and mid-range dynamic patterns in trajectories. Compared to standard linear transformations, causal convolution preserves the temporal order while extracting structural variations between adjacent trajectory points, improving the attention mechanism's responsiveness to local patterns. As shown in Fig. 4, the causal convolution structure applies zero-padding in the forward direction to maintain the original sequence length, effectively capturing feature dependencies within a sliding temporal window.

2.5 Local Neighborhood Search for Output Refinement

To enhance the spatial plausibility and local consistency of the generated trajectories, a local neighborhood search constraint mechanism is designed during the model inference phase. This mechanism addresses common issues associated with high-dimensional classification logits at the output layer, such as abrupt spatial jumps, outliers, or physically unreachable trajectory points. By dynamically pruning the output space, the method enables effective trajectory refinement. Specifically, during the step-by-step sampling of future trajectory points, the original trajectory points—already converted into multi-dimensional one-hot representations through discretization—are transformed into token sequences via the embedding module. The model outputs a conditional distribution $p\left(e_t \middle| e_{0:t-1}\right)$, where the predicted longitude and latitude are represented as logits vectors z_t^{lon}, and z_t^{lat}, respectively. Given that the model may generate bin values in the high-dimensional output space that are far from the current trajectory state, this study introduces an output truncation strategy based on spatial index neighborhoods at each time step.

$$N_t^{\alpha} = \left\{ i \middle| \left| i - i_t^{\alpha} \right| \le r, i \in \left[0, N_{\alpha}\right) \right\}, \quad \alpha \in \{lon, lat\} \tag{20}$$

where, i_t^{α} denotes the discrete index corresponding to the current position at time step t, r is the predefined neighborhood radius, and N_t^{α} represents the set of candidate output bins. The logits vector is truncated as follows:

$$\tilde{z}_t^{\alpha}[i] = \begin{cases} z_t^{\alpha}[i], & if\ i \in N_t^{\alpha} \\ -\infty, & otherwise \end{cases} \tag{21}$$

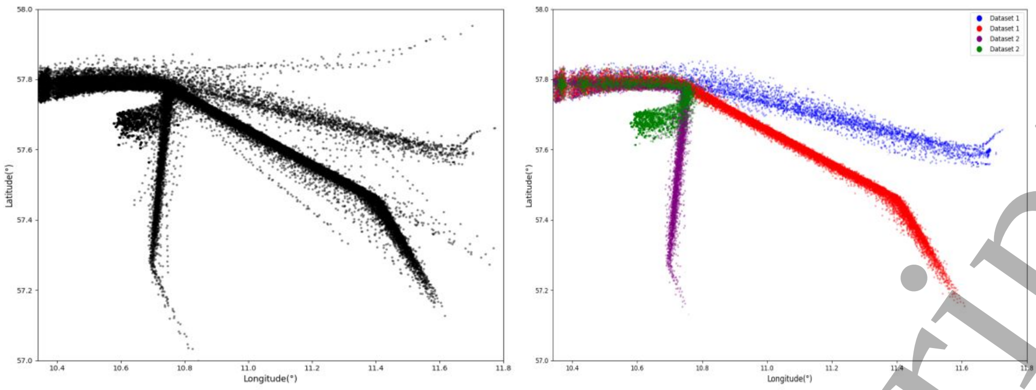
After the softmax operation, the probabilities corresponding to positions outside the defined neighborhood converge to near zero, thereby forcing the model to sample only within the spatial neighborhood of the current position. This mechanism effectively introduces a dynamic spatial prior based on the current state during the sampling process. As a result, it helps prevent the generation of discontinuous, implausible, or abrupt trajectory points, without the need to explicitly model kinematic constraints.

3 Evaluation

3.1 Experimental settings

3.1.1 Dataset

The AIS dataset used in this study was published by the Danish Maritime Authority (DMA) [30], covering navigation records of cargo vessels and tankers operating within a specified maritime area from January 1 to March 31, 2019. The region of study is geographically bounded between latitudes 57.0°N and 58.0°N and longitudes 10.3°E and 11.8°E. A visualization of the trajectories within this region is shown in Fig. 5. The data preprocessing workflow—comprising outlier removal, trajectory simplification, and clustering analysis—is detailed in Section 2.2. After multiple rounds of cleaning and processing, a total of 899 high-quality vessel trajectories were obtained. These trajectories are then clustered based on their morphological characteristics, as illustrated in Fig. 5. Trajectories that lacked significant motion patterns were identified as noise and excluded. Based on the magnitude of turning angle variations in the clustering results, the trajectories in Fig. 5(b) are divided into two subsets. Dataset 1 includes the green and purple clusters, containing 468 trajectories with prominent turning behavior—some segments exhibiting turning angles greater than 90°. Dataset 2 consists of the red and blue clusters, comprising 431 trajectories that are overall smoother with relatively low curvature variation.



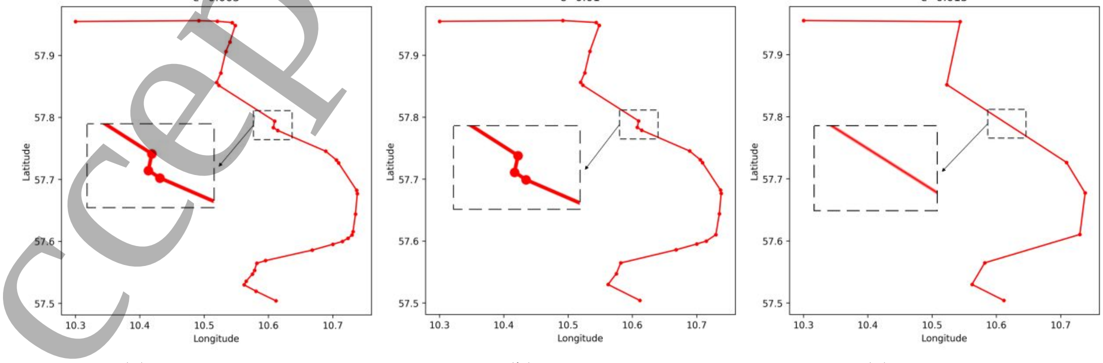
(a) Raw Trajectories (b) Preprocessed Trajectories

Fig. 5. Trajectory Visualization Results

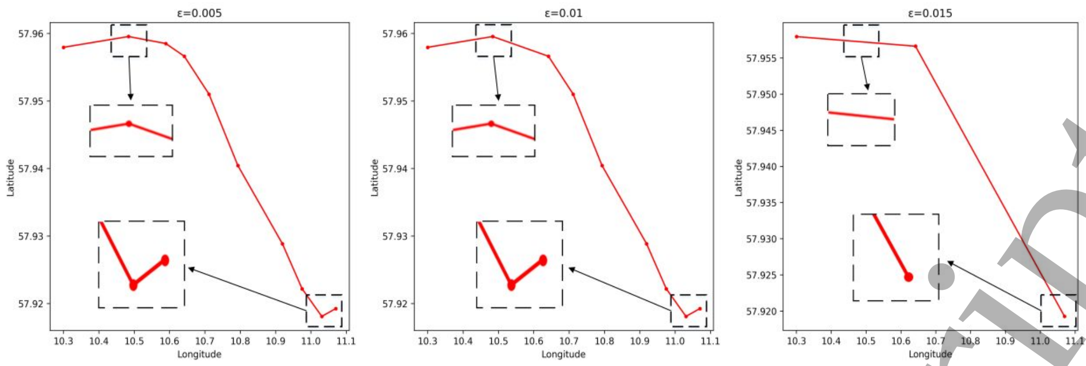
Additionally, the AIS data in this study were uniformly resampled at 10-minute intervals—a commonly adopted setting in the maritime domain. This configuration effectively reduces the impact of redundant data while preserving the dynamic characteristics of vessel movements. During preprocessing, all trajectories were aligned to this unified time step to ensure temporal consistency and comparability across the dataset.

3.1.2 Parameter Sensitivity Analysis for Clustering and Simplification Thresholds

In the Douglas–Peucker algorithm, the parameter ε controls the maximum perpendicular deviation between trajectory points and the simplified line segments. Its value is typically set between 0.001 and 0.05. As ε increases, the number of retained key points in the trajectory gradually decreases, effectively compressing the data scale and reducing the computational burden of the model. In this study, considering the spatial distribution of the AIS data (latitude 57°–58°, longitude 10.3°–11.8°) and the characteristic vessel maneuvers involving noticeable turns within a 1-kilometer spatial scale, ε is set to 0.01. This threshold approximately corresponds to a real-world distance of 1.1 kilometers, enabling the preservation of key turning points while filtering out local jitters and sampling noise. As a result, the trajectory shape is well preserved while reducing data volume. To validate the suitability of $\varepsilon = 0.01$ in the DP algorithm, two representative trajectories were selected for visual comparison under candidate thresholds of 0.005, 0.01, and 0.015 (as shown in the Fig. 6), in order to assess the impact of different parameter settings on trajectory simplification performance.



(a) $\varepsilon = 0.005$ (b) $\varepsilon = 0.01$ (c) $\varepsilon = 0.015$



(d) $\varepsilon = 0.005$ (e) $\varepsilon = 0.01$ (f) $\varepsilon = 0.015$

Fig. 6 Trajectory simplification results: (a)-(c) show the visualization of different threshold values applied to representative trajectory 1; (d)–(f) illustrate the results for representative trajectory 2 under the same threshold settings.

As shown in the Fig. 6, when ε is set to a larger value, the simplification becomes more efficient; however, some critical trajectory shape information is distorted, and certain identifiable turning points are noticeably lost. When $\varepsilon = 0.01$, the simplification effectively removes redundant points while preserving the overall structural integrity of the trajectory. Therefore, the threshold is set to 0.01 in this study.

To determine the optimal values for the two key parameters of the HDBSCAN clustering algorithm, namely the min cluster size and min samples, this study adopts a combined evaluation approach using the Silhouette Coefficient (SC) [25] and the Davies–Bouldin Index (DBI) [26]. A grid search strategy is employed to quantitatively analyze clustering quality under different parameter combinations. The SC measures the cohesion of samples within their assigned cluster and their separation from other clusters, while the DBI evaluates the inter-cluster separability and intra-cluster compactness. Their definitions are as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{22}$$

$$DBI = \frac{1}{K}\sum_{i=1}^{K} \max_{j \neq i}\left(\frac{\sigma_i + \sigma_j}{d_{ij}}\right) \tag{23}$$

where, $a(i)$ denotes the average distance between sample i and other samples within the same cluster, $b(i)$ represents the average distance between sample i and samples in the nearest neighboring cluster, and K is the number of clusters. Additionally, σ_i indicates the average distance of samples within cluster i to the cluster centroid, while d_{ij} refers to the distance between the centroids of clusters i and j. The mean value of $s(i)$ across all samples constitutes the SC value, which ranges from –1 to 1; higher values indicate better clustering performance. Conversely, a lower DBI value signifies superior clustering quality.

To unify the scales of the SC and DBI and achieve an integrated evaluation, this study applies linear normalization to both metrics and uses their difference as a composite scoring function for clustering performance:

$$Score = \frac{SC - \min\left(SC\right)}{\max\left(SC\right) - \min(\mathrm{SC})} - \frac{DBI - \min\left(DBI\right)}{\max\left(DBI\right) - \min(DBI)} \tag{24}$$

Here, a larger Score indicates that the parameter combination achieves an optimal balance between clustering compactness and separation.

Based on the scale of the dataset, the range of min cluster size was set from 2 to 11, and min samples from 2 to 18. This range covers a variety of clustering scenarios, from small and sparse clusters to moderately dense ones, allowing the model to accommodate the structural diversity of trajectory data while reducing the risk of falling into local optima during parameter selection. To intuitively illustrate how clustering performance varies under different parameter combinations, we generated a heatmap of the evaluation scores, as shown in Fig. 7.

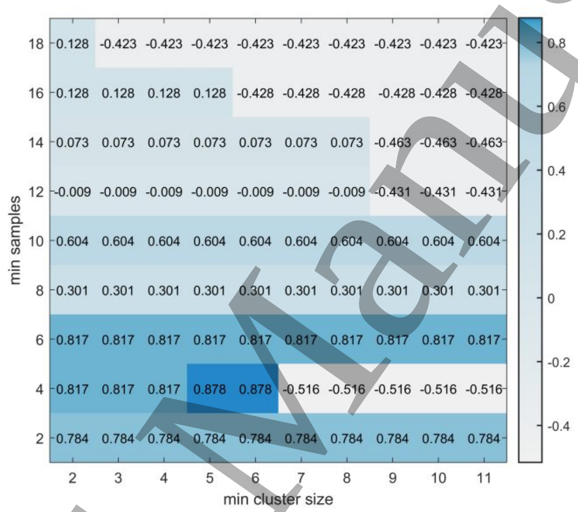


Fig. 7 Heatmap of clustering performance scores under different HDBSCAN parameter combinations

As shown in the heatmap of score distributions, the parameter combinations (4, 5) and (4, 6) both fall within the highest scoring range. However, considering that a larger min cluster size value tends to preserve more representative and densely populated clusters while effectively suppressing small-cluster noise, we ultimately selected min samples = 4 and min cluster size = 6 as the parameter configuration for HDBSCAN.

3.1.3 Experimental environment and hyperparameter settings

The proposed method is implemented in a Python 3.9 environment using the PyTorch 1.13.1+cu116 deep learning framework. All computations were performed on an NVIDIA RTX 4090 GPU platform. The model architecture consists of three LCTA blocks (each containing eight self-attention heads) and LSTM networks, where each LSTM layer contains 32 hidden units. These components are jointly used to model the spatial and temporal dynamics of trajectories. The local neighborhood search mechanism introduced in LCTA-Net constrains the spatial receptive field within a radius-r neighborhood to enhance local representation capacity and suppress redundant computation.

The proposed model takes as input a spatiotemporal sequence of historical vessel states, with each time step comprising four features: longitude, latitude, speed over ground (SOG), and course over ground (COG). Longitude and latitude are normalized to the range [0, 1] based on the geographic scope of the study area; SOG is normalized within the range of 0–30 knots, and COG within 0–360 degrees. The initial input sequence consists of 18 time steps, corresponding to 3 hours of historical trajectory data. During the prediction phase, the model generates future vessel states in an autoregressive manner based on the input sequence, thereby achieving continuous trajectory prediction.

During training, a multi-task cross-entropy loss function was used, targeting four prediction components: position, speed, and course. The AdamW optimizer was employed in conjunction with a learning rate schedule combining linear warm-up and cosine decay. The initial learning rate was set to 0.0006. To improve generalization, weight decay was applied during parameter updates with a decay coefficient of 0.1. All experiments used a batch size of 32 and were trained for 80 epochs. The model parameters achieving the best performance on the validation set were selected for final testing.

3.1.4 Evaluation metrics

To evaluate the model's performance in the trajectory point prediction task, three categories of prediction error metrics are employed: Minimum Distance Error (MDE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). Both MDE and RMSE are calculated based on spherical distance metrics, taking the curvature of the Earth into account. The Haversine formula is used to compute the spatial deviation between predicted and ground truth points, and the results are reported in kilometers (km), reflecting the model's accuracy at a geographical scale. In contrast, MAE is used to assess coordinate-wise prediction deviations separately in the longitudinal and latitudinal directions. This metric is expressed in degrees (°) and captures the model's positional offset characteristics in coordinate space.

$$Haversine(p_1, p_2) = 2r\arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos\phi_1\cos\phi_2\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \tag{25}$$

$$MDE = \frac{1}{N}\sum_{i=1}^{N}\min_k\left(Haversine(\hat{p}_i, p_i)\right) \tag{26}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Haversine(\hat{p}_i, p_i))^2} \tag{27}$$

$$MAE_{\mathrm{lon/lat}} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i - y_i\right| \tag{28}$$

Here, $p_i = (\phi_i, \lambda_i)$ denotes the ground truth latitude and longitude, $\hat{p}_i$ represents the predicted coordinates, r is the Earth's radius (6371 km), N is the total number of trajectory points, and

y_i and $\hat{y}_i$ refer to the true and predicted longitude, and latitude values, respectively. Specifically, MDE reflects the model's ability to produce the most accurate prediction among multiple output branches, effectively capturing its best-case prediction capability. RMSE measures the overall spatial error stability of the model's output and is more sensitive to large deviations. In contrast, MAE quantifies directional errors along the longitude and latitude axes, highlighting the model's offset tendencies across different spatial dimensions.

3.2 Comparison with existing methods

In this study, we conduct a systematic evaluation of several mainstream deep learning models on two datasets characterized by distinct trajectory patterns. The evaluated models include Informer [28], Autoformer [29], LSTM, Conv-LSTM, LSTM-seq2seq-attention [30],and TrAISformer [27]. To ensure the fairness of the comparative experiments, grid search was employed to tune the hyperparameters of all baseline models. All models were trained and evaluated under a unified data split and evaluation metrics, with the optimal configurations selected accordingly. The hyperparameter settings of each baseline model are listed in Table 1.

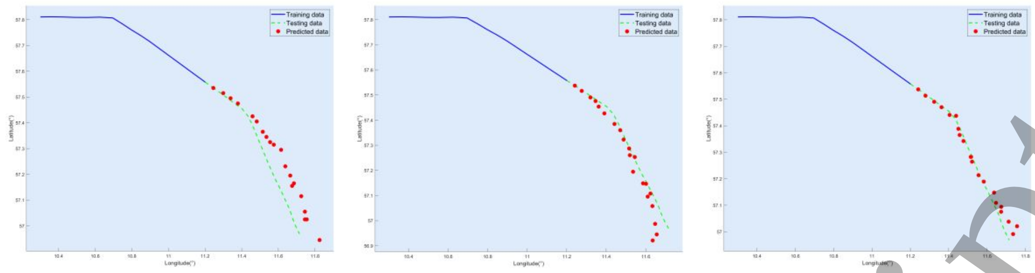
Table 1 Hyperparameter Configurations of Baseline Models

Model	Max epoch	Batch size	Hidden size	Encoder layer	decoder layer	Learning rate	Optimizer
LSTM	60	16	128	--	--	0.001	Adam
Conv-LSTM	60	32	128	--	--	0.001	Adam
LSTM-seq2seq-att	60	32	64	3	2	0.001	Adam
Informer	50	32	--	4	3	0.0005	Adamw
Autoformer	50	32	--	3	2	0.0005	Adamw
TrAISformer	50	32	--	8	--	0.0005	Adamw

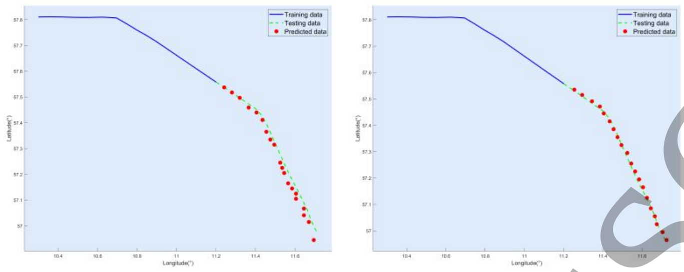
All baseline models adopt a cosine decay strategy for dynamic learning rate adjustment during training, in order to enhance convergence efficiency and overall performance. After determining the optimal hyperparameters for each model, we comprehensively evaluate the effectiveness and robustness of the proposed approach in single-vessel but complex turning scenarios through trajectory visualization, comparison across multiple quantitative error metrics, and box plot analysis of average prediction errors.

3.2.1 Prediction Performance Analysis in Stable Trajectory Scenarios

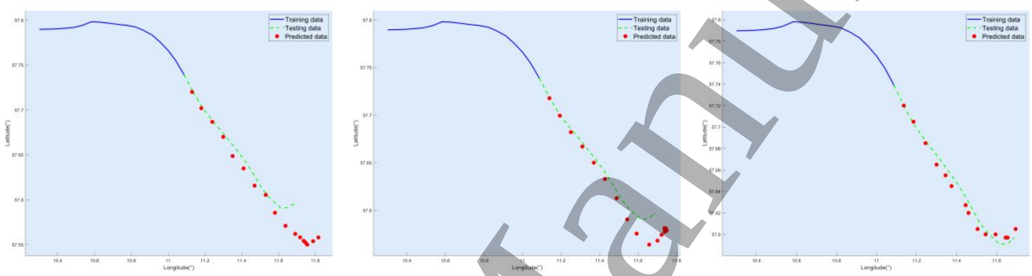
In the visualization analysis, we first selected two representative trajectories from Dataset 1 to visually compare the performance of five models over the 3-hour future prediction interval.



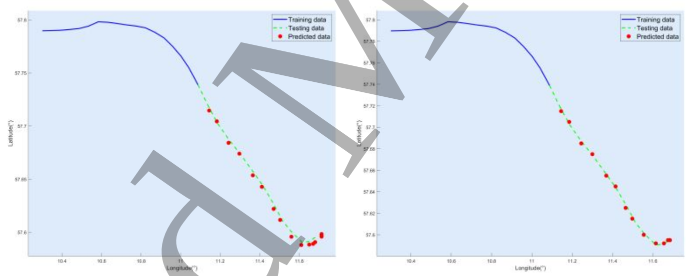
(a) LSTM (b) Conv-LSTM (c) LSTM-seq2seq-att



(d) TrAISformer (d) Proposed



(e) LSTM (f) Conv-LSTM (g) LSTM-seq2seq-att



(h) TrAISformer (i) Proposed

Fig. 8. Prediction results using Dataset 1: (a)-(d) show the test results of each model on typical trajectory 1; (e)-(i) present the test results of each model on typical trajectory 2.

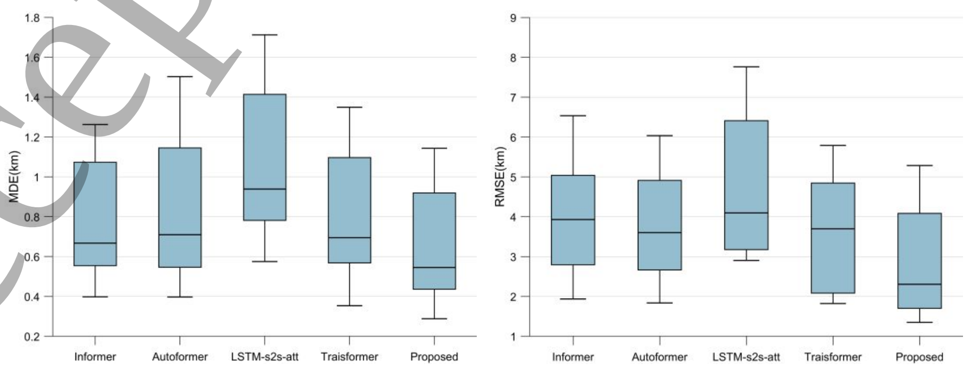
In the visual analysis, two typical trajectories from Dataset 1 were first selected to intuitively compare the performance of five models within a 3-hour future prediction interval. As shown in Fig. 8, the predicted trajectory of LSTM deviates most significantly from the real trajectory, exhibiting drastic fluctuations in both latitude and longitude dimensions, which indicates poor prediction stability. This is consistent with its high error values in MAE and RMSE metrics. Conv-LSTM alleviates such fluctuations to some extent, but obvious trajectory drift still occurs in long-time-step prediction, reflecting its limitation in capturing long-term dependency information. In contrast, the predicted trajectories of LSTM-seq2seq-attention and TrAISformer models are smoother and better fit the real paths. The proposed model shows the highest overlap with the real trajectory throughout all prediction periods, especially at key turning points and in long-term prediction areas, demonstrating stronger dynamic adaptability and trajectory stability. This intuitively reflects the advantages of the designed model in predicting vessel behavior.

Table 2 Predictive performance results of different models using Dataset 1

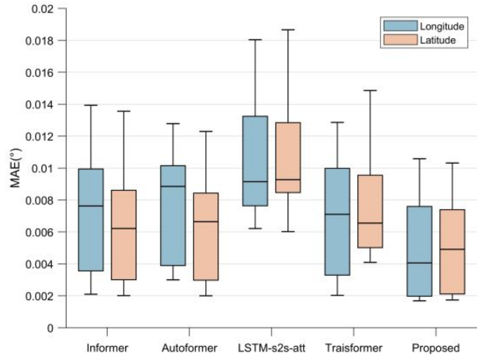
Metric	Time step	Informer	Autoformer	LSTM	Conv-LSTM	LSTM-seq2seq-att	TrAISformer	Proposed
MAE (LON)	1h	0.006962	0.007034	0.026745	0.023757	0.007870	0.006965	**0.003281**
	2h	0.011976	0.012021	0.040235	0.029802	0.013072	0.012326	**0.007313**
	3h	0.018492	0.019875	0.068606	0.052180	0.020480	0.018047	**0.010279**
MAE (LAT)	1h	0.006602	0.006821	0.013427	0.009450	0.006368	0.005126	**0.004257**
	2h	0.011195	0.010611	0.018826	0.018568	0.013574	0.011153	**0.008427**
	3h	0.016298	0.015017	0.027974	0.024608	0.021187	0.019146	**0.008995**
MDE	1h	0.63693	0.642491	1.52185	1.48814	0.80561	0.58907	**0.54412**
	2h	1.09557	1.147890	3.02954	2.62053	1.47246	0.94594	**0.76035**
	3h	1.564545	1.689158	5.01875	4.05827	2.07571	1.68975	**1.07214**
RMSE	1h	3.945338	4.143615	15.67757	11.36828	4.28476	3.50334	**2.86381**
	2h	7.703621	5.354615	21.44752	18.35773	8.08524	6.47252	**4.14692**
	3h	11.93416	9.246094	28.38911	25.26926	11.64821	8.75223	**6.46521**

The unit for MAE is degrees (°), while MDE and RMSE are measured in kilometers (km)

To evaluate the performance of the proposed model in trajectory prediction tasks, we conduct a systematic comparison of seven models across three prediction horizons (1h, 2h, and 3h), using three metrics: mean absolute error (MAE) in longitude and latitude, and spatial distance errors (MDE and RMSE), as shown in Table 2. The results indicate that the prediction error increases with time across all models. Informer and Autoformer, as enhanced Transformer-based models, demonstrate stable and low error in short-term prediction, but show a noticeable increase in error for medium- and long-term forecasts. The traditional LSTM model yields the highest error at the 3-hour mark, while Conv-LSTM performs slightly better in the short term but suffers from significant error accumulation over time. LSTM-seq2seq-attention and TrAISformer, both incorporating attention mechanisms, outperform traditional RNN-based models. The proposed model achieves the lowest MAE in both longitude and latitude across all time steps, with the slowest growth trend, indicating superior directional stability and temporal robustness. In terms of spatial error, both LSTM and Conv-LSTM exhibit severe error accumulation at the 3-hour horizon, with MDE exceeding 4 km and RMSE reaching 25–28 km. While Informer and Autoformer perform better than the traditional models, their 3-hour MDE still reaches 1.56 km and 1.69 km, respectively, and their RMSE values reach 11.93 km and 9.25 km. In contrast, the proposed model achieves a 3-hour MDE of 1.072 km and an RMSE of 6.465 km, significantly outperforming all other approaches.



(a) MDE (b) RMSE



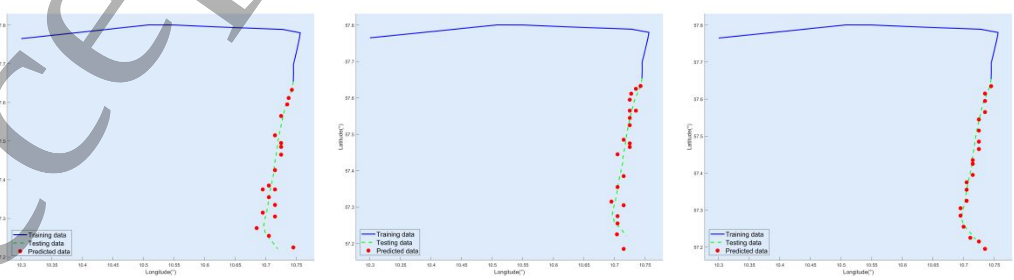
(c) MAE

Fig. 9 Boxplots of Multi-Metric Error Distributions for Each Model based on Dataset 1

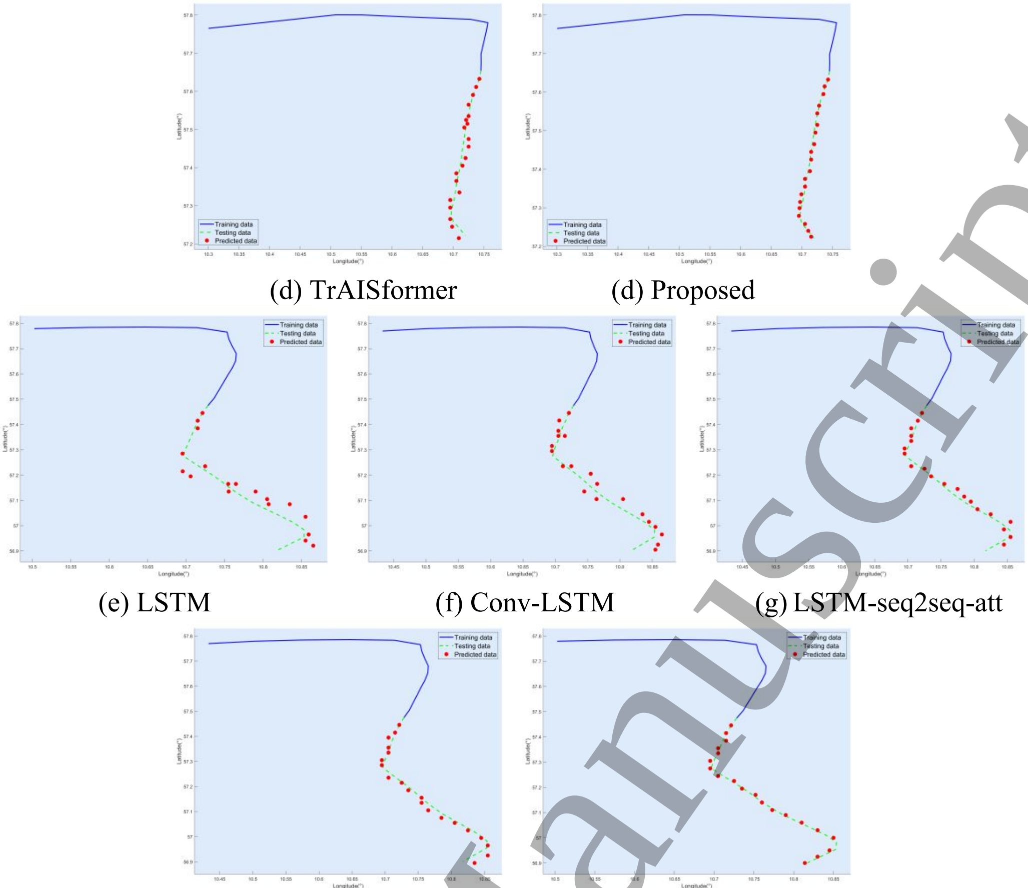
The metrics presented in Table 2 represent the average prediction performance of each model at three fixed forecasting horizons: 1 hour, 2 hours, and 3 hours. To more comprehensively capture the error distribution characteristics, we further calculate the overall average errors across all predicted trajectories within the first three hours, and visualize them using box plots based on three key metrics: MAE, MDE, and RMSE. As shown in Fig. 9, we compare the error distributions of the five best-performing models listed in the table to provide a clearer view of the differences in prediction stability for both spatial and directional errors. For the MDE metric, the proposed model shows a significantly lower box height compared to other methods, with a median consistently below 0.6 km and a compact distribution, indicating not only low overall error but also high stability. The RMSE metric further confirms this performance, with a median below 3 km, and both the interquartile range and whisker span markedly smaller than those of baseline models—reflecting high prediction consistency and few outliers. In terms of directional accuracy, the proposed model also outperforms all other methods. The error boxes for longitude and latitude MAE are entirely lower than those of the comparison models. The median longitude error reaches as low as approximately 0.004°, while the latitude error is kept within 0.006°, both exhibiting tightly clustered distributions.

3.2.2 Prediction Performance Analysis in Complex Turning Scenarios

As shown in Fig. 10 and Table 3, the same comparative experiments conducted on Dataset 1 were repeated on Dataset 2, which contains a greater number of sharply turning trajectories. Compared to the smoother trajectories in Dataset 1, this dataset presents a higher level of prediction difficulty.



(a) LSTM (b) Conv-LSTM (c) LSTM-seq2seq-att



(d) TrAISformer (d) Proposed

(e) LSTM (f) Conv-LSTM (g) LSTM-seq2seq-att

(h) TrAISformer (i) Proposed

Fig. 10. Prediction results of Dataset 2: (a)-(d) show the test results of each model on typical trajectory 1; (e)-(i) present the test results of each model on typical trajectory 2.

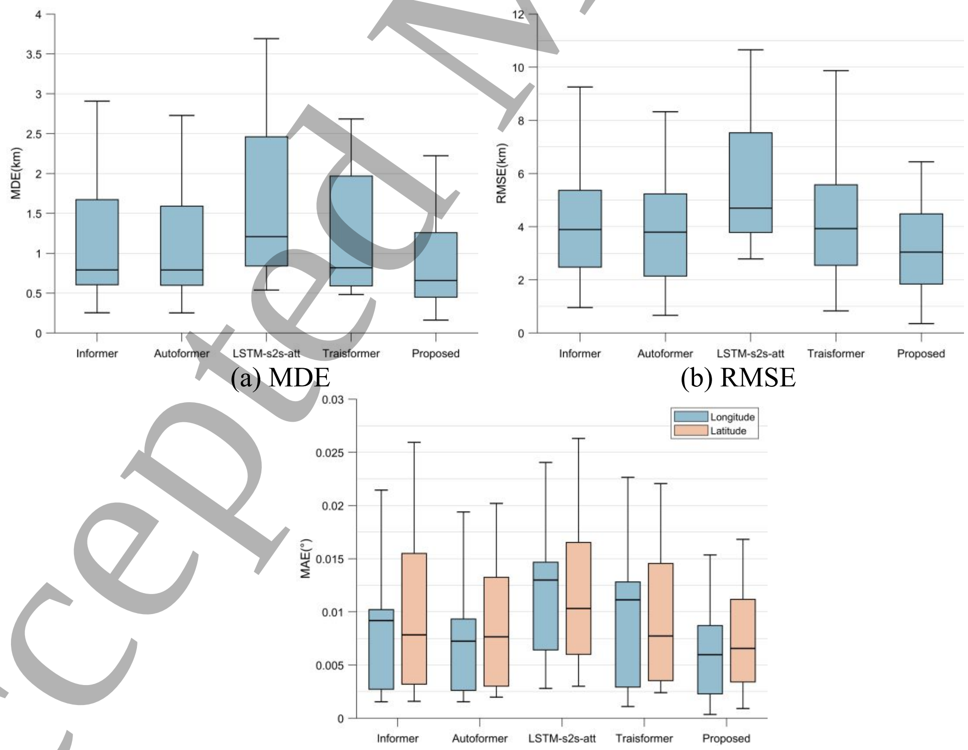
As shown in Table 3, all models exhibit increased errors on the complex trajectory dataset compared to the relatively smoother trajectory dataset, particularly during the longer-term prediction horizons of 2 and 3 hours. Traditional models such as LSTM suffer from noticeable error accumulation, indicating limited stability. Although Conv-LSTM performs slightly better in short-term prediction, its error grows rapidly as the prediction horizon extends, making it difficult to maintain effective control. LSTM-seq2seq-att and TrAISformer show relatively stable performance, with slower error growth trends. Notably, Informer and Autoformer—two Transformer-based enhanced models—demonstrate stronger adaptability on the complex turning trajectory dataset. Their ability to capture long-range dependencies and nonlinear variations in the trajectories results in better error control compared to traditional RNN-based models. In particular, Autoformer slightly outperforms the proposed model in terms of 1-hour RMSE, indicating its advantage in capturing short-term trajectory dynamics. This benefit is attributed to the Auto-Correlation mechanism introduced in Autoformer, which effectively models periodic and trend components in time series data, enabling more accurate short-term fitting. However, as the prediction horizon extends, Autoformer exhibits a sharper increase in error and performs less favorably than the proposed model. Overall, the proposed model achieves consistently lower and more stable errors across different prediction time steps. It demonstrates superior spatial fitting capability and error control, especially in medium- and long-term trajectory forecasting tasks.

Table 3 Predictive performance results of different models using Dataset 2

Metric(°)	Time step	Informer	Autoformer	LSTM	Conv-LSTM	LSTM-seq2seq-att	TrAISformer	Proposed
MAE (LON)	1h	0.006887	0.006733	0.025601	0.025299	0.009801	0.007779	**0.003517**
	2h	0.013989	0.013039	0.052403	0.030772	0.015419	0.014051	**0.008740**
	3h	0.020619	0.018934	0.079031	0.055342	0.022370	0.021143	**0.012882**
MAE (LAT)	1h	0.006587	0.004399	0.015667	0.010454	0.007812	0.005829	**0.004368**
	2h	0.012435	0.011191	0.024761	0.019862	0.014668	0.012713	**0.008652**
	3h	0.022068	0.020863	0.035440	0.030456	0.027139	0.025114	**0.011538**
MDE	1h	0.625061	0.603760	1.42357	1.66034	0.86835	0.64557	**0.56922**
	2h	1.514506	1.464011	3.36707	2.99822	2.07503	1.78421	**1.16217**
	3h	3.535683	3.237374	6.40370	6.07358	4.22299	3.88833	**2.45865**
RMSE	1h	3.812405	**2.904293**	16.14917	11.41824	4.58429	3.85713	2.96025
	2h	8.436887	6.074686	24.06923	19.73957	9.36695	8.15561	**4.90803**
	3h	15.55429	11.238997	33.60191	28.38391	17.56075	16.01375	**9.47207**

The unit for MAE is degrees (°), while MDE and RMSE are measured in kilometers (km)

As illustrated in Fig. 11, the proposed model continues to demonstrate significant advantages in AIS trajectory prediction tasks under complex scenarios. For both spatial error metrics and directional error metrics, our method exhibits lower median values, tighter distributions, and fewer outliers in the error distribution. This indicates that the model maintains high prediction accuracy and stability even when dealing with diverse and irregular trajectory patterns. Consistent with the earlier results on the simpler turning dataset, the proposed model also shows excellent generalization ability and robustness in more challenging environments.



(a) MDE (b) RMSE

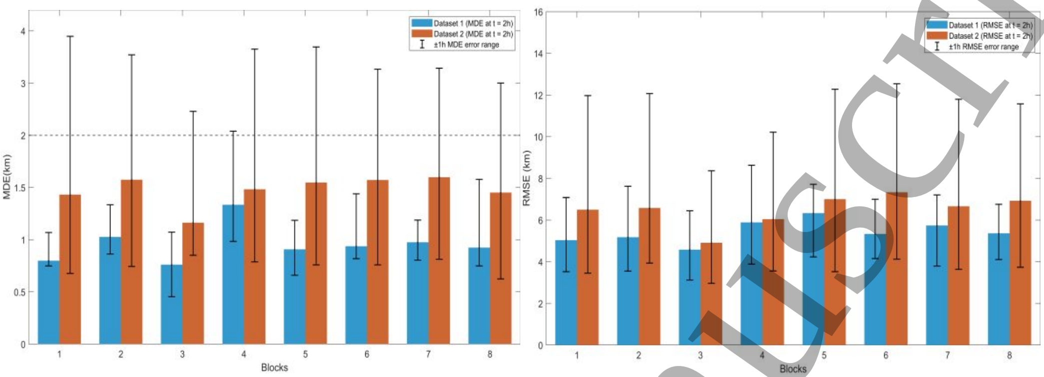
(c) MAE

Fig. 11 Boxplots of Multi-Metric Error Distributions for Each Model based on Dataset 2

3.3 Effectiveness Analysis

3.3.1 Evaluation of the Number of LCTA Blocks

To assess the impact of the number of stacked LCTA blocks on model prediction performance, comparative experiments were conducted on both trajectory prediction datasets, with the number of blocks ranging from 1 to 8 while keeping all other parameters fixed. The figure illustrates the model's MDE and RMSE at the 2-hour prediction horizon across different block configurations. The bars correspond to performance at $t = 2\text{h}$, while the error lines indicate the performance bounds at $t = 1\text{h}$ and $t = 3\text{h}$.



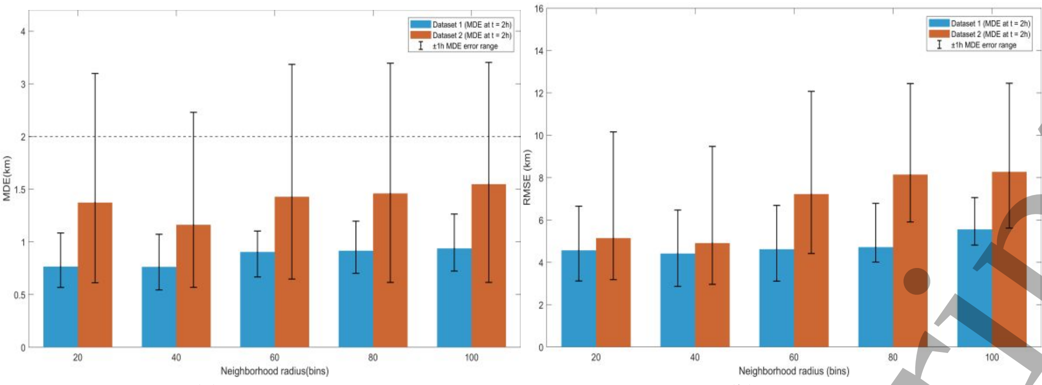
(a) MDE (b) RMSE

Fig. 12. Model performance evaluation with different numbers of LCTA blocks

As shown in Fig.12, the model achieves optimal performance when the number of blocks is set to 3. In this setting, both MDE and RMSE reach their minimum values, and the error range across different prediction horizons is the smallest, indicating superior accuracy and stability. On both datasets, this configuration yields the best error performance—achieving not only high accuracy in predicting the optimal trajectory path but also smoother spatial predictions with lower variance and better bias control. When the number of blocks increases beyond 3, although the model theoretically gains deeper learning capacity, the increased parameter count and local redundancy may lead to overfitting and feature entanglement during optimization, causing higher errors and larger variance. Conversely, with too few blocks, the model's ability to capture the temporal evolution of trajectories is insufficient, resulting in weaker global representations and suboptimal performance. Therefore, we set the number of LCTA blocks to 3 as the optimal configuration.

3.3.2 Evaluation of neighborhood search radius settings

Fig.13 illustrates the variation trends in MDE and RMSE under different values of the neighborhood search radius r as defined in (20), ranging from 20 to 100 bins. It is important to note that the trajectory prediction in this model is formulated as a discrete classification task based on a Four-Hot vector representation, where each dimension corresponds to a bin index for longitude, latitude, speed, and course. The neighborhood search radius r is measured in bins, defining a sampling constraint in the high-dimensional output index space, rather than in physical spatial units. Although bin indices cannot be directly mapped to geographical distances, they indirectly determine the spatial distribution of predicted points. As such, the neighborhood constraint introduces a structural prior during inference, improving the spatial plausibility of predicted trajectories.



(a) MDE (b) RMSE

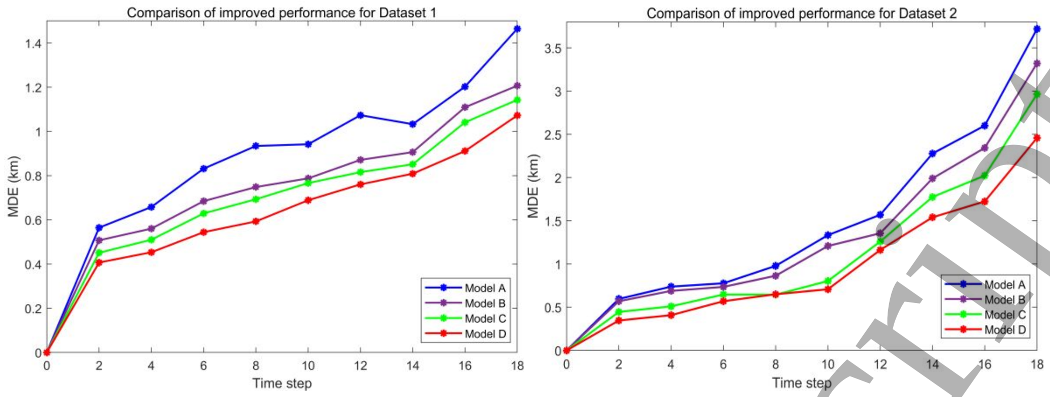
Fig.13. Model performance evaluation under different neighborhood search radius

Overall, the model achieves optimal predictive performance on both datasets when the radius r is set between 20 and 60 bins. Specifically, in Dataset 1, the performance at $r = 20$ and $r = 40$ is comparable, indicating that the model demonstrates strong robustness in locally stable trajectories. In Dataset 2, where trajectories contain more frequent and sharper turns, the model performs significantly better at $r = 40$ than at $r = 20$, suggesting that moderately increasing the neighborhood size enhances the model's ability to respond to local dynamic changes and improves prediction accuracy. However, as the radius increases further to 80 and 100 bins, the error metrics rise substantially. This degradation is primarily attributed to the inclusion of a large number of low-confidence or spatially implausible candidate bins, which undermines the consistency of the output space and introduces discontinuities or offsets in the predictions, thereby reducing overall accuracy and trajectory smoothness. Therefore, a neighborhood radius of $r = 40$ is identified as the optimal setting in this study.

To aid in understanding the spatial meaning of a bin, we provide an estimate based on the geographical coverage of the experimental dataset (57.0°N to 58.0°N, 10.3°E to 11.8°E): the latitude and longitude ranges are divided into 250 and 270 intervals, respectively, corresponding to approximately 0.004° and 0.00556° per bin. Taking $r = 40$ as an example, this translates to an approximate spatial range of 17.76 km in latitude and 13.28 km in longitude, with a maximum diagonal distance of around 22.18 km. Therefore, although the neighborhood radius is defined in the output index space, it maintains an approximately estimable relationship with real-world geographic distances. The constraint imposed by r in index space indirectly affects the spatial distribution of predicted points, enabling the incorporation of structural priors during inference and thereby enhancing the spatial plausibility of predicted trajectories.

3.4 Ablation Study

Additionally, as shown in Fig. 14, an ablation study was conducted to evaluate the contribution of each major component to the overall model performance, using Minimum Distance Error (MDE) as the evaluation metric across different prediction horizons. Model A represents a simplified Transformer architecture. Models B to D build upon their respective predecessors by progressively integrating the following enhancements: a causal feature-augmented attention mechanism, a temporal residual encoding module, and a local neighborhood search strategy.



(a) Dataset 1 (b) Dataset 2

Fig. 14. MDE values under different model configurations

Notably, on Dataset I, the MDE at the final prediction step decreased from 1.464 km (baseline) to 1.072 km after all enhancements were applied, representing a cumulative error reduction of approximately 26.8%. On Dataset II, the MDE was reduced from 3.718 km to 2.459 km, a relative improvement of 33.8%. These results indicate that the proposed enhancements are particularly effective in datasets featuring significant turning behavior, demonstrating the model's superior adaptability to non-stationary trajectory patterns. Overall, each component contributes positively at different prediction steps, resulting in synergistic structural improvements. The experimental findings strongly validate the generalizability and effectiveness of the proposed architecture across trajectory scenarios of varying complexity.

3.5 Generalization Analysis

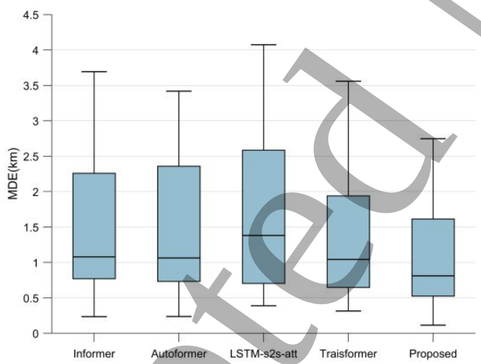
To further evaluate the model's performance in broader and more complex maritime environments, we introduce a recently released AIS historical trajectory dataset published by the Danish Maritime Authority (DMA) [27]. The dataset spans the period from March 2 to April 2, 2024, and covers a spatial range of 10.2°E–12.9°E longitude and 55.5°N–58.0°N latitude, encompassing typical high-frequency shipping areas such as the eastern Danish coast, the southwestern coast of Sweden, and the Kattegat Strait. It contains a total of 13,585 complete trajectories, featuring diverse vessel types and complex movement patterns. As the original data is irregularly sampled in time, we perform temporal standardization during preprocessing. All trajectories are resampled to a uniform time interval, with the first 20 time steps used as model input to predict the subsequent 30 steps of vessel motion. Model performance is then evaluated based on these predictions.

Table 4 Predictive performance results of different models using new Dataset

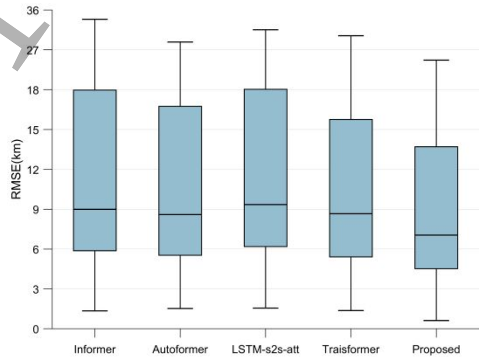
Metric(°)	Forecast points	Informer	Autoformer	LSTM	Conv-LSTM	LSTM-seq2seq-att	TrAISformer	Proposed
MAE (LON)	10	0.012942	0.015018	0.02662	0.015760	0.014068	0.011253	**0.010941**
	20	0.030319	0.034948	0.06267	0.042487	0.038700	0.032337	**0.029546**
	30	**0.052435**	0.060947	0.11118	0.093709	0.071409	0.060553	0.053520
MAE	10	0.019398	0.019932	0.02319	0.017667	0.016259	0.016079	**0.015791**

(LAT)	20	0.053353	**0.043715**	0.07597	0.059953	0.053818	0.051878	0.045251
	30	0.109634	0.089632	0.15295	0.112468	0.096629	0.090837	**0.086347**
MDE	10	1.505832	1.745682	2.58268	1.896836	1.68137	1.28194	**1.13331**
	20	3.758055	3.618130	5.45058	4.922135	4.29198	3.49912	**2.71372**
	30	6.537670	6.121121	9.44003	9.065172	7.12085	5.53136	**4.41447**
RMSE	10	8.345347	8.556036	15.8141	11.88240	9.28111	7.81144	**7.07325**
	20	22.104185	**19.959368**	28.5827	26.54231	25.37459	22.5267	20.0281
	30	37.326786	35.180548	41.0549	40.60552	38.70875	35.1871	**32.9481**

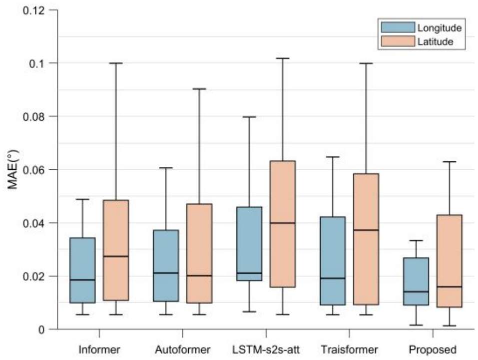
As shown in the tabulated results, the proposed model achieves the best performance across the majority of prediction metrics. In the 30-step prediction task, it attains the lowest MAE values in both longitude (0.0535°) and latitude (0.0863°). Moreover, it significantly outperforms all baseline models in terms of MDE (4.41 km) and RMSE (32.95 km), demonstrating strong robustness in long-horizon trajectory forecasting. It is worth noting that Informer slightly outperforms the proposed model in longitude MAE (0.052435°) for the 30-step setting, while Autoformer achieves marginally better performance in latitude MAE (0.0437°) and RMSE (19.95 km) under the 20-step prediction setting. These local variations in performance may stem from the baseline models' heightened sensitivity to the length of historical input, allowing them to exhibit stronger short-term fitting capabilities with longer input windows. In addition, the richness of trajectory samples in this dataset offers more training opportunities for these models, which may lead to advantages in certain metrics. In contrast, the proposed model maintains excellent stability and adaptability in this more complex AIS trajectory prediction task, benefiting from its architectural design and generalization capability. As illustrated in the Fig. 15, the model also shows superior performance in terms of median error, distribution compactness, and outlier suppression.



(a) MDE



(b) RMSE



(c) MAE

Fig. 15 Boxplots of Multi-Metric Error Distributions for Each Model based on new Dataset

4 Conclusion

This paper addresses key challenges in vessel trajectory prediction, including the presence of anomalous data, diverse behavioral patterns, and limitations in prediction accuracy and generalization capability. A novel trajectory prediction approach is proposed, incorporating a trajectory feature–based clustering strategy to construct datasets with varying turning characteristics, which provides diverse samples to support more effective model learning. A high-precision trajectory prediction model named LCTA-Net is introduced in terms of model architecture. The model integrates a residual LSTM unit with a causal feature-augmented attention mechanism to enhance its ability to capture temporal structure and long-term dependencies in trajectories. In addition, a local neighborhood search mechanism is proposed to improve the spatial coherence of predicted trajectories in the high-dimensional discrete output space. Experimental results demonstrate that the proposed method consistently achieves superior performance across multiple metrics on datasets with varying levels of trajectory complexity. In future work, we plan to incorporate multi-vessel interaction modeling to capture dynamic inter-vessel influences, thereby improving prediction capability in densely navigated or cooperative sailing scenarios.

Acknowledgements

This research is financially supported by the National Natural Science Foundation of China under Grant 61871203 and Postgraduate Research & Practice Innovation Program of Jiangsu Province under SJCX25_2505.

Data availability

The data cannot be made publicly available upon publication because they are owned by a third party and the terms of use prevent public distribution. The data that support the findings of this study are available upon reasonable request from the authors.

References

[1] Li H, Yang Z. 2023 Towards safe navigation environment: The imminent role of spatio-temporal pattern mining in maritime piracy incidents analysis. *Reliability Engineering & System Safety*, 238, 109422. https://doi.org/10.1016/j.ress.2023.109422

[2] Li H, Jiao H, Yang Z. 2023 AIS data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods. *Transportation Research Part E: Logistics and Transportation Review*, 175, 103152. https://doi.org/10.1016/j.tre.2023.103152

[3] Yang T, Wang X, Liu Z. 2022 Ship type recognition based on ship navigating trajectory and convolutional neural network. *Journal of Marine Science and Engineering*, 10(1), 84. https://doi.org/10.3390/jmse10010084

[4] Rong H, Teixeira A P, Soares C G. 2024 A framework for ship abnormal behaviour detection

and classification using AIS data. *Reliability Engineering & System Safety*, 247, 110105. https://doi.org/10.1016/j.ress.2024.110105

[5] Wang N, Yuen K, F Yuan J, Li D. 2024 Ship collision risk assessment: a multi-criteria decision-making framework based on Dempster–Shafer evidence theory. *Applied Soft Computing*, 162, 111823. https://doi.org/10.1016/j.asoc.2024.111823

[6] Li H., Xing W, Jiao, H., Yuen K. F., Gao R., Li Y, Matthews C, Yang Z. 2024 Bi-directional information fusion-driven deep network for ship trajectory prediction in intelligent transportation systems. *Transportation Research Part E: Logistics and Transportation Review*,192, 103770. https://doi.org/10.1016/j.tre.2024.103770

[7] Li H., Xing W., Jiao H., Yang Z, Li Y. 2024 Deep bi-directional information-empowered ship trajectory prediction for maritime autonomous surface ships. *Transportation Research Part E: Logistics and Transportation Review*, 181, 103367. https://doi.org/10.1016/j.tre.2023.103367

[8] Jiang L, Liu B, Jiang Y, Chen S, Wang H and Liang W. 2025 A deep graph convolution spatial-temporal attention learning model for traffic flow prediction. *Measurement Science and Technology*, 36(6), 066114. https://doi.org/10.1088/1361-6501/add955

[9] Maskooki A, Virjonen P, Kallio M. 2021 Assessing the prediction uncertainty in a route optimization model for autonomous maritime logistics. *International Transactions in Operational Research*, 28(4), 1765-1786. https://doi.org/10.1111/itor.12882

[10] Abebe M, Shin Y, Noh Y, Lee S, Lee I. 2020 Machine learning approaches for ship speed prediction towards energy efficient shipping. *Applied Sciences*, 10(7), 2325. htt ps://doi.org/10.3390/app10072325

[11] Bei T, Wang M Meng, Z Lu, B Niu, S Li, B Gao H, Li M. 2025 VIG: trajectory prediction of autonomous vehicles based on VectorNet-Interaction-GAN. *Measurement Science and Technology*. https://doi.org/10.1088/1361-6501/ade69b

[12] Liu J, Shi G, Zhu K. 2020 Online multiple outputs least-squares support vector regression model of ship trajectory prediction based on automatic information system data and selection mechanism. *IEEE Access*, 8, 154727-154745. https://doi.org/10.1109/ACCESS.2020.3018749

[13] 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*, 10, 24302-24315. https://doi.org/10.1109/ACCESS.2022.3154812

[14] Wang L, Que H, Wu F. 2024 The CNN-LSTM-attention model for short term prediction of the polar motion. *Measurement Science and Technology*, 36(1), 016323. https://doi.org/10.1088/1361-6501/ad8be5

[15] Zhang Y, Han Z, Zhou X, Zhang L, Wang L, Zhen E, Wang S, Zhao Z, Guo Z. 2023 PESO: a Seq2Seq-based vessel trajectory prediction method with parallel encoders and ship-oriented decoder. *Applied Sciences*,13(7), 4307. https://doi.org/10.3390/app13074307

[16] Lin S, Jiang Y, Hong F, Xu L., Huang H, Wang B. 2025 HDFormer: A transformer-based model for fishing vessel trajectory prediction via multi-source data fusion. *Ocean Engineering*,

320, 120309. https://doi.org/10.1016/j.oceaneng.2025.120309

[17] Altunkaynak A, Celik A, Kartal E. 2025 Eigen-ensemble-transformer model: An exceptionally accurate approach to spatiotemporal sea water level prediction. *Ocean Engineering*, 335, 121727. https://doi.org/10.1016/j.oceaneng.2025.121727

[18] Jiang D, Shi G, Li N, Ma L., Li W, Shi J. 2023 TRFM-ls: Transformer-based deep l earning method for vessel trajectory prediction. *Journal of Marine Science and Engine ering*, 11(4), 880. https://doi.org/10.3390/jmse11040880

[19] Huang F, Liu Z, Li X, Mou F., Li P, Fan Z. 2024 Vessel Trajectory Prediction Based on AIS Data: Dual-Path Spatial–Temporal Attention Network with Multi-Attribute Information. *Journal of Marine Science and Engineering*, 12(11), 2031. https://doi.org/10.3390/jmse12112031

[20] Murray B, Perera L P. 2022 Ship behavior prediction via trajectory extraction-based clustering for maritime situation awareness. *Journal of Ocean Engineering and Science*, 7(1), 1-13. https://doi.org/10.1016/j.joes.2021.03.001

[21] Zhu M, Han P, Wang C, Skulstad R, Zhang H, Li G. 2022 Ship behavior prediction via trajectory extraction-based clustering for maritime situation awareness. *Journal of Ocean Engineering and Science*, 7(1), 1-13. https://doi.org/10.1109/TSMC.2024.3458469

[22] Xu H, Zhang L, Li P, Zhu F. 2022 Outlier detection algorithm based on k-nearest neighbors-local outlier factor. *Journal of Algorithms & Computational Technology*, 16, 17483026221078111. https://doi.org/10.1177/17483026221078111

[23] Park J, Jeong J, Park Y. 2021 Ship trajectory prediction based on bi-LSTM using spectral-clustered AIS data. *Journal of marine science and engineering*, 9(9), 1037. https://doi.org/10.3390/jmse9091037

[24] 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 Transactions on Intelligent Transportation Systems*, 23(6), 5655-5667. https://doi.org/10.1109/TITS.2021.3055614

[25] Bagirov A M, Aliguliyev R M, Sultanova N. 2023 Finding compact and well-separated clusters: Clustering using silhouette coefficients. *Pattern Recognition*, 135, 109144. https://doi.org/10.1016/j.patcog.2022.109144

[26] Wang L, Chen P, Chen L, Mou, J. 2021 Ship AIS trajectory clustering: An HDBSCAN-based approach. *Journal of Marine Science and Engineering*, 9(6), 566. https://doi.org/10.3390/jmse9060566

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

[28] Zhou H, Zhang S, Peng J., Zhang S, Li J, Xiong H, Zhang, W. 2021 Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference*

on artificial intelligence, 35(12): 11106-11115. https://doi.org/10.1609/aaai.v35i12.17325

[29] Wu H, Xu J, Wang J, Long M. 2021 Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34, 22419-22430.

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