Cluster-aware deep learning-based vessel trajectory prediction for maritime traffic management

Yan Li [a], Lan Song [b *], Zaili Yang [c], Huanhuan Li [d]

[a] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China

[b] College of Engineering, Eastern Institute of Technology, Ningbo, China

[c] Liverpool Logistics, Offshore and Marine (LOOM) Research Institute, Liverpool John Moores University, Liverpool, UK

[d] School of Engineering, University of Southampton, Southampton, UK

* Corresponding authors: lansong@eitech.edu.cn

Abstract: Accurate vessel trajectory prediction is essential for maritime safety and traffic management. While Automatic Identification System (AIS) data offer valuable spatiotemporal information, reliable forecasting remains challenging due to heterogeneous traffic patterns and dynamic behaviors. This study proposes a two-stage, cluster-aware framework, IDCMSA-MSIFE, that integrates trajectory clustering with multi-source deep learning to enhance prediction accuracy and robustness. In the first stage, an Improved Density Clustering with Mapping and Spatial Accessibility (IDCMSA) method is developed to identify representative traffic patterns by mapping trajectories to characteristic points and detecting their density-reachable relationships within a grid-based structure. This process distinguishes heterogeneous navigation behaviors in complex maritime environments. In the second stage, trajectories in each cluster are fed into a Multi-Source Information Fusion Enhanced (MSIFE) deep learning network. The model integrates multiple AIS-derived features, including trajectory coordinates, Course Over Ground (COG), and Speed Over Ground (SOG), through a self-attention mechanism, while Bidirectional Long Short-Term Memory (Bi-LSTM) units capture temporal movement dynamics. To enhance training stability and predictive consistency, correction terms are added to the loss function. Experimental results on real-world AIS datasets from two maritime regions show that our framework outperforms seven benchmarks in accuracy and robustness, demonstrating its value for navigation support and collision risk mitigation. This study provides a data-driven approach for safer maritime management.

Keywords: Maritime IoT, AIS data, Vessel trajectory prediction, Clustering algorithm, Deep learning

1. Introduction

Economic globalization has intensified the cross-border movement of goods, raw materials, and resources, resulting in increasingly frequent international trade activities (W. Xing et al., 2023). As the backbone of global commerce, transportation services are indispensable because the movement of goods between countries cannot occur without them (Baeza et al., 2022). Among various transport modes, maritime shipping remains the dominant choice due to its substantially lower cost and significantly higher carrying capacity compared with air or pipeline transport (Y. Li et al., 2024). As maritime transport grows, vessel traffic surges and interactions become more frequent within constrained waterways. As a result, this leads to greater traffic pressure and higher risks of congestion and collision incidents (Gan et al., 2023). Traffic congestion undermines operational efficiency and increases transport costs, whereas collisions may result in catastrophic consequences, including loss of life, property damage, and environmental pollution (Liu et al., 2025). These challenges underscore the urgent need for

advanced technical solutions that support proactive maritime risk mitigation.

Vessel trajectory prediction constitutes a critical research focus for addressing these issues. By analyzing the evolution of historical vessel movement patterns, trajectory prediction models can infer future vessel positions and thereby identify potential high-risk encounters in advance (H. Li et al., 2024; Wang et al., 2025). As illustrated in Fig. 1, accurate forecasting of trajectory development trends enables maritime authorities to identify collision-prone regions promptly and implement timely interventions.

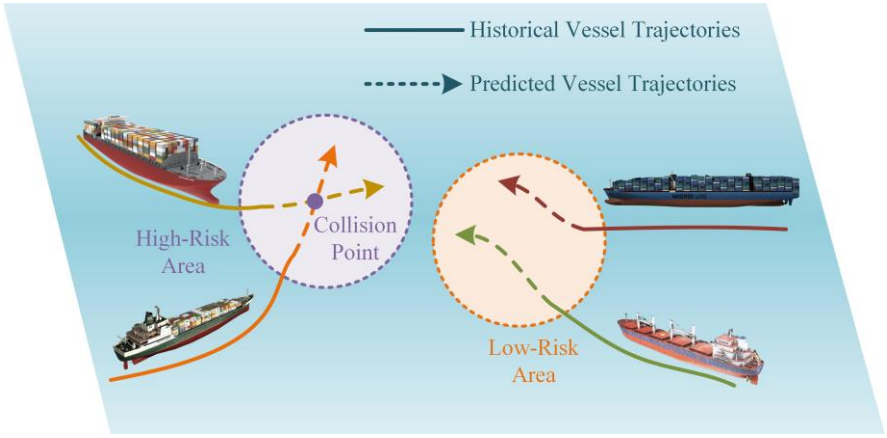

Fig. 1. Visualization of the application background for vessel trajectory prediction.

Recent innovations in Location-Based Services (LBS) and the maritime Internet of Things (IoT) have further advanced this field. The Automatic Identification System (AIS), a core component of the maritime IoT, provides high-frequency dynamic and static information on vessel navigation, anchoring, and mooring activities (Liang et al., 2025). Dynamic attributes include timestamps, geographical coordinates, Course over Ground (COG), and Speed over Ground (SOG), whereas static attributes include vessel type, dimensions, and Maritime Mobile Service Identity (MMSI) (Shu et al., 2024). This wealth of AIS data forms a crucial foundation for data-driven vessel trajectory prediction, supporting both maritime situational awareness and the broader development of Intelligent Transportation Systems (ITS) in the marine environment (see Fig. 2).

The availability of rich data enables the investigation of new AI solutions in ship trajectory prediction. Deep learning (DL) (Shin and Yang, 2025) and neural network (NN) (Deng et al., 2025) models have become prominent tools for vessel trajectory prediction due to their capacity to capture nonlinear patterns embedded in large-scale AIS datasets. Although DL extends the architecture of traditional NNs through deeper and more complex layers (Li et al., 2025), a common limitation persists: most existing studies train a single prediction model on all trajectories within the target region. In complex waterways, where vessel movements exhibit diverse structural patterns, mixing heterogeneous trajectories in a single model can obscure the unique features of specific trajectory types, thereby reducing prediction accuracy and stability. To alleviate this challenge, several studies have incorporated clustering techniques that categorize trajectories according to spatial and temporal patterns before model training (Alam et al., 2024; Park et al., 2021; Yang et al., 2022a).

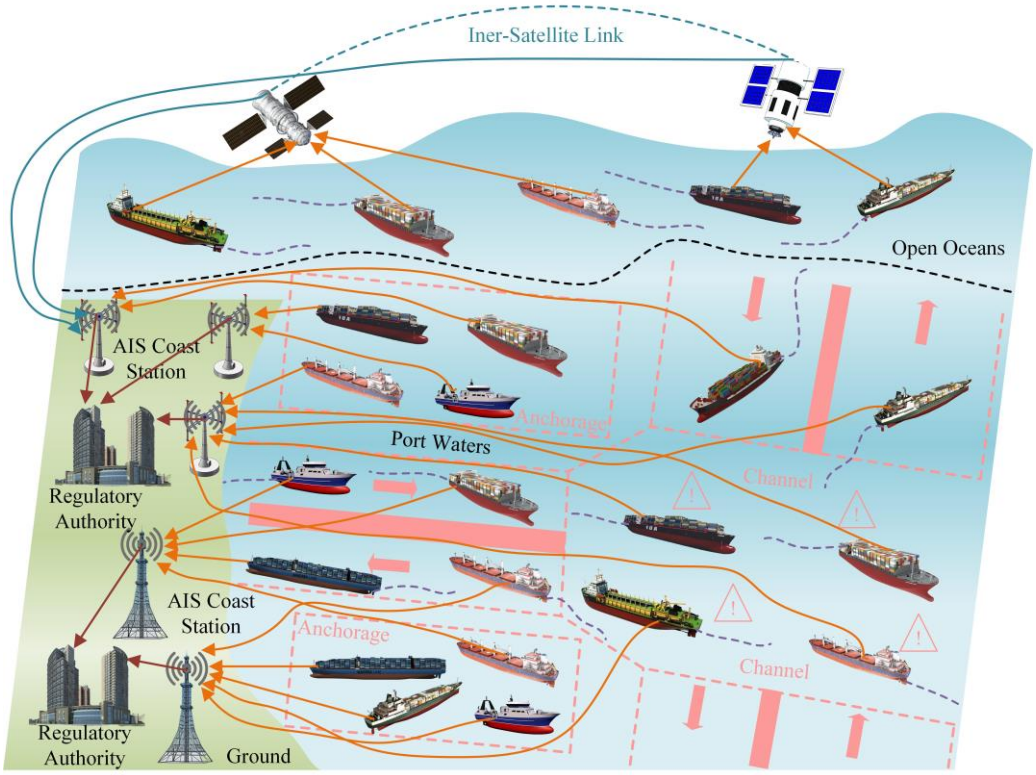

Fig. 2. Visual illustration of the maritime IoT system.

Building on this idea, this study develops a two-stage prediction framework. It aims to improve accuracy, robustness, and interpretability. The focus is on forecasting vessel trajectories in complex maritime environments. In such environments, many state-of-the-art methods fail to deliver reliable predictions. The proposed framework consists of two components. First, a novel trajectory clustering algorithm called Improved Density Clustering with Mapping and Spatial Accessibility (IDCMSA). Second, a multi-source information fusion deep learning model, named the Multi-Source Information Fusion Enhanced (MSIFE) network. Together, these components form the IDCMSA-MSIFE trajectory prediction framework.

This study offers four key contributions to advance vessel trajectory prediction and address gaps in maritime intelligent transportation research:

(1) Development of an integrated hierarchical prediction framework (IDCMSA-MSIFE).

A new two-stage trajectory prediction methodology is constructed. The framework first performs trajectory clustering to reduce data heterogeneity and improve clustering precision, followed by independent DL-based training for each cluster to enhance prediction reliability. This hierarchical structure overcomes limitations in existing single-model approaches.

(2) Proposal of the IDCMSA algorithm for vessel trajectory clustering.

A novel clustering technique is designed by mapping each trajectory into a point via Multidimensional Scaling (MDS), projecting these points into a density matrix, and applying convolution to detect spatial accessibility relationships. Trajectories with density-reachable relationships are grouped into the same cluster. This approach retains the advantages of density-based clustering (e.g., Density-Based Spatial Clustering of Applications with Noise (DBSCAN)) while addressing its shortcomings: (i) eliminating the need for complex hyperparameter tuning and (ii) improving robustness to trajectory noise.

(3) Development of the MSIFE network for multi-source trajectory learning.

The MSIFE model fully leverages multi-source AIS information, trajectory coordinates, COG, and SOG, through an attention-based feature fusion mechanism. Bidirectional Long Short-Term Memory (Bi-LSTM) units capture temporal evolution patterns, while three corrective loss terms enhance backward optimization, enabling the model to better learn intricate trajectory dynamics and achieve higher prediction accuracy.

The remainder of the paper is organized as follows. Section 2 presents a literature review on vessel trajectory clustering and prediction, highlighting the existing research gaps that motivate the development of the proposed framework. Section 3 presents key definitions, describes the AIS datasets, and explains the data preprocessing procedures. Section 4 elaborates on the theoretical principles of IDCMSA and MSIFE, including the limitations of existing approaches that the proposed methods address. Section 5 reports the results of ablation and comparative experiments that validate the effectiveness of the framework. Section 6 summarizes the main findings, discusses practical implications, and outlines future research directions.

2. Literature review

Recent years have witnessed substantial progress in vessel trajectory prediction, driven by the increasing demand for accurate forecasting to support safe and efficient maritime operations. A common strategy in the literature involves first clustering trajectories in the target region and then training NN or DL models on cluster-specific subsets to generate predictions for future time steps. To clearly define the strategy by evidence and provide a systematic understanding of existing research, this section is organized into three parts. Section 2.1 reviews vessel trajectory clustering algorithms. Section 2.2 summarizes NN- and DL-based trajectory prediction models. Section 2.3 identifies the key research gaps motivating the present study.

2.1. Overview of vessel trajectory clustering algorithms

Differences between vessel trajectories are primarily reflected in their spatial and temporal similarities (Guo et al., 2024). Clustering algorithms exploit these relationships to group trajectories exhibiting common movement patterns, thereby enhancing interpretability and facilitating downstream prediction tasks (Sheng et al., 2018). Existing clustering approaches can be broadly classified into seven methodological categories. (1) Partition-based clustering method. The K-Means clustering algorithm (Ikotun et al., 2023) represents this category. It partitions trajectories into clusters based on proximity to cluster centroids. Although computationally efficient, its reliance on predefined cluster numbers and sensitivity to initialization can limit performance. (2) Hierarchical clustering algorithm. This technique iteratively merges or splits clusters to construct a tree-like structure that captures multi-level data organization (Ran et al., 2023). It offers interpretability but incurs high computational costs on large-scale AIS datasets. (3) Model-based clustering. The Gaussian Mixture Model (GMM) (Fu et al., 2021) is a typical representative that can probabilistically model trajectory coordinate distributions. However, GMM struggles with irregular trajectory shapes and noise. (4) Grid- or network-based clustering. These methods map trajectories onto grid matrices and determine clusters based on cell densities, as seen in Statistical Information Grid (Bureva et al., 2017) and Clustering In QUEst (Ma et al., 2024). They are suitable for large datasets but may be sensitive to grid resolution. (5) Graph-based clustering. Trajectories are treated as nodes connected by spatial–temporal proximity, forming the basis for clustering via graph partitioning techniques (Chen and Pan, 2018) to get different clusters (Ding et al., 2024). (6) Fuzzy clustering. Fuzzy C-Means (FCM) (Bezdek et al., 1984) assigns each trajectory a membership degree across

clusters. Although suitable for ambiguous boundaries, overlapping vessel trajectories can lead to instability in final classifications. (7) Density-based clustering. DBSCAN (Latifi-Pakdehi and Daneshpour, 2021) identifies clusters based on neighbourhood density parameters. While robust to noise, its performance heavily depends on optimal parameter selection, which can be challenging in heterogeneous traffic environments.

Although these methods provide diverse perspectives, they share a common limitation: their performance declines when dealing with increased irregular, noisy, or highly variable maritime trajectories, a challenge that motivates the development of new adaptive and robust clustering mechanisms.

2.2. Overview of vessel trajectory prediction models

Trajectory prediction models built on training networks typically consist of five components: input layer, hidden layer, output layer, optimizer, and loss function. The hidden layer structure, optimizer selection, and loss function design are particularly critical, as they directly influence learning capability and prediction performance. Depending on the depth and complexity of the hidden layers, these models fall into two categories: NN-based and DL-based approaches.

2.2.1. NN-based prediction models

Traditional NNs consist of a single or a limited number of hidden layers with relatively simple structures. They have been widely used in vessel trajectory prediction due to their ability to approximate nonlinear functions. Representative NN architectures include Back Propagation Neural Network (BPNN) (Chen et al., 2022), Wavelet Neural Network (WNN) (Ali and Aly, 2024), Generalized Regression Neural Network (GRNN) (Rutkowski, 2004), Elman Neural Network (ENN) (Xu et al., 2023), Fuzzy Neural Network (FNN) (Zhang et al., 2021), Grey Neural Network (GNN) (Shi et al., 2024).

To enhance their prediction accuracy and robustness, many studies incorporate optimization techniques such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Differential Evolution (DE). Examples include GA-ENN for improved forecasting accuracy (B. Liu et al., 2022), GA-BPNN for enhanced network fitting capability (Yi et al., 2021), and hybrid GA-PSO-based optimizations to mitigate issues arising from randomly initialized weights (Ma et al., 2020).

Despite these improvements, NN-based models often struggle with complex and long-range temporal dependencies inherent in vessel trajectories due to their limited representational capacity.

2.2.2. DL-based prediction models

Deep learning models extend NNs through deeper architectures and larger numbers of neurons, enabling the extraction of high-level abstract features essential for modelling complex vessel trajectory patterns (Jiao et al., 2026; Shu et al., 2026). DL models can be broadly categorized into discriminative and generative approaches (Fig. 3). Specifically, discriminative networks model the conditional probability distribution between an input sequence and its corresponding label, enabling them to perform tasks such as label prediction (Li et al., 2023a, 2023b). Their primary applications include time series forecasting (Huang et al., 2022), image or text classification (Chen and Feng, 2020), recommendation systems (Milanes et al., 2021). In contrast, generative networks learn the joint probability distribution of input sequences and labels, allowing them to generate new data samples. They are widely used in applications such as image or text generation (de Rosa and Papa, 2021; Xu et al., 2018), data augmentation

(Antoniou et al., 2018), style transfer (Zhang and Dana, 2018).

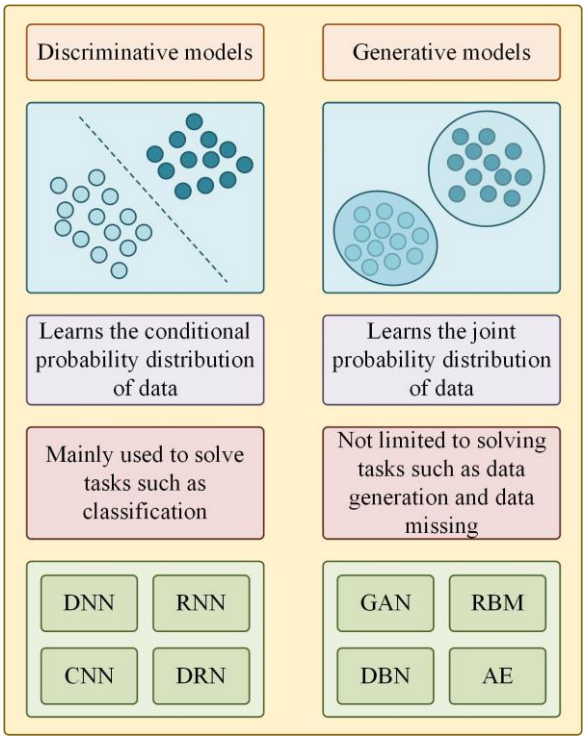

Fig. 3. Visualization of DL classification structure.

Discriminative networks mainly include Deep Neural Network (DNN) (Samek et al., 2021), Recurrent Neural Network (RNN) (Mienye et al., 2024), Convolutional Neural Network (CNN) (Yang et al., 2021), Deep Residual Network (DRN) (Hu et al., 2023), etc.

DNN offers stronger hierarchical feature extraction capabilities than BPNN. X. Li et al. (2022) integrated DNN with LSTM while incorporating environmental variables (e.g., wind, temperature, ocean currents) to enhance trajectory prediction. RNN captures temporal dependencies but suffers from vanishing/exploding gradients. LSTM (Gao et al., 2021) and Gated Recurrent Unit (GRU) (Xue et al., 2024) address these limitations, while their bi-directional variants, Bi-LSTM (Yang et al., 2022b) and Bi-directional Gated Recurrent Unit (Bi-GRU) (Reza et al., 2023), further capture global contextual information. In practical applications, Y. Li et al. (2024) converted ship trajectory coordinates into Geohash-encoded embedding vectors, reduced their dimensionality using a BPNN, and then applied an LSTM to capture temporal dependencies for trajectory prediction. CNNs, which are designed for matrix-structured data, extract local features through convolution operations and progressively build high-level semantic representations through their hierarchical architecture. CNNs extract spatial features in trajectory data and are often combined with RNN variants for improved learning, including CNN-LSTM (Lu et al., 2021; Syed and Ahmed, 2023), CNN-GRU (Du et al., 2025), CNN-BiLSTM (Yang et al., 2024), and CNN-BiGRU (Chang et al., 2024), etc. CNNs learn the spatial patterns of trajectories, while RNNs capture their temporal dynamics. Combining the two can therefore improve the model's overall learning capability. In maritime ITS, dynamic vessel interactions, waterway topologies, and port layouts often exhibit complex graph structures that are not fully captured by traditional grid or sequence representations (Gan et al., 2025; Gong et al., 2025). To model such inherently non-Euclidean traffic data more effectively, Graph Convolutional Networks (GCNs) have been introduced to this domain.

GCNs extend the convolution operation to graph structures, learning deep representations of waterway networks and vessel interactions by aggregating information from nodes and their neighborhoods (Yin et al., 2025). This approach naturally captures cooperative and avoidance behaviors among vessels, channel connectivity, and port-area influences. As a result, GCNs provide a more powerful modeling tool for tasks such as vessel trajectory prediction and navigation conflict detection. The core idea of DBNs is to use residual learning to address gradient explosion or vanishing issues during deep network training, allowing the model to support more hidden layers and better learn data variation patterns.

Generative approaches learn joint probability distributions and can produce synthetic trajectories or latent representations useful for prediction. They mainly include Generative Adversarial Network (GAN) (Aggarwal et al., 2021), Restricted Boltzmann Machine (RBM) (Zhang et al., 2018), Deep Belief Network (DBN) (Sohn, 2021), and Auto-Encoder (AE) (Wang et al., 2016), etc. Specifically, GAN consists of a generator and a discriminator trained in an adversarial framework. The generator produces realistic samples, while the discriminator distinguishes real data from generated data. Because GANs learn directly from the data distribution without requiring labels, they are well-suited for unsupervised high-quality sample generation. Jia and Ma (2023) applied this concept to vessel trajectory prediction by proposing a Conditional Temporal GAN (CTGAN). Their model uses a trajectory generator to learn vessel movement patterns and produce future trajectory distributions, while an intent classifier evaluates whether the generated trajectories align with realistic behaviour. Through adversarial training, the two components jointly improve prediction performance. RBM is an unsupervised generative model composed of visible and hidden layers. It is effective at capturing latent features and is particularly useful for dimensionality reduction. Mocanu et al. (2017) developed a disjunctive factored four-way conditional RBM that integrates tensor factorization to better model high-dimensional data. DBN is a hierarchical generation model formed by stacking multiple RBMs. It employs layer-wise unsupervised pre-training followed by supervised fine-tuning, helping to mitigate gradient vanishing or explosion in deep networks. B. Xing et al. (2023) used DBN for trajectory prediction and improved performance by optimizing key hyperparameters, including the number of hidden layers, hidden units, and learning rate. AE uses an encoder-decoder architecture to learn low-dimensional representations of data, enabling feature extraction, data dimensionality reduction, and generation tasks. Wang et al. (2024) proposed SocialVAE and Transformer-based models for vessel trajectory prediction, explicitly modelling vessel collision-avoidance behaviour to improve prediction accuracy in emergency scenarios.

2.3. Research Gaps

Despite considerable progress in vessel trajectory clustering and prediction, several fundamental limitations remain unresolved in existing studies.

(1) Limited robustness of conventional clustering algorithms for complex maritime trajectories.

Widely adopted clustering approaches, such as DBSCAN and K-means, exhibit inherent difficulties when applied to irregular, noisy, and highly diverse vessel trajectories. Their performance is highly sensitive to heterogeneous spatial densities, variable trajectory lengths, and parameter selection, which often leads to unstable clustering outcomes and poor discrimination between adjacent or overlapping routes in congested waterways. In addition,

susceptibility to noise and outliers frequently leads to cluster fragmentation or erroneous merging. These limitations reveal the necessity for a clustering mechanism that is both density-adaptive and robust to noise while preserving fine-grained structural characteristics of maritime traffic flows.

To address this gap, the proposed IDCMSA introduces a density-aware multi-scale aggregation strategy that explicitly accounts for spatial heterogeneity and noise, enabling more stable and reproducible clustering of complex vessel trajectories.

(2) Insufficient integration between trajectory clustering and DL-based prediction.

Although clustering has been employed in some studies as a preprocessing step to reduce data heterogeneity, most existing frameworks treat clustering and prediction as loosely coupled or sequential modules. Such designs typically rely on relatively simple clustering schemes and fail to tightly integrate refined clustering outcomes with modern deep learning architectures. Moreover, the joint exploitation of multi-source dynamic features (e.g., SOG and COG) within a unified clustering-prediction pipeline remains limited. This separation constrains the ability of prediction models to fully leverage cluster-specific motion characteristics.

To bridge this gap, this study develops a unified IDCMSA-MSIFE framework in which clustering and prediction are co-designed and tightly coupled. The clustering results produced by IDCMSA directly inform the MSIFE network, enabling cluster-aware representation learning and prediction using fused dynamic features.

(3) Inadequate optimization objectives for heterogeneous trajectory behaviors.

Most existing DL-based trajectory prediction models adopt loss functions based solely on point-wise Euclidean distance between predicted and observed positions. Such formulations implicitly assume homogeneous motion patterns and therefore fail to capture the distinct characteristics of different vessel behaviors, including stationary, low-speed drifting, and high-maneuvering motions. When applied to highly heterogeneous trajectory clusters, these generic objectives can lead to suboptimal convergence and biased learning.

To overcome this limitation, the proposed MSIFE incorporates a tailored optimization strategy that accounts for behavioral heterogeneity and supports joint modeling of stationary and moving vessels, thereby aligning the learning objective more closely with the intrinsic properties of diverse trajectory patterns.

3. Preliminary

This section provides the basic definitions and data structures required to understand the proposed prediction framework, followed by a detailed description of the experimental dataset used to validate the method. In addition, because AIS data are recorded at irregular time intervals, interpolation is necessary before trajectory prediction. Accordingly, Sections 3.1, 3.2, and 3.3 present the formal definitions, dataset description, and data preprocessing procedures, respectively.

3.1. Definitions

Definition 1. Vessel trajectory.

A vessel trajectory during navigation consists of n sequential points, denoted as

$$vtp = \{pt_1, pt_2, \cdots, pt_m, pt_{m+1}, \cdots pt_n\}, pt_m = (\alpha_m, \beta_m), m = 1, 2, \cdots n-1, n \quad (1)$$

where (α_m, β_m) is the latitude and longitude of the m-th vessel trajectory point.

Definition 2. Multi-source information. A sequence *msi* with size n includes the dynamic

variables recorded at each trajectory point:

$$msi = \{ pti_1, pti_2, \cdots, pti_m, pti_{m+1}, \cdots pti_n \}, pti_m = (\alpha_m, \beta_m, c_m, s_m), m = 1, 2, \cdots, n-1, n \quad (2)$$

where c_m and s_m are the COG and SOG of the m-th vessel trajectory point, respectively. The meanings of α_m and β_m remain consistent with Definition 1.

Definition 3. Training model input sequence. The MSIFE network processes sequences of consecutive trajectory observations. Each input sequence is defined as

$$Mat_{tn\times 2} = \begin{bmatrix} \alpha_{tn-i} & \beta_{tn-i} & c_{tn-i} & s_{tn-i} \\ \alpha_{tn-i+1} & \beta_{tn-i+1} & c_{tn-i+1} & s_{tn-i+1} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{tn-1} & \beta_{tn-1} & c_{tn-1} & s_{tn-1} \\ \alpha_{tn} & \beta_{tn} & c_{tn} & s_{tn} \end{bmatrix}, tn > 0 \quad (3)$$

where tn represents the length, consistent with the "Sequence Length" parameter described in Section 5.3.1 is the same as tn.

This study focuses solely on the dynamic information from AIS data for trajectory prediction. Although AIS also provides static attributes such as vessel type, dimensions, and MMSI, these are excluded for three reasons. First, static attributes are highly heterogeneous and may introduce unnecessary complexity and reduce model generalizability across vessel types. Second, short-term trajectory prediction is mainly determined by real-time kinematic states (position, COG, and SOG), while static properties have limited influence over short horizons. Third, excluding static features reduces input dimensionality, thereby improving training and inference efficiency without sacrificing prediction accuracy.

3.2. Experimental dataset description

To evaluate the effectiveness and generalizability of the proposed trajectory prediction framework, realistic AIS data from two representative maritime regions are employed: Caofeidian Port (CfdP) and Chengshan Jiao Promontory (CsJP). Fig. 4 shows the distribution of vessel trajectories projected onto electronic nautical charts, while Table 1 summarizes key statistics.

The CsJP region contains more irregular and intersecting trajectories than CfdP, reflecting higher navigational complexity. This characteristic imposes greater challenges for clustering and prediction, making CsJP an informative environment for validating model robustness.

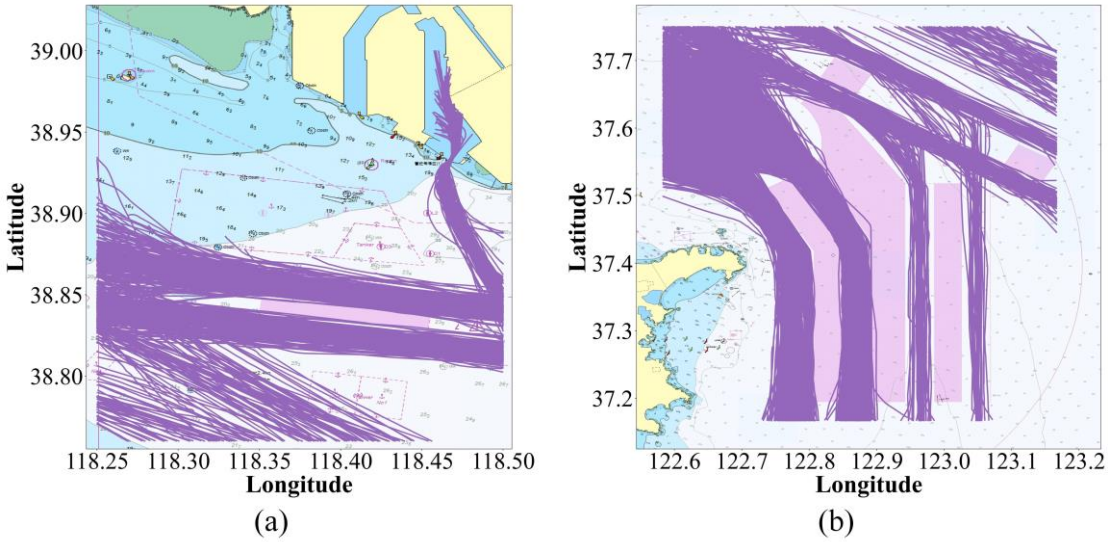

Fig. 4. Visualization of realistic vessel trajectory distribution: (a) CfdP and (b) CsJP.

Table 1. Statistical data related to the two research water regions.

Water Regions	Number of Vessel Trajectories	Period	Boundary Points	Longitude(°)	Latitude(°)
CfdP	2649	Jul. 6th -19th, 2020	Left Top	118.2500	39.0000
			Right Bottom	118.5000	38.7600
CsJP	2948	Jul. 6th -19th, 2020	Left Top	122.5833	37.7500
			Right Bottom	123.1667	37.1667

3.3. AIS data preprocessing

AIS data are typically recorded at non-uniform time intervals due to variations in vessel motion, sailing status, and transmission conditions. As illustrated in Fig. 5(a), intervals may range from a few seconds during high-speed navigation to several minutes when vessels are anchored or moored. However, deep learning models require uniformly sampled time-series data. Thus, interpolation of longitude, latitude, COG, and SOG at equal time intervals is necessary before model training (Y. Zhang et al., 2025).

Cubic Spline Interpolation (CSI) is adopted for this purpose due to its simplicity, smoothness, and strong generalization capability (Guo and Fu, 2020). CSI constructs spline functions for each attribute (longitude, latitude, COG, SOG) based on original timestamps, and generates interpolated values by evaluating these functions at uniform time intervals, as depicted in Fig. 5(b). The Python library "scipy.interpolate.spi" provides efficient implementations to facilitate the interpolation process (Y. Li et al., 2024). Algorithm 1 shows that the CSI algorithm relies on a Python library to perform the interpolation calculations used in this study.

Algorithm 1: AIS data interpolation

Input: *tOri*, *αOri*, *βOri*, *cOri*, *sOri*, *tInter*// *tOri*, *αOri*, *βOri*, *cOri*, and *sOri*, respectively, denote the time information, longitude, latitude, COG, and SOG collected based on the original AIS data. *tInter* represents equidistant timestamp sequence data. They are all stored in a list structure.

Output: *αInter*, *βInter*, *cInter*, *sInter* // *αInter*, *βInter*, *cInter*, and *sInter*, respectively, represent equal time interval longitude, latitude, COG, and SOG.

// Construct the functional relationship between the original timestamp and longitude, latitude, COG, and SOG, respectively.

1. f_α = scipy.interpolate.spi.splrep(*tOri*, *αOri*);
2. f_β = scipy.interpolate.spi.splrep(*tOri*, *βOri*);

3. f_c = scipy.interpolate.spi.splrep(*tOri*, *cOri*);
4. f_s = scipy.interpolate.spi.splrep(*tOri*, *sOri*);
//
5. *αInter* = scipy.interpolate.spi.splev(*tInter*, f_α);
6. *βInter* = scipy.interpolate.spi.splev(*tInter*, f_β);
7. *cInter* = scipy.interpolate.spi.splev(*tInter*, f_c);
8. *sInter* = scipy.interpolate.spi.splev(*tInter*, f_s);
9. **reutrn** *αInter*, *βInter*, *cInter*, *sInter*

Fig. 6 presents a comparative analysis of two vessels in the CsJP waters. Plots (a, c) and (b, d) display the scatter distributions of the original trajectory, COG, and SOG for the first and second vessel, respectively. Their corresponding distributions after interpolation are shown in (e, g) and (f, h). The comparisons confirm that CSI accurately reconstructs continuous vessel motion while preserving the inherent structure and characteristics of the original AIS data. This preprocessing step ensures that the interpolated sequences meet the input requirements of the MSIFE network and support reliable trajectory prediction.

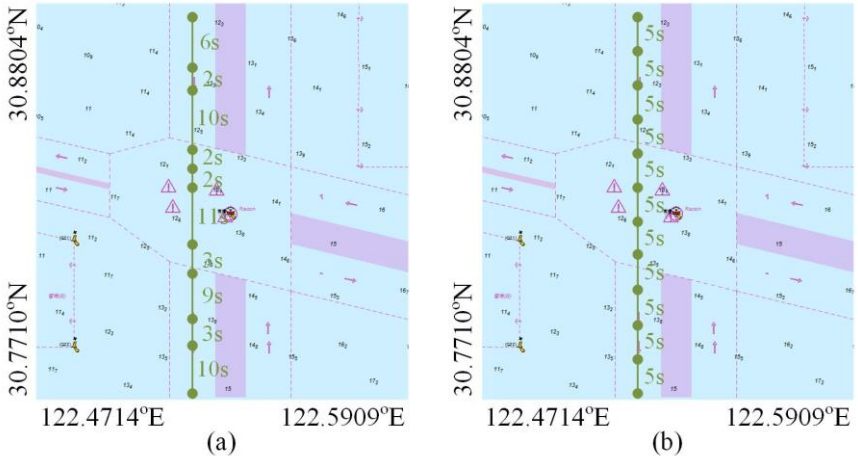

Fig. 5. Visualization demonstration of equidistant time interval interpolation for vessel trajectory points. From left to right: (a) original vessel trajectory, (b) interpolated vessel trajectory.

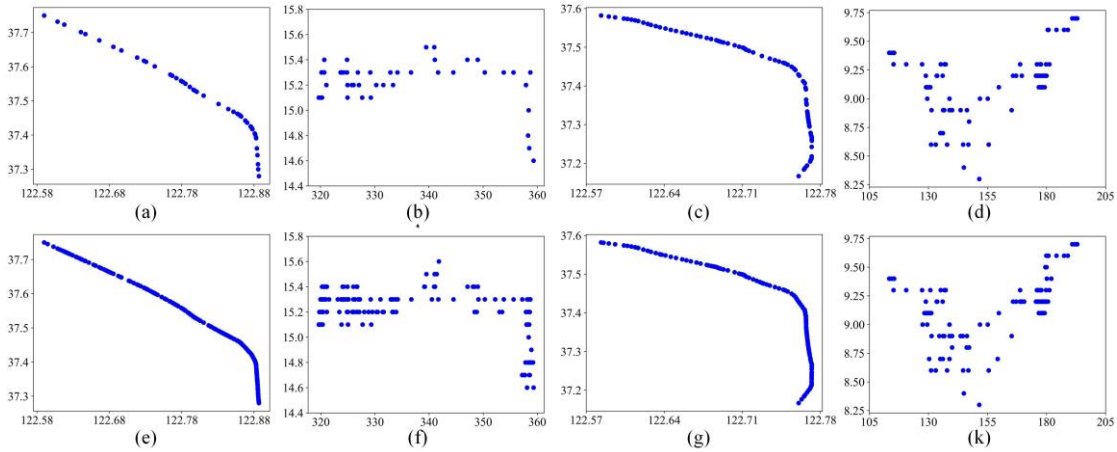

Fig. 6. Comparative visualization of vessel trajectory coordinates, COG, and SOG before and after interpolation. (a, b, e, f) Original and interpolated trajectory, COG, and SOG scatter plots for one vessel; (c, d, g, h) corresponding plots for another vessel.

4. Methodology

This study proposes an innovative vessel trajectory prediction framework that has new features to address the research gaps in Section 2.3 and can improve ship trajectory prediction

accuracy when trajectories expose more irregular and interaction patterns. The central idea is to first cluster trajectories within the study area and then train separate prediction models for different trajectory categories to improve accuracy and stability. The resulting model can simultaneously integrate multiple dynamic data sources, including vessel coordinates, COG, and SOG. The overall framework is shown in Fig. 7 and comprises two main components: (1) the IDCMSA clustering method introduced in Section 4.1, and (2) the MSIFE prediction network described in Section 4.2, which can effectively fuse multi-source information effectively.

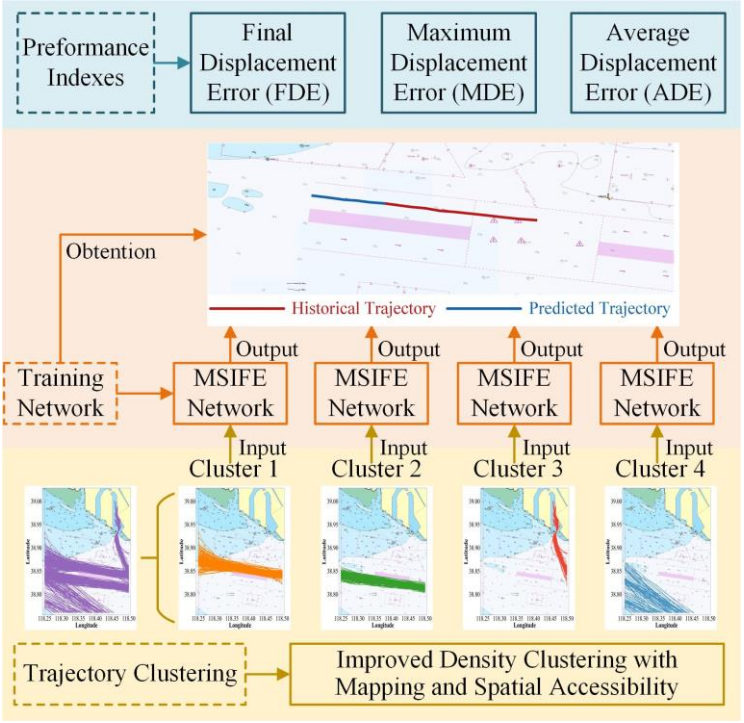

Fig. 7. The proposed IDCMSA-MSIFE prediction framework.

4.1. The proposed IDCMSA clustering algorithm

As reviewed in Section 2.1, density-based clustering methods such as DBSCAN group trajectories based on global similarity and spatial-topological structure. Although widely adopted due to their simplicity (Y. Li et al., 2024), they suffer from two main limitations: (1) the need to predefine hyperparameters (e.g., neighbourhood radius, minimum points), and (2) misclassification of trajectories that share similar shapes but differ in spatial proximity (e.g., vessels following parallel routes far apart).

The IDCMSA algorithm addresses these issues through the key steps, including: (1) trajectory embedding using classical MDS (Kalia and Paul, 2021), (2) construction of a density matrix, (3) convolution-based smoothing of the density field, and (4) cluster assignment based on direct density reachability. A 2D embedding space is deliberately adopted for three practical considerations. First, a 2D representation facilitates the subsequent construction of the density matrix and grid-based convolution operations, while also enabling intuitive visual interpretation of the clustering results. Second, it substantially reduces computational complexity and enhances clustering efficiency. Third, two dimensions are sufficient to preserve the essential similarity structure among trajectories; increasing dimensionality yields marginal gains in clustering quality and may instead introduce redundant noise. Consequently, the 2D embedding

provides an effective balance between information preservation and computational efficiency. The workflow is shown in Fig. 8. The algorithm proceeds through the following four steps.

Step 1: Mapping trajectories into low-dimensional points using MDS.

Each trajectory is first mapped to a two-dimensional point using MDS, as illustrated in Figs. 11(a)-(b). The similarity matrix *sm* is constructed using pairwise trajectory distances δ in Eq. (4) computed via DTW (Gong et al., 2022), which aligns trajectories of different lengths. MDS then finds points vtp_1, vtp_2, $\cdots$, $vtp_Q \in R^2$ such that the distances in a low-dimensional space preserve the DTW-based similarities, as expressed in Eqs. (4)-(6).

$$sm = \begin{bmatrix} \delta_{1,1} & \delta_{1,2} & \cdots & \delta_{1,Q} \\ \delta_{2,1} & \delta_{2,2} & \cdots & \delta_{2,Q} \\ \vdots & \vdots & \cdots & \vdots \\ \delta_{Q,1} & \delta_{Q,2} & \cdots & \delta_{Q,Q} \end{bmatrix} \tag{4}$$

where Q is the number of vessel trajectories in the dataset.

$$\left\| vtp_a - vtp_b \right\| \approx \delta_{a,b}, a,b \in Q, \tag{5}$$

$$\min_{vtp_1, vtp_2, \cdots, vtp_Q} \sum_{a,b,a \neq b} \left(\left\| vtp_a - vtp_b \right\| - \delta_{a,b} \right)^2 . \tag{6}$$

Step 2: Constructing the density matrix

A $u \times v$ initialization matrix is constructed according to the range of all mapping points, with each cell initially set to zero. Each point is then projected into this grid, forming a density matrix (Figs. 11(c)-(d)). The projection rules are given in Eqs. (7)-(8).

$$xLine = \left\lceil \frac{x - xMi}{xMa - xMi} \times u \right\rceil, \tag{7}$$

$$yColumn = \left\lceil \frac{y - yMi}{yMa - yMi} \times v \right\rceil. \tag{8}$$

where (x, y) represents the coordinates of the mapping point. $(xLine, yColumn)$ is the projection position of the mapping point in the density matrix. $[xMi, xMa]$ and $[yMi, yMa]$ are the value ranges of all mapping points' horizontal and vertical coordinates, respectively. $u \times v$ represents the dimension of the density matrix. $\lceil \cdot \rceil$ denotes the rounding-up calculation process. The grid cell size is determined from the average distance between mapped points, yielding the expressions for u and v are shown in Eqs. (9) and (10). Using this average distance as the grid dimension allows the cell size to adapt naturally to local point density. In dense areas, the smaller average distance yields finer grid resolution, helping to distinguish detailed features; in sparse regions, it avoids generating excessive empty cells, thus representing the spatial distribution more appropriately. Moreover, this statistically derived scale maintains sufficient spatial detail while effectively controlling the total number of grid cells, balancing efficiency and precision for subsequent density analysis and clustering. In this study, the *cellSize* is set to 0.52 for the CfdP waters and 0.63 for the CsJP waters.

$$u = \frac{xMa - xMi}{cellSize}, \tag{9}$$

$$v = \frac{yMa - yMi}{cellSize}. \tag{10}$$

Step 3: Applying convolution for density smoothing

A *ck*×*ck* convolution kernel (with *ck*=3) is applied to calculate convolution on the density matrix, as shown in Fig. 8(e). The convolution process is described in Eqs. (11)-(13). Using a small kernel avoids overly broad neighbourhoods that may distort cluster boundaries.

$$DM'(s,t) = DM(s,t) \otimes f(d,e), \tag{11}$$

$$DM'(s,t) = \sum_{d=-g}^{g} \sum_{e=-g}^{g} DM(s-d,t-e) \otimes f(d,e). \tag{12}$$

where *DM* stands for the density matrix. *DM'* represents the result of *DM* after convolution calculation. ⊗ is the convolution operator. (*s*, *t*) denotes the element in the *s-th* row and the *t-th* column in *DM*. *f*(•,•) represents a two-dimensional convolution kernel constructed based on a Gaussian function.

$$f(s,t) = \left(\frac{1}{\sqrt{2\pi}}\right)^2 \times e^{-\frac{s^2+t^2}{2}} \tag{13}$$

Step 4: Identifying clusters using direct density reachability

Finally, each mapped point's neighbourhood matrix is examined to determine whether pairs of points satisfy the direct density-reachability condition. If so, the corresponding trajectories belong to the same cluster; otherwise, they are separated (Fig. 8(f)). The size of each neighbourhood matrix matches the convolution kernel. This step corrects DBSCAN's sensitivity to spatial separation, allowing trajectories with similar shapes but different positions to be clustered consistently.

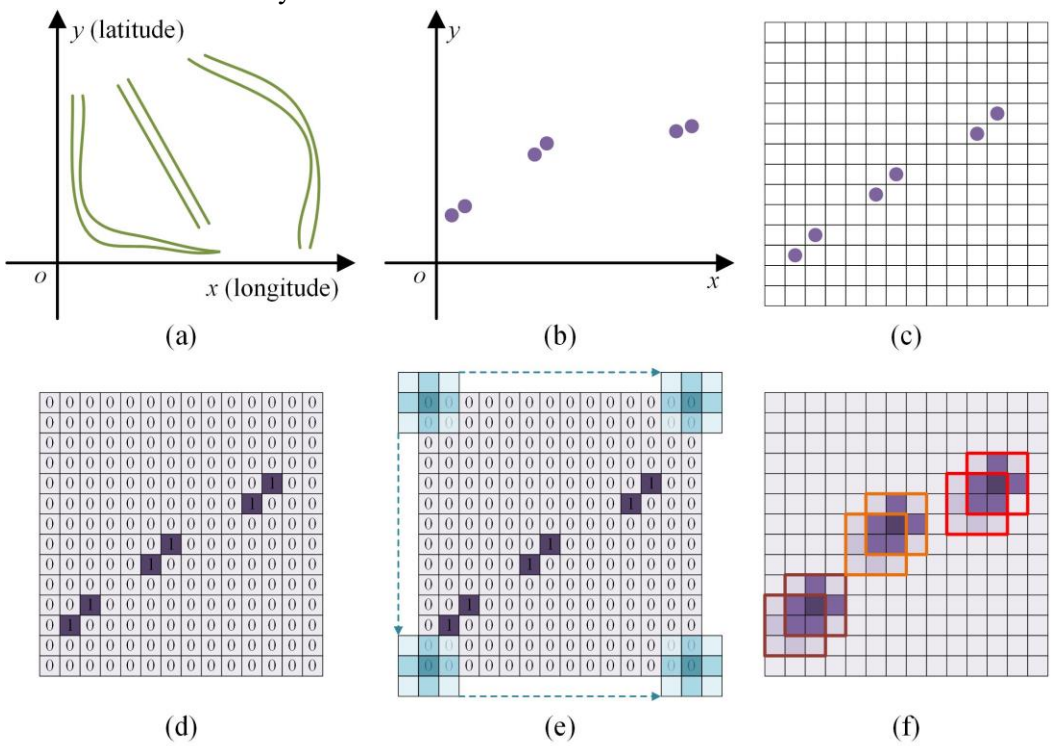

Fig. 8. The implementation flow chart of the IDCMSA vessel trajectory clustering method proposed in

this paper.

4.2. Optimized MSIFE network

Accurate vessel trajectory prediction supports intelligent maritime traffic systems by enabling collision-risk assessment, congestion forecasting, and navigation optimization. Dynamic navigation variables, including coordinates, COG, and SOG, substantially influence future vessel movement. However, many existing models use only positional data.

To integrate multi-source dynamic information, the MSIFE model is proposed, and its overall architecture is shown in Fig. 9. The network contains two main components:

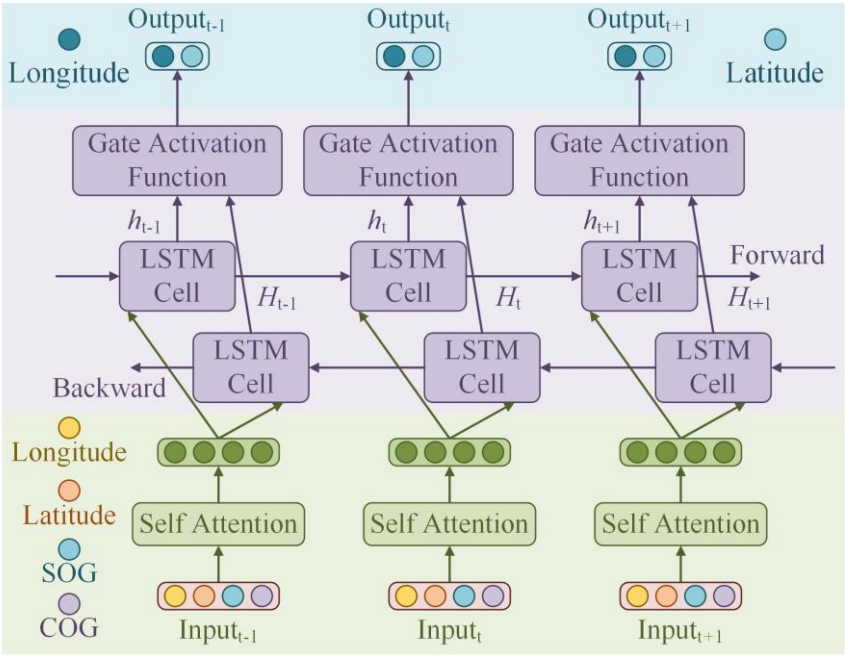

Fig. 9. The proposed MSIFE network for vessel trajectory prediction consists of two core components: a fusion view (green) and a temporal view (purple). The fusion view, implemented using a self-attention mechanism, integrates trajectory coordinates, COG, and SOG into a unified representation. The temporal view, built on a Bi-LSTM structure, captures the evolving motion patterns of vessel trajectories over time.

(1) Fusion view, a self-attention module that integrates trajectory coordinates, COG, and SOG across time; (2) Temporal view, a Bi-LSTM module that models temporal dependencies in vessel motion. The model addresses four key technical challenges:

(a) Handling inconsistent feature dimensions

Because longitude, latitude, COG, and SOG differ in scale and semantics, each variable is normalized individually, as shown in Eq. (14).

$$va_{nor} = \frac{va - 0.5 \times va_{mi}}{2 \times va_{ma} - 0.5 \times va_{mi}} \tag{14}$$

where va represents four different types of data (i.e., longitude, latitude, COG, and SOG). va_{mi} and va_{ma} are certain data types' minimum and maximum values, respectively. In Eq. (14), the maximum value is doubled and the minimum value is halved to prevent boundary effects that could reduce training accuracy.

(b) Construct the fusion view

The MSIFE model employs a self-attention mechanism to construct the fusion view, enabling

the integration of trajectory coordinates, COG, and SOG across time. As illustrated in the green portion of Fig. 9, the fusion view processes dynamic vessel data over multiple consecutive time nodes, with the number of self-attention units matching the number of time steps. Based on the interpolation procedure in Section 3.3, adjacent time nodes are spaced 5 seconds apart. The "sequence length" parameter defined in Section 5.3.1 therefore also corresponds to the number of self-attention operations. The functional form of the self-attention mechanism for fusing dynamic data at a given time node is provided in Eq. (15).

$$\left[b\alpha_t, b\beta_t, bc_t, bs_t\right] = f_{SA}\left(\left[\alpha_t, \beta_t, c_t, s_t\right]\right) \tag{15}$$

where $f_{SA}(\cdot)$ is the self-attention operation. α_t, β_t, c_t, and s_t represent the longitude, latitude, COG, and SOG at time t, respectively. The network parameters involved in self-attention and their corresponding values will be introduced in Section 5.3.1. In practical navigation scenarios, vessels may remain stationary (e.g., when anchored or berthed), leading to zero values of SOG. The MSIFE network accommodates such conditions through its self-attention mechanism, which fuses multi-source features following unified normalization (see Eq. (14)). Attention weights are computed via Query, Key, and Value projections, enabling the model to learn temporal dependencies jointly across all feature channels, including longitude, latitude, COG, and SOG. Consequently, even when SOG equals zero at certain time steps, its representation remains integrated within the attention mechanism alongside other non-zero features, thereby avoiding attention bias or degradation. This design supports robust modeling of both dynamic motion patterns and stationary states, and enhances the network's adaptability to complex and heterogeneous navigation conditions.

(c) Build a temporal view

A Bi-LSTM captures the temporal evolution of vessel movement patterns (purple block in Fig. 9). Fused features are fed into the Bi-LSTM, and the forward and backward outputs are processed through a fully connected layer to obtain predicted longitude and latitude (Eqs. (16)-(17)).

$$\left[h_t, \cdots h_{t+sq}\right] = f_{BiLSTM}\left(\left\{\left[b\alpha_t, b\beta_t, bc_t, bs_t\right], \cdots, \left[b\alpha_{t+sq}, b\beta_{t+sq}, bc_{t+sq}, bs_{t+sq}\right]\right\}, hInit, cInit\right) \tag{16}$$

$$\left[\alpha_{pre}, \beta_{pre}\right] = f_{FC}\left(\left[h_t, \cdots, h_{t+sq}\right]\right). \tag{17}$$

where $f_{BiLSTM}(\cdot)$ denotes the Bi-LSTM operation. *hInit* and *cInit* are the initialized sequences of hidden and cell states. *sq* represents the number of time nodes and has the same meaning as the 'sequence length' parameter mentioned in section 5.3.1. $f_{FC}(\cdot)$ denotes a fully connected layer, essentially a linear consumption function. α_{pre} (or β_{pre}) is the longitude (or latitude) of the predicted future time node, respectively. Because the predicted longitude and latitude are normalized, inverse normalization is required, as shown in Eqs. (18) and (19).

$$\alpha_{out} = \frac{\alpha_{pre} \times \left(2 \times \alpha_{ma} - 0.5 \times \alpha_{mi}\right)}{0.5 \times \alpha_{mi}}, \tag{18}$$

$$\beta_{out} = \frac{\beta_{pre} \times \left(2 \times \beta_{ma} - 0.5 \times \beta_{mi}\right)}{0.5 \times \beta_{mi}}. \tag{19}$$

where $[\alpha_{mi}, \alpha_{ma}]$ and $[\beta_{mi}, \beta_{ma}]$ are the value ranges of longitude and latitude, respectively. α_{pre}

and β_{pre} are the prediction results based on the training network. In Eq. (14), the maximum is doubled and the minimum is halved to reduce boundary effects, so these adjustments must be reversed during inverse normalization.

(d) Designing the loss function

In the training network, the loss function guides backpropagation by measuring the difference between predicted and actual values. To enhance this process, three correction terms are introduced. As shown in Fig. 10 (a), the first term computes the Euclidean distance between the predicted and actual points, as defined in Eq. (20).

$$loss_1 = \sqrt{\left(\alpha_C - \alpha_B\right)^2 + \left(\beta_C - \beta_B\right)^2} \tag{20}$$

where α_C and α_B are the predicted and actual longitudes, respectively. β_C and β_B denote the predicted and actual latitudes, respectively. As shown in Fig. 10 (b), the second correction term is the deflection angle generated between the predicted point, the actual point, and the previous trajectory point, with its calculation given in Eq. (21).

$$loss_2 = \arccos\left(\frac{\left(\alpha_B - \alpha_A\right)\times\left(\alpha_C - \alpha_A\right) + \left(\beta_B - \beta_A\right)\times\left(\beta_C - \beta_A\right)}{\sqrt{\left(\alpha_B - \alpha_A\right)^2 + \left(\beta_B - \beta_A\right)^2} \times \sqrt{\left(\alpha_C - \alpha_A\right)^2 + \left(\beta_C - \beta_A\right)^2}}\right) \tag{21}$$

where α_A (or β_B) denotes the longitude (or latitude) of the previous time node, respectively. As shown in Fig. 10 (b), the third correction term measures the difference between the predicted point's distance to the baseline and the actual point's distance to the same baseline, where the baseline is the line between points A and B. The corresponding expressions are provided in Eqs. (22) and (23).

$$k = \frac{\beta_A - \beta_D}{\alpha_A - \alpha_D}\left(\alpha_A \neq \alpha_D\right), \tag{22}$$

$$b = \beta_A - k \times \alpha_A. \tag{23}$$

where k and b are the slope and intercept of the baseline equation, respectively. (α_A, β_A) and (α_D, β_D) denote the coordinates of points A and B in Fig. 10, respectively. When $\alpha_A = \alpha_D$ holds, k does not exist, and the baseline equation is $x = \alpha_A$ (or $x = \alpha_D$). The distance from the predicted (Point C) or actual (Point B) point to the baseline can be obtained through Eq. (24).

$$dis = \frac{\left|k \times \alpha - \beta + b\right|}{\sqrt{1 + k^2}} \tag{24}$$

where α and β are the longitude and latitude of the predicted (or actual) point, respectively. When k does not exist, the calculation process of distance dis is shown in Eq. (25).

$$dis = \left|\alpha - \alpha_A\right| \tag{25}$$

According to Eqs. (24) and (25), the distance from the predicted and actual points to the baseline, denoted as dis_C and dis_B, can be obtained, respectively. Ideally, when $dis_B = dis_C$, the model achieves a better fit. However, the predicted and actual points may lie on the same side or on opposite sides of the baseline. When both points lie on the same side, the third correction term is defined as Eq. (26). When they lie on opposite sides, the corresponding expression is given in Eq. (27). In Eq. (26), both dis_B and dis_C are positive values. In Eq. (27), one distance is positive, and the other is negative, determined by substituting the predicted and actual points

into the baseline equation: a value greater than zero indicates a positive sign, and a value less than zero indicates a negative sign.

$$loss_3 = dis_B - dis_C\,, \tag{26}$$

$$loss_3 = dis_B + dis_C\,. \tag{27}$$

The final loss function is formed by combining the three correction terms, as shown in Eq. (28). A loss value approaching zero indicates an optimal model fit. The influence of each correction term on prediction accuracy and training stability is further examined in the ablation study presented in Section 5.4.2.

$$loss = loss_1 + loss_2 + loss_3 \tag{28}$$

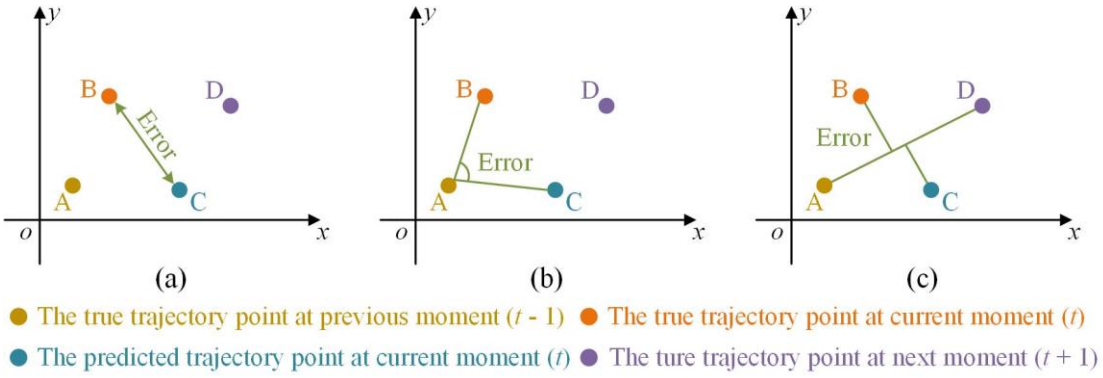

Fig. 10. Structure diagram of the loss function. In particular, the loss function of the MSIFE network consists of three correction terms. (a) to (c) illustrate the structure of Eqs. (20) to (27), respectively.

This section explains the MSIFE model by addressing four core issues involved in designing the training network. Combined with the IDCMSA clustering algorithm described in Section 4.1, it forms the IDCMSA-MSIFE framework, which strengthens the model's ability to learn from complex vessel trajectories.

5. Experimental results and analysis

This section evaluates the effectiveness and robustness of the proposed IDCMSA-MSIFE model for vessel trajectory prediction. Comparative experiments are conducted using real AIS datasets from two maritime regions (CfdP and CsJP), covering traditional neural networks, deep learning methods, and contemporary models. The influence of key parameters on prediction performance is then examined, followed by ablation studies that isolate the contribution of individual components within the proposed framework.

5.1. Performance indexes

To assess prediction accuracy, three commonly used metrics, Final Displacement Error (FDE), Maximum Displacement Error (MDE), and Average Displacement Error (ADE), are adopted (Y. Li et al., 2024). Their functional expressions are as follows.

$$FDE = \frac{\sum_{p=1}^{a}\sqrt{\left(\hat{\alpha}_p^b - \alpha_p^b\right)^2 + \left(\hat{\beta}_p^b - \beta_p^b\right)^2}}{a}, \tag{29}$$

$$MDE = \frac{\sum_{p=1}^{a}\max\limits_{1\le q\le b}\sqrt{\left(\hat{\alpha}_p^q - \alpha_p^q\right)^2 + \left(\hat{\beta}_p^q - \beta_p^q\right)^2}}{a}, \tag{30}$$

$$ADE = \frac{\sum_{p=1}^{a}\sum_{q=1}^{b}\left[\left(\hat{\alpha}_p^q - \alpha_p^q\right)^2 + \left(\hat{\beta}_p^q - \beta_p^q\right)^2\right]}{a \times b}. \tag{31}$$

where a denotes the total number of vessel trajectories in the testing dataset. b represents the number of prediction points required in each vessel trajectory. α_p^b (or β_p^b) and $\hat{\alpha}_p^b$ (or $\hat{\beta}_p^b$) indicates the longitude (or latitude) of the true and predicted coordinates at the p-th trajectory. α_p^q (or β_p^q) and $\hat{\alpha}_p^q$ (or $\hat{\beta}_p^q$) denotes the true and predicted coordinates of the q-th predicted point in the p-th trajectory.

Although these metrics capture prediction accuracy, they do not directly reflect model stability. To address this, each model is trained and evaluated ten times; mean values and standard deviations of all metrics are computed. Lower averages and smaller deviations indicate both higher accuracy and stronger robustness.

5.2. Analysis of clustering results

Traditional clustering algorithms classify trajectories solely based on trajectory-to-trajectory similarity. In contrast, IDCMSA maps each trajectory to a point and performs clustering based on density-reachability relationships between these mapped points. This enables improved discrimination of trajectory groups, particularly in complex waterways.

Fig. 11 presents the navigational channels of the two study regions, while Figs. 12 and 13 display the corresponding clustering outcomes.

In the CfdP region, IDCMSA identifies four meaningful trajectory clusters, as shown in Fig. 13. Cluster 1 corresponds to the trajectories along channel C_1 (Fig. 11 (a)), while Cluster 2 includes the trajectories in channel C_2. Cluster 3 represents the vessel entry and exit routes through channel C_3 at Caofeidian Port. Cluster 4 consists of the nearly straight-line trajectories that occur outside the main navigational channels.

In the CsJP region, IDCMSA yields seven distinct clusters (Fig. 13). Clusters 1 and 2 correspond to the movements along channels C_1 and C_2, which exhibit similar spatial patterns but opposite travel directions. Clusters 3 and 4 represent the trajectories in channels C_5-C_3 and C_4-C_8, respectively; these routes differ not only in directions but also in turning amplitudes, with the C_5-C_3 path showing smaller curvature at turning points. Clusters 5 and 6 capture the vessel movements in channels C_5-C_6 and C_7-C_8, respectively, which again follow opposing directions but share similar structural characteristics. Finally, Cluster 7 comprises the straight-line trajectories outside designated shipping channels, consistent with the behavior observed in region C_9 of Fig. 11 (b).

As illustrated in Fig. 12, the IDCMSA algorithm groups vessel trajectories in the CfdP waters into four clusters. Cluster 1 comprises approximately 34.08% of all trajectories, Cluster 2 accounts for about 36.94%, Cluster 3 makes up 12.74%, and Cluster 4 constitutes the remaining 16.24%. A similar clustering pattern is observed for the CsJP waters in Fig. 13, where seven clusters are identified. Their respective proportions are roughly 19.48% for Cluster 1, 20.88% for Cluster 2, 13.84% for Cluster 3, 12.19% for Cluster 4, 11.60% for Cluster 5, 12.31% for Cluster 6, and 8.31% for Cluster 7. These percentages reflect the distribution of different navigation patterns within the overall traffic and provide a quantitative foundation for the subsequent cluster-based trajectory prediction modeling.

Overall, these clustering results demonstrate that IDCMSA effectively identifies meaningful navigational patterns and overcomes the parameter sensitivity limitations commonly associated with classical DBSCAN.

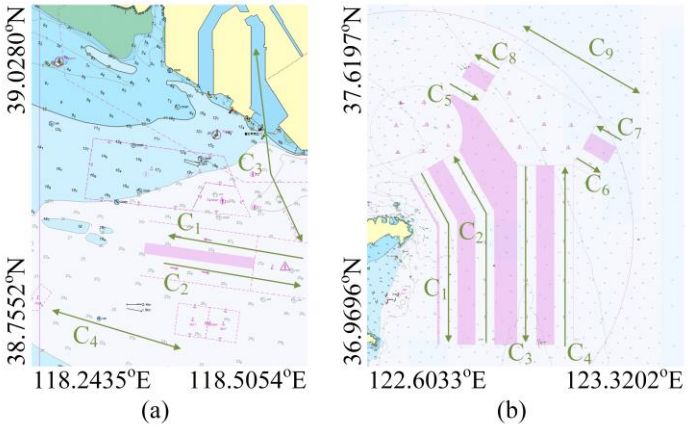

Fig. 11. Visual display of channel distribution: (a) CfdP, (b) CsJP.

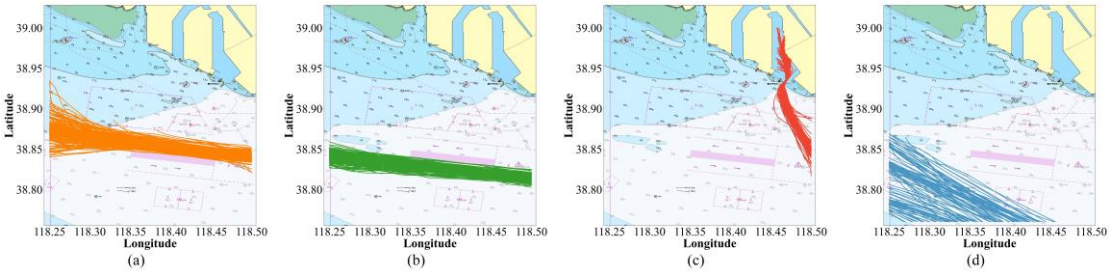

Fig. 12. Clustering results of different trajectories using IDCMSA in the CfdP water region.

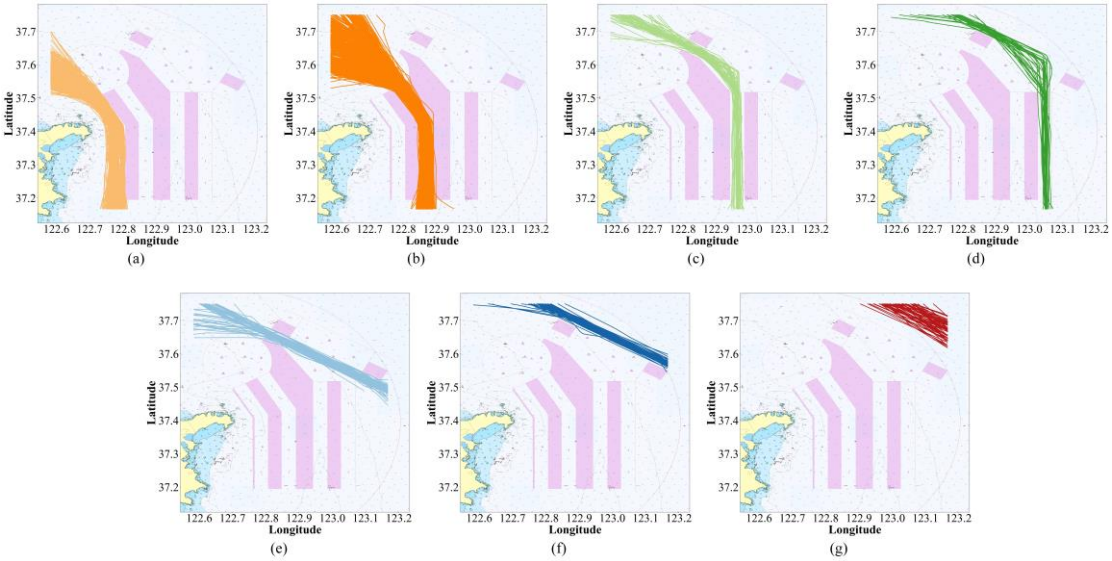

Fig. 13. Clustering results of different trajectories using the IDCMSA in the CsJP water region.

While IDCMSA significantly outperforms existing algorithms such as DBSCAN in clustering quality, it incurs a higher computational cost. As described in Section 4.1, the algorithm comprises four core steps, whose time complexities are analyzed below. Step 1 applies classical MDS to reduce the dimensionality of each trajectory, with a complexity of $O(N^2 \times L^2 + N^3)$, where N and L denote the number of trajectories and their average length, respectively. Here, $O(N^2 \times L^2)$ accounts for building the DTW distance matrix, and $O(N^3)$ corresponds to the eigendecomposition in MDS. In Step 2, all dimension-reduced points are projected into a grid matrix, with a complexity of $O(N)$. Step 3 performs convolutional smoothing on the resulting density matrix, with a complexity of $O(u \times v \times k^2)$, where k, u, and v denote the kernel size and

the dimensions of the density matrix, respectively. Finally, Step 4 carries out clustering using the concept of direct density reachability, with complexity O($N \times k^2$). In practice, clustering the trajectories in the CfdP and CsJP water regions required approximately 1.56 hours and 2.25 hours, respectively. It is noteworthy that the IDCMSA clustering algorithm was implemented and accelerated on a GPU-based parallel computing architecture. Specifically, all clustering tasks for the two water regions were performed using an NVIDIA GeForce RTX 3080 graphics card.

5.3. Experimental settings

Three sets of experiments are designed to validate the proposed approach. Section 5.3.1 examines the influence of key network parameters on prediction accuracy and stability. Section 5.3.2 presents ablation studies that isolate the contribution of individual components, including the loss-function correction terms and the IDCMSA clustering module. Section 5.3.3 evaluates the overall performance of IDCMSA-MSIFE through comparisons with seven benchmark models.

5.3.1. Network parameter settings

The MSIFE network comprises two core modules: a fusion view implemented using self-attention, which integrates trajectory coordinates, COG, and SOG; and a temporal view implemented using Bi-LSTM, which captures motion dynamics over time. Table 2 summarizes the network architecture. Three parameters, number of iterations, learning rate, and sequence length, are left unspecified because they significantly influence optimization behavior.

The iteration count and learning rate jointly determine the model's fitting performance. Excessively large values can lead to overfitting (Fig. 14 (c)), whereas values that are too small result in underfitting (Fig. 14 (a)). Appropriate parameter choices achieve a desirable balance, as illustrated in Fig. 14 (b). Sequence length, which specifies how many consecutive trajectory points are fed into the temporal module, must also be chosen carefully. Very large values hinder gradient propagation and increase computational cost, while overly small values make it difficult for the model to capture temporal evolution patterns. Table 3 lists the candidate parameter values tested in this study, and Section 5.4.1 presents the corresponding evaluation results.

Table 2. Setting information on network parameters.

Component Name	Parameter Name	Parameter Value	Parameter Name	Parameter Value
General Parameters	Optimiser Algorithm	ASGD	Iteration	#
	Learning Rate	#	Sequence Length	#
Fusion View	Input Size	4	Head Size	4
	Output Size	4	Batch Size	1
Temporal View	Input Size	4	Hidden Size	8
	Number of Hidden Layers	1	Batch Size	1
	Output Size	2	-	-

Table 3. Different network configurations of iteration, learning rate, and sequence length.

Parameter Name	Parameter Value				
Iteration	100	300	500	700	900
Learning Rate	0.0001	0.0005	0.001	0.005	0.01
Sequence Length	3	6	9	12	15

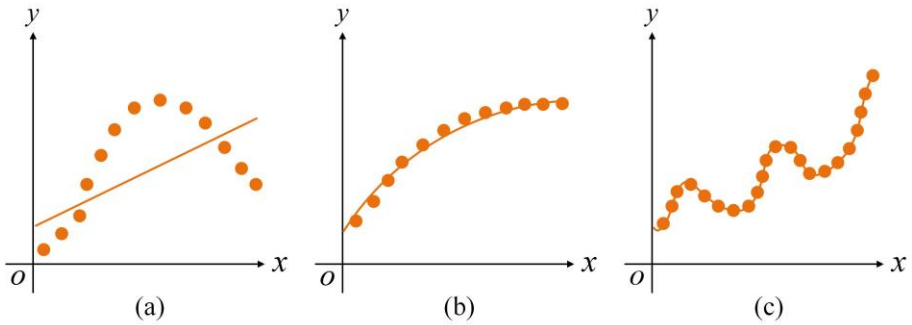

Fig. 14. Analysis of fitting in the process of network training: (a) underfitting, (b) suitability, and (c) overfitting.

5.3.2. Ablation experiment

Three ablation studies are conducted. The first examines the effect of different loss-function configurations. Based on Eqs. (20)-(27), seven variants are tested: $loss_1$, $loss_2$, $loss_3$, $loss_1 + loss_2$, $loss_1 + loss_3$, $loss_2 + loss_3$, and $loss_1 + loss_2 + loss_3$. These experiments reveal the individual and combined contributions of the three correction terms to prediction accuracy and robustness.

The second ablation study evaluates the effect of clustering. To determine whether clustering improves predictive performance, IDCMSA-MSIFE is compared with a version of MSIFE trained on the full dataset without clustering. This comparison highlights the value of category-specific training. Detailed results and analysis are provided in Section 5.4.2.

The third ablation experiment was designed to verify the effectiveness of Eq. (14) in mitigating boundary effects. Specifically, the predictive performance is evaluated when using the original dataset's minimum and maximum values directly, without applying the doubling of the maximum and halving of the minimum as prescribed by the formula.

5.3.3. Comparison with different models

Seven baseline models are used for comparison and grouped into three categories. The first category includes traditional neural networks, represented by BPNN (Jiao and Ma, 2021), which have a simple architecture and limited ability to capture complex temporal dependencies. The second category comprises classical and advanced deep learning models: RNN (Suo et al., 2020), which learns temporal patterns but is prone to gradient instability; LSTM (Gao et al., 2021), which introduces gating mechanisms to alleviate long-term dependency issues; GRU (Xue et al., 2024), a simplified and more efficient alternative to LSTM; and bidirectional architectures such as Bi-LSTM (Yang et al., 2022b) and Bi-GRU (Huang et al., 2021), which incorporate both forward and backward temporal information.

The third category includes contemporary models, represented by DBSCAN-GeoCLSTM (Y. Li et al., 2024). This method combines DBSCAN clustering with a GeoCLSTM network and has shown strong predictive capability. However, it suffers from three limitations: reliance on manually tuned DBSCAN parameters, exclusion of dynamic features such as COG and SOG, and a loss function based solely on Euclidean distance. The proposed IDCMSA-MSIFE model addresses all of these issues through improved clustering, multi-source feature fusion, and a more comprehensive loss-function design.

5.4. Prediction results in analysis

Section 5.3 examined the effects of key parameter settings, the contributions of individual components, and the baseline models used for comparison. Building on that foundation, this

section presents the detailed experimental procedures and analyses corresponding to these three aspects.

5.4.1. Prediction results of different network parameters

In vessel trajectory prediction, several core parameters of the IDCMSA-MSIFE model, namely the number of iterations, learning rate, and sequence length, directly shape the network's optimization behaviour and predictive performance. This subsection evaluates these parameters through systematic experiments.

Table 4 reports prediction accuracy and robustness under five iteration values (100, 300, 500, 700, and 900). The mean FDE, MDE, and ADE values reflect overall accuracy, whereas their standard deviations measure prediction stability. For this experiment, the learning rate and sequence length are fixed at 0.001 and 3, respectively, and the loss function adopts the full formulation ($loss_1 + loss_2 + loss_3$). As highlighted in Table 4, an iteration value of 300 yields the lowest means and standard deviations across all three metrics, indicating that the model achieves its best fitting performance at this setting.

Table 4. The prediction results (i.e., FDE, MDE, and ADE) of five various iterations (i.e., 100, 300, 500, 700, and 900) for trajectories at future time points based on IDCMSA-MSIFE in two research water regions.

Water Regions	Iteration	FDE ↓		MDE ↓		ADE ↓	
		Average Value	Standard Deviation	Average Value	Standard Deviation	Average Value	Standard Deviation
CfdP	100	0.053041	0.010697	0.079032	0.032813	0.003450	0.001203
	300	**0.038832**	**0.009452**	**0.061663**	**0.027272**	**0.002019**	**0.000974**
	500	0.043976	0.012902	0.068469	0.033944	0.002601	0.001546
	700	0.065918	0.016021	0.082595	0.029443	0.004514	0.001991
	900	0.077022	0.018718	0.090481	0.036468	0.005665	0.002459
CsJP	100	0.061051	0.015253	0.087269	0.039996	0.004771	0.003184
	300	**0.044847**	**0.013562**	**0.068071**	**0.028812**	**0.002955**	**0.002922**
	500	0.051349	0.017765	0.074198	0.030993	0.003592	0.003676
	700	0.070144	0.020131	0.099426	0.035667	0.005506	0.003994
	900	0.077256	0.021981	0.111610	0.044360	0.006452	0.004384

To analyze the learning rate, the iteration count and sequence length were fixed at 300 and 3, with the same composite loss function. Table 5 shows the results for five learning rates. The bolded values identify 0.001 as the optimal learning rate, achieving the minimal average errors and standard deviations. This confirms that overly small learning rates impede convergence, while excessively large ones cause unstable optimization.

Table 5. The prediction results of five various learning rates (i.e., 0.0001, 0.0005, 0.001, 0.005, and 0.01) for trajectories at future time points based on IDCMSA-MSIFE in two research water regions.

Water Regions	Learning Rate	FDE ↓		MDE ↓		ADE ↓	
		Average Value	Standard Deviation	Average Value	Standard Deviation	Average Value	Standard Deviation
CfdP	0.0001	0.058294	0.014182	0.086753	0.035270	0.004168	0.001842
	0.0005	0.043699	0.010755	0.067305	0.029065	0.002615	0.001214
	0.001	**0.038832**	**0.009452**	**0.061663**	**0.027272**	**0.002019**	**0.000974**
	0.005	0.044183	0.010629	0.065919	0.028620	0.002545	0.001454
	0.01	0.052043	0.012651	0.073058	0.030905	0.003212	0.001180
CsJP	0.0001	0.060649	0.017662	0.087464	0.035735	0.005104	0.003813
	0.0005	0.050642	0.015778	0.074115	0.032399	0.003592	0.003356
	0.001	**0.044847**	**0.013562**	**0.068071**	**0.028812**	**0.002955**	**0.002922**

0.005	0.050212	0.014956	0.072659	0.030445	0.003427	0.003116
0.01	0.053428	0.015058	0.078148	0.030959	0.003994	0.003184

The previous findings indicate that iteration count and learning rate must be balanced to avoid underfitting or overshooting during training. To examine the third key parameter, sequence length, experiments were conducted using values of 3, 6, 9, 12, and 15, while fixing iterations at 300 and the learning rate at 0.001. Table 6 (formerly Table 10) shows that a sequence length of 6 results in the lowest mean errors and standard deviations, demonstrating the highest prediction accuracy and robustness. This pattern reflects the nature of temporal modelling: If sequence length is too large, gradient propagation in the temporal module becomes difficult, reducing predictive accuracy. If the sequence length is too small, the network cannot sufficiently capture the evolving temporal patterns of vessel motion.

Thus, a sequence length of 6 provides the best balance between temporal richness and computational feasibility, enabling the IDCMSA-MSIFE model to learn trajectory dynamics effectively.

Table 6. The prediction results of five various sequence lengths (i.e., 3, 6, 9, 12, and 15) for trajectories at future time nodes based on IDCMSA-MSIFE in two research water regions.

Water Regions	Sequence Length	FDE ↓		MDE ↓		ADE ↓	
		Average Value	Standard Deviation	Average Value	Standard Deviation	Average Value	Standard Deviation
CfdP	3	0.038832	0.009452	0.061663	0.027272	0.002019	0.000974
	6	**0.029974**	**0.010086**	**0.047012**	**0.010886**	**0.001392**	**0.001226**
	9	0.036539	0.009852	0.055066	0.024398	0.002249	0.001031
	12	0.044011	0.011706	0.076568	0.040597	0.002767	0.001162
	15	0.049458	0.010756	0.092245	0.033826	0.003311	0.001298
CsJP	3	0.044847	0.013562	0.068071	0.028812	0.002955	0.002922
	6	**0.037110**	**0.012519**	**0.052108**	**0.018987**	**0.002196**	**0.002071**
	9	0.043202	0.014477	0.060658	0.036565	0.002770	0.003053
	12	0.047151	0.013943	0.078448	0.032042	0.003215	0.003235
	15	0.052785	0.015782	0.103813	0.047620	0.003851	0.003682

5.4.2. Prediction results of the ablation experiment

Several essential components of the IDCMSA-MSIFE model play a decisive role in enhancing its ability to process complex trajectory data. Section 5.3.2 identified which components are expected to contribute most to prediction accuracy and stability. The ablation experiments reported in this section empirically verify these contributions.

Table 7 presents the prediction performance obtained using different combinations of the three correction terms in the loss function. The bolded results show that the full loss function ($loss_1$+$loss_2$+$loss_3$) achieves the highest accuracy and robustness. All pairwise combinations perform worse than the complete loss, while the worst results occur when only individual correction terms are used. Among the individual terms, $loss_1$ performs best, $loss_2$ is moderate, and $loss_3$ contributes the least. These findings indicate that the Euclidean distance term is the most influential factor during backpropagation, followed by the deflection-angle term, whereas the baseline-distance difference provides more limited, but still beneficial, information. The results verify that incorporating all three terms leads to the most effective optimization behaviour.

The second ablation study evaluates whether clustering the trajectories prior to training improves predictive performance. Table 8 compares the IDCMSA-MSIFE model, which trains separate networks for each cluster, with MSIFE trained on all trajectories without clustering.

The bolded results show that IDCMSA-MSIFE consistently achieves lower mean errors and smaller standard deviations. This confirms that trajectory-specific modelling substantially enhances both accuracy and stability.

The third ablation experiment validates the effectiveness of boundary adjustment in the data normalization step. Specifically, we compare two normalization strategies: one directly employs the original minimum and maximum values, while the other first doubles the maximum and halves the minimum to mitigate boundary effects. Table 9 presents the trajectory prediction results for both water regions under the two strategies. The bold values indicate that the boundary-adjusted normalization achieves lower mean errors and smaller standard deviations across all metrics. This confirms that appropriately scaling the extreme values enhances both prediction accuracy and stability.

Table 7. The prediction results of seven various loss functions (i.e., $loss_1$, $loss_2$, $loss_3$, $loss_1 + loss_2$, $loss_1 + loss_3$, $loss_2 + loss_3$, and $loss_1 + loss_2 + loss_3$) for vessel trajectories at future time points based on IDCMSA-MSIFE in two research water regions.

Water Regions	Loss Function	FDE ↓		MDE ↓		ADE ↓	
		Average Value	Standard Deviation	Average Value	Standard Deviation	Average Value	Standard Deviation
CfdP	$loss_1$	0.038648	0.013707	0.054094	0.015007	0.002059	0.001408
	$loss_2$	0.039902	0.013098	0.055747	0.015210	0.002190	0.001393
	$loss_3$	0.041094	0.014176	0.057661	0.016096	0.002303	0.001474
	$loss_1 + loss_2$	0.032967	0.011702	0.049779	0.010966	0.001584	0.001298
	$loss_1 + loss_3$	0.033698	0.011974	0.050871	0.011096	0.001649	0.001207
	$loss_2 + loss_3$	0.035009	0.012446	0.051250	0.012479	0.001808	0.001314
	$loss_1 + loss_2 + loss_3$	**0.029974**	**0.010086**	**0.047012**	**0.010886**	**0.001392**	**0.001226**
CsJP	$loss_1$	0.045049	0.017104	0.061194	0.021070	0.002997	0.002177
	$loss_2$	0.046761	0.017083	0.063038	0.021966	0.003145	0.002301
	$loss_3$	0.049002	0.017907	0.064121	0.020521	0.003298	0.002265
	$loss_1 + loss_2$	0.039664	0.013092	0.055026	0.020749	0.002564	0.002108
	$loss_1 + loss_3$	0.040795	0.013002	0.056773	0.020492	0.002610	0.002097
	$loss_2 + loss_3$	0.041906	0.014713	0.058020	0.020048	0.002774	0.002005
	$loss_1 + loss_2 + loss_3$	**0.037110**	**0.012519**	**0.052108**	**0.018987**	**0.002196**	**0.002071**

Table 8. The prediction results of two combination methods (i.e., MSIFE and IDCMSA-MSIFE) for vessel trajectories at future time points in two research water regions.

Water Regions	Methods	FDE ↓		MDE ↓		ADE ↓	
		Average Value	Standard Deviation	Average Value	Standard Deviation	Average Value	Standard Deviation
CfdP	MSIFE	0.052564	0.015773	0.087493	0.020018	0.004012	0.002502
	IDCMSA-MSIFE	**0.029974**	**0.010086**	**0.047012**	**0.010886**	**0.001392**	**0.001226**
CsJP	MSIFE	0.063589	0.016728	0.099834	0.025483	0.005764	0.003450
	IDCMSA-MSIFE	**0.037110**	**0.012519**	**0.052108**	**0.018987**	**0.002196**	**0.002071**

Table 9. The prediction results under two normalization schemes (w/ vs. w/o boundary adjustment) for vessel trajectories at future time points in two research water regions.

Water Regions	Methods	FDE ↓		MDE ↓		ADE ↓	
		Average Value	Standard Deviation	Average Value	Standard Deviation	Average Value	Standard Deviation
CfdP	w/o boundary	0.032205	0.012604	0.058002	0.014913	0.002187	0.001964

	adjustment w/ boundary adjustment	**0.029974**	**0.010086**	**0.047012**	**0.010886**	**0.001392**	**0.001226**
CsJP	w/o boundary adjustment	0.046342	0.015277	0.067220	0.021982	0.003319	0.002963
	w/ boundary adjustment	**0.037110**	**0.012519**	**0.052108**	**0.018987**	**0.002196**	**0.002071**

5.4.3. Prediction results of different models

This section compares IDCMSA-MSIFE with seven alternative prediction models, classified into three groups, based on their characteristics:

(1) a traditional neural network (BPNN),

(2) five deep learning methods (RNN, LSTM, GRU, Bi-LSTM, and Bi-GRU), and

(3) an innovative clustering-based model (DBSCAN-GeoCLSTM).

Since these approaches also rely on learning temporal patterns from historical trajectories, several parameter settings overlap with those of IDCMSA-MSIFE. To ensure fair comparison, all shared parameters are aligned across models, as summarised in Table 10.

Table 10. Setting information on network parameters for other comparison methods.

Methods	Component Name	Parameter Name	Parameter Value	Parameter Name	Parameter Value
BPNN	-	Optimiser Algorithm	ASGD	Iteration	500
		Learning Rate	0.001	Input Size	12
		Hidden Size	24	Output Size	2
RNN, LSTM, GRU, Bi-LSTM, Bi-GRU	-	Optimiser Algorithm	ASGD	Iteration	500
		Learning Rate	0.001	Sequence Length	6
		Input Size	2	Hidden Size	8
		Output Size	2	Batch Size	1
		Number of Hidden Layers	1	-	-
DBSCAN-GeoCLSTM	General Parameters	Optimiser Algorithm	ASGD	Iteration	500
		Learning Rate	0.001	Sequence Length	6
		Geohash Coding Size	40	Feature Vector Size	14
	Dimension Reduction View	Input Size	80	Hidden Size	320
		Output Size	14	-	-
	Prediction View	Input Size	14	Hidden Size	28
		Output Size	14	Number of Hidden Layers	1
		Batch Size	1	-	-
	Dimension Promotion View	Input Size	14	Hidden Size	28
		Output Size	80	-	-

Mapping View	Input Size	1	Hidden Size	6
	Output Size	2	Number of Hidden Layers	1
	Batch Size	1	-	-

Prediction performance is assessed using both qualitative and quantitative analyses. Qualitatively, the predicted trajectories of representative vessels from each study area are visualized on electronic nautical charts (Figs. 15-18) and compared with the corresponding real trajectories. Quantitatively, the average and standard deviation of FDE, MDE, and ADE are computed for each model across all trajectories (Table 11).

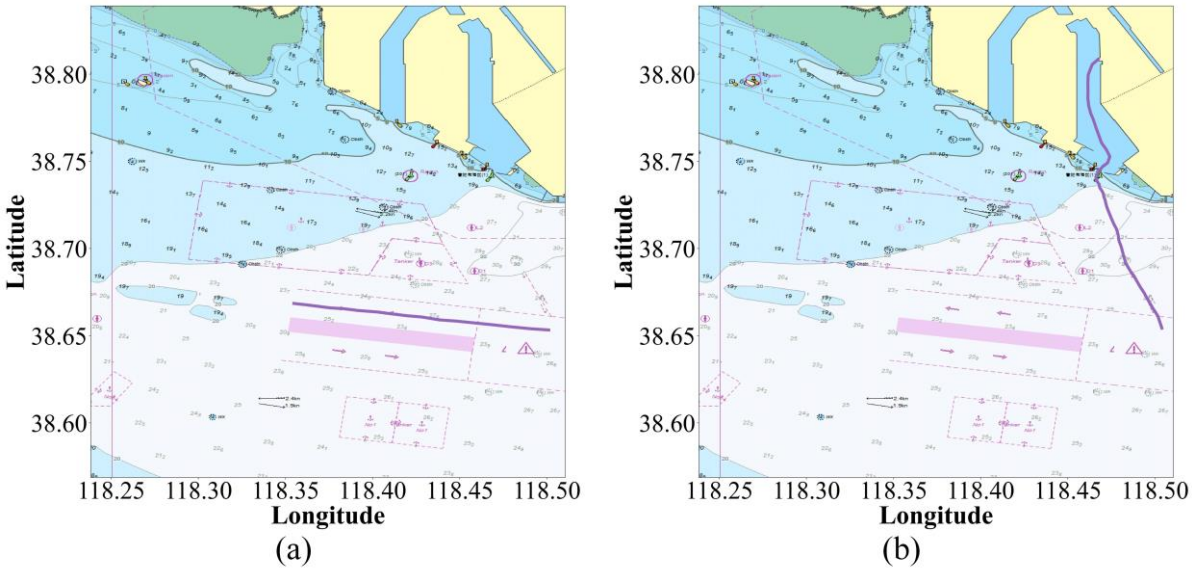

Fig. 15. Visualization of representative vessel trajectories in the CfdP water region.

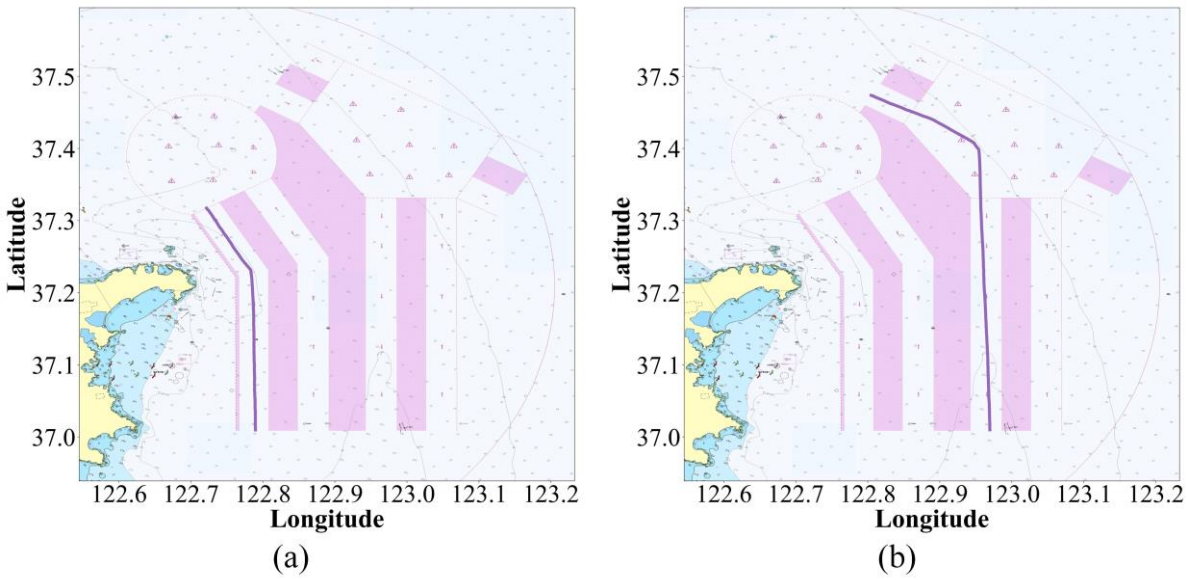

Fig. 16. Visualization of representative vessel trajectories in the CsJP water region.

Among traditional and deep learning models, Bi-LSTM achieves the best performance. Its predicted trajectories most closely resemble the real paths, and it achieves the lowest means

and standard deviations across all metrics. The ranking of the remaining methods, from the best to the worst, is: Bi-GRU, LSTM, GRU, RNN, and BPNN in order. These results reflect inherent differences in their architectures: BPNN lacks temporal modelling capability, resulting in the weakest performance. RNN improves upon this by incorporating hidden and cell states, yet remains vulnerable to gradient instability. LSTM and GRU mitigate long-term dependency problems and therefore perform better than RNN. Bi-LSTM and Bi-GRU further enhance prediction by modelling forward and backward temporal dependencies simultaneously, capturing global trajectory patterns more effectively.

Particular attention is given to comparing IDCMSA-MSIFE with DBSCAN-GeoCLSTM, as both adopt a clustering-plus-prediction architecture. However, IDCMSA-MSIFE consistently outperforms DBSCAN-GeoCLSTM in both accuracy and stability. The superior performance arises from three key improvements: 1) Clustering without manual parameter tuning, enabled by IDCMSA, 2) Integration of dynamic features (COG and SOG), which are absent in GeoCLSTM, and 3) A more comprehensive loss function, which incorporates three correction terms instead of the sole use of Euclidean distance.

Overall, IDCMSA-MSIFE provides the most accurate and stable predictions among all tested models, demonstrating its strong capability for modelling vessel movement in complex maritime environments.

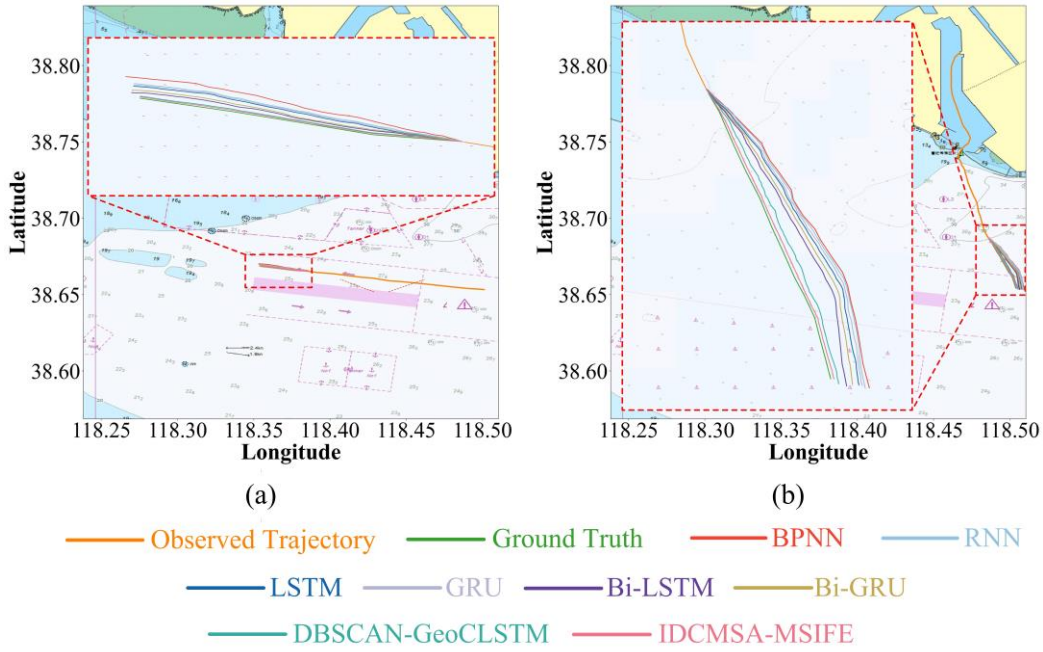

Fig. 17. Visual comparison of forecast results for two representative vessel trajectories based on eight methods in CfdP.

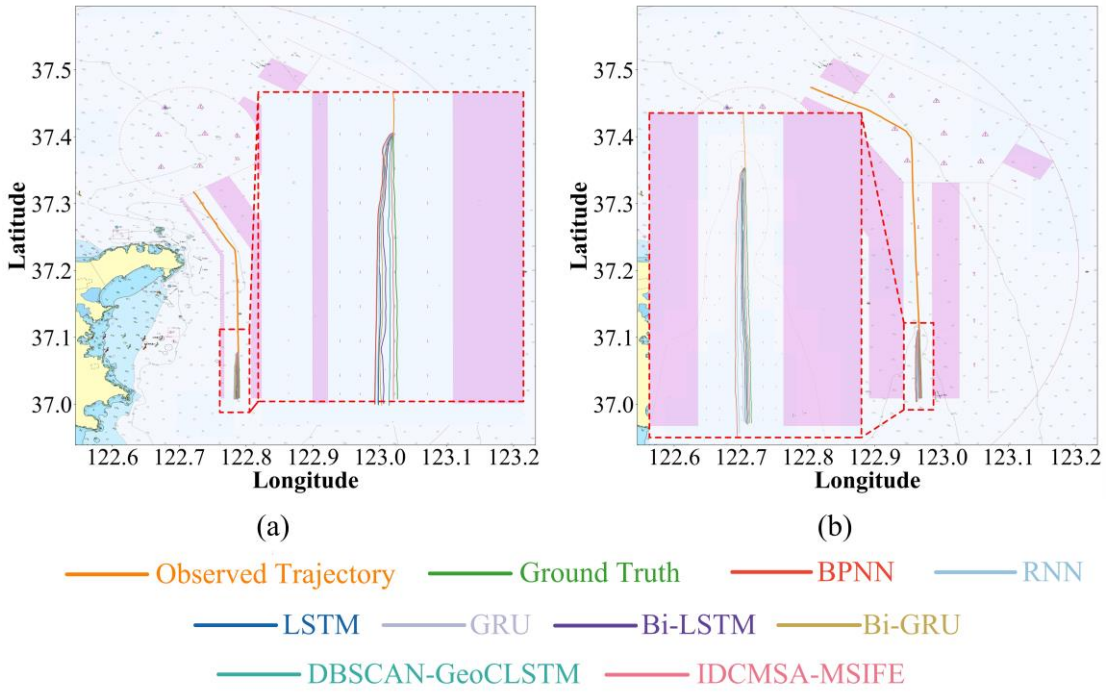

Fig. 18. Visual comparison of prediction results for two representative vessel trajectories based on eight methods in CsJP.

Table 11. The prediction results of vessel trajectories at future time points based on eight methods in two research water regions.

Water Regions	Methods	FDE ↓		MDE ↓		ADE ↓	
		Average Value	Standard Deviation	Average Value	Standard Deviation	Average Value	Standard Deviation
CfdP	BPNN	0.193456	0.057897	0.232346	0.086343	0.057687	0.029564
	RNN	0.136459	0.034589	0.165478	0.060235	0.033244	0.008439
	LSTM	0.100345	0.020345	0.130233	0.043425	0.009345	0.005123
	GRU	0.110324	0.028685	0.145679	0.047312	0.010184	0.005209
	Bi-LSTM	0.073982	0.017094	0.109491	0.031606	0.007055	0.003977
	Bi-GRU	0.080995	0.019081	0.116482	0.038643	0.007962	0.004458
	DBSCAN-GeoCLSTM	0.038328	0.012336	0.058771	0.019447	0.002055	0.001967
	IDCMSA-MSIFE	**0.029974**	**0.010086**	**0.047012**	**0.010886**	**0.001392**	**0.001226**
CsJP	BPNN	0.237456	0.058678	0.283465	0.126455	0.058667	0.026756
	RNN	0.154323	0.039233	0.206785	0.093425	0.040214	0.009213
	LSTM	0.126452	0.029456	0.164356	0.063456	0.011546	0.007234
	GRU	0.137012	0.030916	0.176209	0.069810	0.013778	0.008554
	Bi-LSTM	0.087964	0.020447	0.120592	0.048836	0.009857	0.005013
	Bi-GRU	0.096332	0.028736	0.131895	0.050224	0.010989	0.005964
	DBSCAN-GeoCLSTM	0.049346	0.015449	0.063570	0.029772	0.003922	0.003274
	IDCMSA-MSIFE	**0.037110**	**0.012519**	**0.052108**	**0.018987**	**0.002196**	**0.002071**

Beyond predictive performance, the computational cost of the complete training process for each compared model was evaluated under identical hardware conditions (a single NVIDIA GeForce RTX 3080 GPU). As reported in Table 12, conventional neural networks (e.g., BPNN and RNN) and their variants (LSTM, GRU, Bi-LSTM, and Bi-GRU) exhibit relatively short training times; however, they incur substantially higher prediction errors than the two more advanced frameworks, DBSCAN-GeoCLSTM and the proposed IDCMSA-MSIFE.

The latter two approaches integrate trajectory clustering and more sophisticated network architectures, which inevitably incur increased training time. Nevertheless, in practical

maritime monitoring and decision-support scenarios, model training is typically performed offline. After deployment, the trained models can generate predictions for incoming trajectory data within milliseconds, thereby fully satisfying real-time operational requirements.

Overall, IDCMSA-MSIFE achieves a favorable balance between computational cost and predictive accuracy, demonstrating a compelling trade-off that highlights its practicality and strong potential for real-world maritime applications.

Table 12. Comparative analysis of total training time across different methods on vessel trajectory data from two water regions

Water Regions	Methdos	Training Time (s)
CfdP	BPNN	487
	RNN	1204
	LSTM	2113
	GRU	1827
	Bi-LSTM	3126
	Bi-GRU	2766
	DBSCAN-GeoCLSTM	25128
	IDCMSA-MSIFE	30867
CsJP	BPNN	669
	RNN	1805
	LSTM	3027
	GRU	2642
	Bi-LSTM	4287
	Bi-GRU	3628
	DBSCAN-GeoCLSTM	34167
	IDCMSA-MSIFE	42625

5.4.4 Analysis of representative navigation scenarios

To further validate the predictive advantage of the proposed framework in complex and realistic real-world navigation scenarios, this section presents two representative case studies, each based on a single trajectory selected from the CfdP and CsJP waters, respectively. The selected cases involve challenging navigation scenarios, including vessel turning, speed variation, and interactions in dense traffic, thereby providing an intuitive illustration of the framework's performance under highly dynamic conditions.

(1) Case Study 1: Vessel Departure in CfdP Waters. The first case involves a vessel departing from a port within the CfdP waters. The spatial distribution of this trajectory is shown in Fig. 15(b), and the prediction results from different methods are compared in Fig. 17(b). This trajectory belongs to Cluster 3 of the CfdP region, which primarily represents turning behaviors associated with port entry and exit. During the departure process, the vessel performs multiple turning maneuvers.

The prediction generated by the proposed framework (pink curve) closely follows the ground-truth trajectory and accurately reproduces the curvature variations after each turn. This result indicates that the IDCMSA module effectively groups trajectories sharing similar spatial patterns, such as turning movements near port entrances and exits. As a consequence, the subsequent MSIFE network is able to concentrate on learning cluster-specific dynamic characteristics, including changes in COG and SOG during maneuvering. The fusion of COG and SOG information plays a critical role in enabling the network to capture these motion dynamics.

This case demonstrates that the IDCMSA-MSIFE framework can effectively model complex

vessel maneuvers at critical waypoints, such as port approaches and departures. The resulting accurate trajectory prediction is particularly valuable for port traffic organization, collision avoidance, and berthing schedule coordination.

(2) Case Study 2: Trajectory in a Congested Waterway Intersection (CsJP). The second case examines a vessel trajectory within a complex intersection area of the CsJP waters, where multiple channels converge. The trajectory distribution is shown in Fig. 16(b), and the corresponding prediction results are compared in Fig. 18(b). This trajectory belongs to Cluster 3 of CsJP and is characterized by frequent subtle course adjustments, reflecting vessel interactions and adaptations to channel geometry rather than straight-line navigation.

The proposed framework again produces a prediction (pink curve) that closely matches the overall trend of the true trajectory. This performance can be attributed to the strong temporal modeling capability of the Bi-LSTM module within the MSIFE network, together with the guidance of the tailored loss function that incorporates angle deviation and baseline distance correction terms. These components enable the model to learn and reproduce fine-grained navigational corrections embedded in the trajectory.

This case highlights the robustness of the proposed framework in congested and interactive waterways. By reducing trajectory heterogeneity through clustering and adopting a behavior-aware optimization strategy, the model effectively captures subtle course variations that arise from complex traffic interactions. The ability to anticipate such minor adjustments is essential for early warning of close-quarters situations and collision risk mitigation.

Overall, the two case studies demonstrate that the IDCMSA-MSIFE framework not only achieves superior quantitative accuracy but also reliably predicts critical vessel maneuvers, such as turning and speed variation, and captures fine-scale path corrections in congested waters. These results confirm the practical value of the integrated design, which combines density-aware clustering, multi-source feature fusion, and a tailored loss function within a unified two-stage architecture.

5.5. Management and policy implications for ocean and coastal operations

The proposed cluster-aware vessel trajectory prediction framework offers important implications for both maritime management practice and policy development in ocean and coastal operational contexts. By enhancing the accuracy and stability of trajectory forecasting in heterogeneous traffic environments, the framework provides a reliable analytical foundation for informed decision-making by maritime administrations, port authorities, Vessel Traffic Service (VTS) operators, and coastal regulators.

From an operational management perspective, the ability to distinguish heterogeneous vessel movement patterns through trajectory clustering enables more targeted and adaptive traffic management strategies. Instead of treating maritime traffic as a homogeneous flow, cluster-level trajectory prediction enables the identification of behavior-specific navigation characteristics across different vessel groups and operational contexts. This supports the design, evaluation, and dynamic adjustment of traffic separation schemes, fairway layouts, routing measures, and speed management policies, particularly in congested coastal waters and port-approach areas where navigational complexity and safety risks are elevated.

In the domain of maritime safety management, the improved predictive performance of the framework strengthens the capacity for proactive risk identification and mitigation. Short-term trajectory forecasts can be used to support early warning of potential collision risks, near-miss

situations, and abnormal maneuvering behaviors. When integrated into shore-based monitoring platforms or VTS systems, such predictive capability facilitates a transition from predominantly reactive incident response toward preventive and anticipatory safety management, aligning with international best practices in maritime safety governance.

The framework also contributes to operational efficiency and environmental performance. More accurate anticipation of vessel movements enables improved traffic flow coordination, congestion management, and arrival time optimization. These capabilities are particularly relevant for ports and coastal authorities seeking to balance safety, efficiency, and environmental objectives (Wang et al., 2023). By supporting smoother traffic operations and reducing unnecessary maneuvering, waiting times, and stop-go behavior, the proposed approach indirectly contributes to fuel consumption reduction and emissions mitigation in busy maritime corridors (Chen et al., 2025; W. Zhang et al., 2025).

From a policy and governance perspective, the proposed framework supports evidence-based maritime decision-making by relying primarily on widely available AIS data without requiring additional onboard equipment or intrusive data collection. This characteristic enhances its scalability and transferability across different regulatory regimes and geographic regions. The framework can inform maritime spatial planning, the assessment of traffic management interventions, and the development of adaptive regulatory strategies that reflect observed vessel behavior patterns rather than static or assumption-driven models.

Finally, as ocean and coastal operations increasingly undergo digital transformation, the integration of cluster-aware predictive analytics aligns with broader policy initiatives related to intelligent transportation systems, smart ports, and maritime digitalization. The proposed approach provides a practical pathway for embedding advanced data-driven intelligence into operational and governance frameworks, supporting long-term objectives of safe, efficient, and sustainable ocean and coastal management.

To illustrate these implications, a representative application scenario has been added for the intelligent traffic management of the Deepwater Channel at the Yangtze Estuary, China. This waterway is characterized by high traffic density, mixed vessel types, frequent crossing movements, and strong tidal influences, resulting in substantial operational complexity.

In this scenario, the IDCMSA-MSIFE framework can be integrated into the local VTS center to establish an intelligent situational awareness and early-warning system. First, the IDCMSA module performs clustering on historical AIS data to identify dominant traffic flow patterns, including bidirectional main routes, crossing streams, turning zones for large vessels, and access channels to anchorages. Second, for each identified cluster, the MSIFE network is trained using cluster-specific historical trajectories and applied in real time to generate short-term trajectory predictions (e.g., 10-30 minutes ahead) based on dynamic features such as position, speed, and course. Third, predicted trajectories are used to compute future encounter indicators, including Distance to Closest Point of Approach (DCPA) and Time to Closest Point of Approach (TCPA), enabling early identification of potential conflict zones and high-risk situations. Fourth, during peak traffic periods or adverse environmental conditions, the system can support dynamic traffic management by recommending speed adjustments, staggered entry into complex segments, or temporary sequencing of vessel movements.

6. Conclusions

This study proposes IDCMSA-MSIFE, a unified framework for vessel trajectory prediction

that combines an enhanced clustering method with a multi-source fusion network. The IDCMSA algorithm improves density-based clustering by eliminating the need for manually defined hyperparameters and demonstrating increased robustness to noise in complex waterways. Building on this clustering step, the MSIFE network leverages trajectory coordinates, COG, and SOG to learn rich motion features, while a three-term loss function enhances model stability during optimization.

Experiments on real AIS datasets from two coastal regions show that IDCMSA-MSIFE achieves consistently higher accuracy and stability than a range of traditional and deep learning benchmarks, confirming its effectiveness in modelling complex vessel movement patterns. The integration of improved clustering with multi-source fusion offers a promising direction for advancing data-driven maritime traffic intelligence.

While the proposed framework shows promising results in current experimental settings, further research could focus on two key directions to enhance its practical applicability. First, the computational efficiency of the IDCMSA clustering algorithm requires improvement for large-scale, high-frequency AIS data processing, especially during density-reachability analysis and convolutional smoothing. Implementing the algorithm on GPU-based parallel architectures, utilizing strategies such as block-wise computation of the grid density matrix and CUDA kernel optimization, could significantly reduce runtime. This would improve scalability and support real-time analysis in extensive areas like port clusters and coastal networks. Second, current models are primarily trained on routine sailing patterns, yet real-world trajectories often contain sudden deviations caused by collision avoidance, emergency maneuvers, or external disruptions. These irregular and transient behaviors are not well captured by existing prediction approaches. Future work should therefore integrate such uncertainty-aware modeling, for instance, through anomaly detection modules, intent-aware context weighting, or more robust loss functions and network designs. These advancements would strengthen the model's robustness and generalization in dynamic maritime environments, supporting safer and more efficient traffic management.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work presented in this study is financially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 42407114 and No. 52571407.

References

Aggarwal, A., Mittal, M., Battineni, G., 2021. Generative adversarial network: An overview of theory and applications. International Journal of Information Management Data Insights 1, 100004. https://doi.org/10.1016/j.jjimei.2020.100004

Alam, M.M., Spadon, G., Etemad, M., Torgo, L., Milios, E., 2024. Enhancing short-term vessel trajectory prediction with clustering for heterogeneous and multi-modal movement patterns. Ocean Engineering 308, 118303. https://doi.org/10.1016/j.oceaneng.2024.118303

Ali, Y., Aly, H.H., 2024. Short term wind speed forecasting using artificial and wavelet neural networks with and without wavelet filtered data based on feature selections technique.

Engineering Applications of Artificial Intelligence 133, 108201. https://doi.org/10.1016/j.engappai.2024.108201

Antoniou, A., Storkey, A., Edwards, H., 2018. Augmenting Image Classifiers Using Data Augmentation Generative Adversarial Networks, in: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (Eds.), Artificial Neural Networks and Machine Learning – ICANN 2018. Springer International Publishing, Cham, pp. 594–603. https://doi.org/10.1007/978-3-030-01424-7_58

Baeza, V.M., Ortiz, F., Garcia, S.H., Lagunas, E., Baeza, V.M., Ortiz, F., Garcia, S.H., Lagunas, E., 2022. Enhanced Communications on Satellite-Based IoT Systems to Support Maritime Transportation Services. Sensors 22. https://doi.org/10.3390/s22176450

Bezdek, J.C., Ehrlich, R., Full, W., 1984. FCM: The fuzzy *c*-means clustering algorithm. Computers & Geosciences 10, 191–203. https://doi.org/10.1016/0098-3004(84)90020-7

Bureva, V., Sotirova, E., Popov, S., Mavrov, D., Traneva, V., 2017. Generalized Net of Cluster Analysis Process Using STING: A Statistical Information Grid Approach to Spatial Data Mining, in: Christiansen, H., Jaudoin, H., Chountas, P., Andreasen, T., Legind Larsen, H. (Eds.), Flexible Query Answering Systems. Springer International Publishing, Cham, pp. 239–248. https://doi.org/10.1007/978-3-319-59692-1_21

Chang, Y., Ma, J., Sun, L., Ma, Z., Zhou, Y., 2024. Vessel Traffic Flow Prediction in Port Waterways Based on POA-CNN-BiGRU Model. Journal of Marine Science and Engineering 12, 2091. https://doi.org/10.3390/jmse12112091

Chen, C.L.P., Feng, S., 2020. Generative and Discriminative Fuzzy Restricted Boltzmann Machine Learning for Text and Image Classification. IEEE Transactions on Cybernetics 50, 2237–2248. https://doi.org/10.1109/TCYB.2018.2869902

Chen, W., Zhang, W., Chen, J., Wu, X., Geng, J., Shi, J., Xu, J., 2025. Performance evaluation of low-carbon development at key nodes in the maritime supply chain. International Journal of Logistics Research and Applications 28, 1522–1545. https://doi.org/10.1080/13675567.2024.2357234

Chen, X., Pan, L., 2018. A Survey of Graph Cuts/Graph Search Based Medical Image Segmentation. IEEE Reviews in Biomedical Engineering 11, 112–124. https://doi.org/10.1109/RBME.2018.2798701

Chen, Y., Hou, F., Dong, S., Guo, L., Xia, T., He, G., 2022. Reliability evaluation of corroded pipeline under combined loadings based on back propagation neural network method. Ocean Engineering 262, 111910. https://doi.org/10.1016/j.oceaneng.2022.111910

de Rosa, G.H., Papa, J.P., 2021. A survey on text generation using generative adversarial networks. Pattern Recognition 119, 108098. https://doi.org/10.1016/j.patcog.2021.108098

Deng, Z., Cui, J., Zhu, Z., Dan, Y., Xie, H., Hu, Y., 2025. Adaptive trajectory tracking control of underactuated ships using parameter prediction based neural network. Ocean Engineering 329, 121188. https://doi.org/10.1016/j.oceaneng.2025.121188

Ding, L., Li, C., Jin, D., Ding, S., 2024. Survey of spectral clustering based on graph theory. Pattern Recognition 151, 110366. https://doi.org/10.1016/j.patcog.2024.110366

Du, J., Lu, D., Li, F., Liu, K., Qiu, X., 2025. Trajectory Prediction and Intention Recognition Based on CNN-GRU. IEEE Access 13, 26945–26957. https://doi.org/10.1109/ACCESS.2025.3539931

Fu, Y., Liu, X., Sarkar, S., Wu, T., 2021. Gaussian mixture model with feature selection: An

embedded approach. Computers & Industrial Engineering 152, 107000. https://doi.org/10.1016/j.cie.2020.107000

Gan, L., Gao, Z., Zhang, X., Xu, Y., Liu, R.W., Xie, C., Shu, Y., 2025. Graph neural networks enabled accident causation prediction for maritime vessel traffic. Reliability Engineering & System Safety 257, 110804. https://doi.org/10.1016/j.ress.2025.110804

Gan, L., Ye, B., Huang, Z., Xu, Y., Chen, Q., Shu, Y., 2023. Knowledge graph construction based on ship collision accident reports to improve maritime traffic safety. Ocean & Coastal Management 240, 106660. https://doi.org/10.1016/j.ocecoaman.2023.106660

Gao, D., Zhu, Y., Zhang, J., He, Y., Yan, K., Yan, B., 2021. A novel MP-LSTM method for ship trajectory prediction based on AIS data. Ocean Engineering 228, 108956. https://doi.org/10.1016/j.oceaneng.2021.108956

Gong, J., Li, H., Jiao, H., Yang, Z., 2025. Uncertainty-aware ship trajectory prediction via Spatio-Temporal Graph Transformer. Transportation Research Part E: Logistics and Transportation Review 203, 104315. https://doi.org/10.1016/j.tre.2025.104315

Gong, L., Chen, B., Xu, W., Liu, C., Li, X., Zhao, Z., Zhao, L., 2022. Motion Similarity Evaluation between Human and a Tri-Co Robot during Real-Time Imitation with a Trajectory Dynamic Time Warping Model. Sensors 22, 1968. https://doi.org/10.3390/s22051968

Guo, S., Fu, Y., 2020. A Method of Phase Correction in Multichannel Channelized Receiver Based on Cubic Spline Interpolation Algorithm, in: 2020 7th International Conference on Information Science and Control Engineering (ICISCE). Presented at the 2020 7th International Conference on Information Science and Control Engineering (ICISCE), pp. 40–44. https://doi.org/10.1109/ICISCE50968.2020.00019

Guo, S., Lu, J., Qin, Y., 2024. Analysis of the coupled spatial and temporal development characteristics of global liner shipping connectivity driven by trade. Ocean & Coastal Management 251, 107071. https://doi.org/10.1016/j.ocecoaman.2024.107071

Hu, Y., Tang, H., Pan, G., 2023. Spiking Deep Residual Networks. IEEE Transactions on Neural Networks and Learning Systems 34, 5200–5205. https://doi.org/10.1109/TNNLS.2021.3119238

Huang, D., Shen, L., Yu, Z., Zheng, Z., Huang, M., Ma, Q., 2022. Efficient time series anomaly detection by multiresolution self-supervised discriminative network. Neurocomputing 491, 261–272. https://doi.org/10.1016/j.neucom.2022.03.048

Huang, M., Zhu, M., Xiao, Y., Liu, Y., 2021. Bayonet-corpus: a trajectory prediction method based on bayonet context and bidirectional GRU. Digital Communications and Networks 7, 72–81. https://doi.org/10.1016/j.dcan.2020.03.002

Ikotun, A.M., Ezugwu, A.E., Abualigah, L., Abuhaija, B., Heming, J., 2023. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences 622, 178–210. https://doi.org/10.1016/j.ins.2022.11.139

Jia, C., Ma, J., 2023. Conditional temporal GAN for intent-aware vessel trajectory prediction in the precautionary area. Engineering Applications of Artificial Intelligence 126, 106776. https://doi.org/10.1016/j.engappai.2023.106776

Jiao, H., Gong, J., Li, H., Lam, J.S.L., Shu, Y., Wang, J., Yang, Z., 2026. LLM4STP: A large language model-driven multi-feature fusion method for ship trajectory prediction. Transportation Research Part E: Logistics and Transportation Review 207, 104599. https://doi.org/10.1016/j.tre.2025.104599

Jiao, W., Ma, C., 2021. Track prediction algorithm based on GA-BPNN, in: 2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT). Presented at the 2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT), pp. 211–217. https://doi.org/10.1109/ICCASIT53235.2021.9633519

Kalia, P., Paul, J., 2021. E-service quality and e-retailers: Attribute-based multi-dimensional scaling. Computers in Human Behavior 115, 106608. https://doi.org/10.1016/j.chb.2020.106608

Latifi-Pakdehi, A., Daneshpour, N., 2021. DBHC: A DBSCAN-based hierarchical clustering algorithm. Data & Knowledge Engineering 135, 101922. https://doi.org/10.1016/j.datak.2021.101922

Li, H., Jiao, H., Yang, Z., 2023a. Ship trajectory prediction based on machine learning and deep learning: A systematic review and methods analysis. Engineering Applications of Artificial Intelligence 126, 107062. https://doi.org/10.1016/j.engappai.2023.107062

Li, H., Jiao, H., Yang, Z., 2023b. 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

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

Li, H., Zhang, Y., Li, Y., Lam, J.S.L., Matthews, C., Yang, Z., 2025. Deep multi-view information-powered vessel traffic flow prediction for intelligent transportation management. Transportation Research Part E: Logistics and Transportation Review 197, 104072. https://doi.org/10.1016/j.tre.2025.104072

Li, X., Wang, K., Tang, M., Qin, J., Wu, P., Yang, T., Zhang, H., 2022. Marine Drifting Trajectory Prediction Based on LSTM-DNN Algorithm. Wireless Communications and Mobile Computing 2022, 7099494. https://doi.org/10.1155/2022/7099494

Li, Y., Chen, B.Y., Liu, Q., Zhang, Y., 2024. Geohash coding-powered deep learning network for vessel trajectory prediction using clustered AIS data in maritime Internet of Things industries. Computers and Electrical Engineering 120, 109611. https://doi.org/10.1016/j.compeleceng.2024.109611

Liang, M., Liu, K., Gao, R., Li, Y., 2025. Integrating GPU-Accelerated for Fast Large-Scale Vessel Trajectories Visualization in Maritime IoT Systems. IEEE Transactions on Intelligent Transportation Systems 26, 4048–4065. https://doi.org/10.1109/TITS.2024.3521050

Liu, B., Gao, D., Sun, Y., 2022. A Speed Prediction Method of Electric Propulsion Ship Based on GA-Elman Neural Network, in: 2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI). Presented at the 2022 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI), pp. 35–40. https://doi.org/10.1109/ISSI55442.2022.9963147

Liu, H., Wu, C., Li, B., Zong, Z., Shu, Y., 2025. Research on Ship Anomaly Detection Algorithm Based on Transformer-GSA Encoder. IEEE Transactions on Intelligent Transportation Systems 26, 8752–8763. https://doi.org/10.1109/TITS.2025.3536483

Lu, B., Lin, R., Zou, H., 2021. A Novel CNN-LSTM Method for Ship Trajectory Prediction, in:

2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys). Presented at the 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), pp. 2431–2436. https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00366

Ma, F., Wang, C., Huang, J., Zhong, Q., Zhang, T., 2024. Key grids based batch-incremental CLIQUE clustering algorithm considering cluster structure changes. Information Sciences 660, 120109. https://doi.org/10.1016/j.ins.2024.120109

Ma, S., Liu, S., Meng, X., 2020. Optimized BP neural network algorithm for predicting ship trajectory, in: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). Presented at the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 525–532. https://doi.org/10.1109/ITNEC48623.2020.9085154

Mienye, I.D., Swart, T.G., Obaido, G., 2024. Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. Information 15, 517. https://doi.org/10.3390/info15090517

Milanes, C.B., Pérez Montero, O., Cabrera, J.A., Cuker, B., 2021. Recommendations for coastal planning and beach management in Caribbean insular states during and after the COVID-19 pandemic. Ocean & Coastal Management 208, 105575. https://doi.org/10.1016/j.ocecoaman.2021.105575

Mocanu, D.C., Bou Ammar, H., Puig, L., Eaton, E., Liotta, A., 2017. Estimating 3D trajectories from 2D projections via disjunctive factored four-way conditional restricted Boltzmann machines. Pattern Recognition 69, 325–335. https://doi.org/10.1016/j.patcog.2017.04.017

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, 1037. https://doi.org/10.3390/jmse9091037

Ran, X., Xi, Y., Lu, Y., Wang, X., Lu, Z., 2023. Comprehensive survey on hierarchical clustering algorithms and the recent developments. Artif Intell Rev 56, 8219–8264. https://doi.org/10.1007/s10462-022-10366-3

Reza, S., Ferreira, M.C., Machado, J.J.M., Tavares, J.M.R.S., 2023. A customized residual neural network and bi-directional gated recurrent unit-based automatic speech recognition model. Expert Systems with Applications 215, 119293. https://doi.org/10.1016/j.eswa.2022.119293

Rutkowski, L., 2004. Generalized regression neural networks in time-varying environment. IEEE Transactions on Neural Networks 15, 576–596. https://doi.org/10.1109/TNN.2004.826127

Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.-R., 2021. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. Proceedings of the IEEE 109, 247–278. https://doi.org/10.1109/JPROC.2021.3060483

Sheng, K., Liu, Z., Zhou, D., He, A., Feng, C., 2018. Research on Ship Classification Based on Trajectory Features. The Journal of Navigation 71, 100–116. https://doi.org/10.1017/S0373463317000546

Shi, W., Huang, J., Liu, Y., Jing, S., Zhou, H., Li, W., Wang, Z., Zhang, Z., 2024. Prediction of Spatiotemporal Pollution of Soil Heavy Metal in Mining Areas Based on Grey Neural Network Algorithm. Water Air Soil Pollut 235, 780. https://doi.org/10.1007/s11270-024-07587-3

Shin, G., Yang, H., 2025. Vessel trajectory prediction in harbors: A deep learning approach with maritime-based data preprocessing and Berthing Side Integration. Ocean Engineering 316, 119908. https://doi.org/10.1016/j.oceaneng.2024.119908

Shu, Y., Cui, H., Song, L., Gan, L., Xu, S., Wu, J., Zheng, C., 2024. Influence of sea ice on ship routes and speed along the Arctic Northeast Passage. Ocean & Coastal Management 256, 107320. https://doi.org/10.1016/j.ocecoaman.2024.107320

Shu, Y., Dong, A., Liu, C., Gan, L., Song, L., 2026. Anomaly detection of ship behavior based on deep neural networks. Reliability Engineering & System Safety 266, 111801. https://doi.org/10.1016/j.ress.2025.111801

Sohn, I., 2021. Deep belief network based intrusion detection techniques: A survey. Expert Systems with Applications 167, 114170. https://doi.org/10.1016/j.eswa.2020.114170

Suo, Y., Chen, W., Claramunt, C., Yang, S., 2020. A Ship Trajectory Prediction Framework Based on a Recurrent Neural Network. Sensors 20, 5133. https://doi.org/10.3390/s20185133

Syed, M.A.B., Ahmed, I., 2023. A CNN-LSTM Architecture for Marine Vessel Track Association Using Automatic Identification System (AIS) Data. Sensors 23, 6400. https://doi.org/10.3390/s23146400

Wang, P., Pan, M., Liu, Z., Li, S., Chen, Y., Wei, Y., 2024. Ship Trajectory Prediction in Complex Waterways Based on Transformer and Social Variational Autoencoder (SocialVAE). Journal of Marine Science and Engineering 12, 2233. https://doi.org/10.3390/jmse12122233

Wang, W., Yi, Z., Zhao, L., Jia, P., Kuang, H., 2025. Application of switching-input LSTM network for vessel trajectory prediction. Appl Intell 55, 289. https://doi.org/10.1007/s10489-024-06079-5

Wang, Y., Yao, H., Zhao, S., 2016. Auto-encoder based dimensionality reduction. Neurocomputing, RoLoD: Robust Local Descriptors for Computer Vision 2014 184, 232–242. https://doi.org/10.1016/j.neucom.2015.08.104

Wang, Y., Zhu, J., Wang, X., Li, X., Zhang, W., 2023. Navigating cross-cultural port tourism: Unleashing the psychological safety valve for risk management. Ocean & Coastal Management 246, 106888. https://doi.org/10.1016/j.ocecoaman.2023.106888

Xing, B., Wu, W., Liao, F., Tu, M., Lu, M., 2023. Trajectory Prediction Model of Blended Wing Body Impact Entry Water Based on Deep Belief Network, in: Yan, L., Duan, H., Deng, Y. (Eds.), Advances in Guidance, Navigation and Control. Springer Nature, Singapore, pp. 7200–7208. https://doi.org/10.1007/978-981-19-6613-2_695

Xing, W., Wang, J., Zhou, K., Li, H., Li, Y., Yang, Z., 2023. A hierarchical methodology for vessel traffic flow prediction using Bayesian tensor decomposition and similarity grouping. Ocean Engineering 286, 115687. https://doi.org/10.1016/j.oceaneng.2023.115687

Xu, L., Wang, L., Xue, W., Zhao, S., Zhou, L., 2023. Elman neural network for predicting aero optical imaging deviation based on improved slime mould algorithm. Optoelectron. Lett. 19, 290–295. https://doi.org/10.1007/s11801-023-2137-7

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X., 2018. AttnGAN: Fine-Grained

Text to Image Generation With Attentional Generative Adversarial Networks. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1316–1324.

Xue, H., Wang, S., Xia, M., Guo, S., 2024. G-Trans: A hierarchical approach to vessel trajectory prediction with GRU-based transformer. Ocean Engineering 300, 117431. https://doi.org/10.1016/j.oceaneng.2024.117431

Yang, C.-H., Lin, G.-C., Wu, C.-H., Liu, Y.-H., Wang, Y.-C., Chen, K.-C., 2022a. Deep Learning for Vessel Trajectory Prediction Using Clustered AIS Data. Mathematics 10, 2936. https://doi.org/10.3390/math10162936

Yang, C.-H., Wu, C.-H., Shao, J.-C., Wang, Y.-C., Hsieh, C.-M., 2022b. AIS-Based Intelligent Vessel Trajectory Prediction Using Bi-LSTM. IEEE Access 10, 24302–24315. https://doi.org/10.1109/ACCESS.2022.3154812

Yang, G., Wang, G., Li, Y., Yu, W., 2024. ACBL:Attentive CNN-BiLSTM Model For Trajectory Prediction, in: 2024 43rd Chinese Control Conference (CCC). Presented at the 2024 43rd Chinese Control Conference (CCC), pp. 8243–8248. https://doi.org/10.23919/CCC63176.2024.10662836

Yang, K., Yang, T., Yao, Y., Fan, S., 2021. A transfer learning-based convolutional neural network and its novel application in ship spare-parts classification. Ocean & Coastal Management 215, 105971. https://doi.org/10.1016/j.ocecoaman.2021.105971

Yi, Q., Zuo, Y., Li, T., Mao, Y., Xiao, Y., 2021. Forecasting of Vessel Traffic Flow Using BPNN Based on Genetic Algorithm Optimization, in: 2021 International Wireless Communications and Mobile Computing (IWCMC). Presented at the 2021 International Wireless Communications and Mobile Computing (IWCMC), pp. 1059–1063. https://doi.org/10.1109/IWCMC51323.2021.9498607

Yin, X., Zhang, Z., Huang, Y., Hu, H., 2025. Dynamic Diffusion Graph Convolutional Network with Scene-Guided Gating for Trajectory Prediction in IoT-Based Intelligent Transportation Systems. IEEE Internet of Things Journal 1–1. https://doi.org/10.1109/JIOT.2025.3647420

Zhang, H., Dana, K., 2018. Multi-style Generative Network for Real-time Transfer. Presented at the Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pp. 0–0.

Zhang, N., Ding, S., Zhang, J., Xue, Y., 2018. An overview on Restricted Boltzmann Machines. Neurocomputing 275, 1186–1199. https://doi.org/10.1016/j.neucom.2017.09.065

Zhang, W., Chen, W., Wu, X., Chen, J., Zhang, Z., 2025. Evaluation and spatiotemporal analysis of low-carbon efficiency of the “ship-port” system. Maritime Policy & Management 0, 1–33. https://doi.org/10.1080/03088839.2025.2514765

Zhang, Y., Ma, H., Xu, J., 2021. Neural network-based fuzzy vibration controller for offshore platform with random time delay. Ocean Engineering 225, 108733. https://doi.org/10.1016/j.oceaneng.2021.108733

Zhang, Y., Tu, P., Zhao, Z., Chen, X.-Y., 2025. Incorporating prior knowledge of collision risk into deep learning networks for ship trajectory prediction in the maritime Internet of Things industry. Engineering Applications of Artificial Intelligence 146, 110311. https://doi.org/10.1016/j.engappai.2025.110311