ADSRP: A DBSCAN-SNN Framework for AIS-Based Ship Route Planning Using Spatiotemporal Feature Fusion

Yan Deng[1], Yunye Ren[2,*]
[1]Yangtze River Administration of Navigational Affairs Archives Center, Wuhan 430014, China
[2]School of Nautical Technology, Jiangsu Shipping College, Nantong 226010, China
E-mail: dy027@163.com, R2058662858@126.com
*Corresponding author

Keywords: Route planning, AIS, DBSCAN, SNN, ship

Received: April 22, 2025

Addressing the issues of noise interference, inadequate modeling of nonlinear characteristics, and computational inefficiency in ship trajectory planning, this study introduces a multi-stage joint optimization model. The model is built upon Automatic Identification System (AIS) data cleaning, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and a Siamese Neural Network (SNN). The operation of the AIS Data and DBSCAN-Based Ship Route Planning Model (ADSRP) involves the following steps: First, AIS data is cleansed by employing a dynamic neighborhood radius and linear interpolation with a sliding window. Key steering points are then extracted by integrating the Douglas-Peucker (DP) algorithm, resulting in an 85.4% reduction in trajectory redundancy. Subsequently, DBSCAN is utilized for density-based clustering of trajectory endpoints, achieving a 93.6% filtering accuracy for noise points. Finally, a symmetric-weight SNN architecture (comprising a 4-layer Transformer encoder and multi-head attention) is designed to filter high-density routes based on cosine similarity.Experimental results demonstrate that, in comparison to the traditional genetic algorithm-based Whole Process Route Planning (WPRP), ADSRP enhances trajectory fitting in the simulation environment by 21% (with an average cosine similarity of 0.86 for ADSRP and 0.71 for WPRP) and shortens the planning time by 67.8% (8.11s for ADSRP and 25.24s for WPRP). In real-world port scenarios, ADSRP reduces voyage deviation by 36.8% (0.98nmi for ADSRP and 1.55nmi for WPRP), cuts fuel consumption by 20.8% (362.58L for ADSRP and 457.89L for WPRP), and optimizes memory usage to 27.5% (compared to the benchmark's 42.5%). Parameter sensitivity analysis verifies the significant impact of key parameters on clustering fragmentation and port identification accuracy (F1-score difference of 22%). The model is co-optimized by data-driven clustering and deep metric learning, providing a high-accuracy, low-energy solution for dynamic path planning in complex sea areas and supporting edge device deployment.

Povzetek: Pomorsko načrtovanje poti je narejeno s tristančnim okvirjem (čiščenje AIS + DBSCAN + SNN/Transformer). V simulacijah in praksi izboljša poti, čas, odklon, porabo goriva in potrošnjo pomnilnika.

1 Introduction

Ship route planning, as a core component of the maritime transportation system, directly affects shipping efficiency and safety. Ship route planning methods are divided into mathematical model-based planning methods and data-driven planning methods. The former solves the shortest path or minimum cost in a static environment through graph theory, dynamic programming, and other methods [1]. The latter processes dynamic multi-objective optimization and complex constraint problems through evolutionary algorithms, reinforcement learning, and other methods [2]. However, against the backdrop of intensified climate change and surging competition in shipping, traditional methods are unable to meet the comprehensive needs of modern shipping. For example, sequential decision-making methods face the "curse of dimensionality", and when the number of ship state variables increases, the computational complexity grows exponentially, making it difficult to meet real-time requirements [3].The linear programming method needs to simplify multi-objective conflicts into a single objective function, resulting in distorted trade-offs between economy and safety in practical scenarios [4].Genetic algorithms are prone to premature convergence problems, and the design of crossover/mutation operators relies on prior knowledge, resulting in insufficient adaptability under complex constraints [5].

In summary, traditional ship route planning methods mainly rely on crew experience for manual design, which has the problems of low efficiency and strong subjectivity. The Automatic Identification System (AIS) data can accurately record ship status parameters and support trajectory mining and pattern analysis [6]. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm can automatically identify high-density core points, boundary points, and noise points

through density accessibility [7]. However, AIS data has missing trajectories and noise, and there are more abnormal drift points and duplicate records at high sampling intervals. Moreover, the DBSCAN algorithm is sensitive to parameters and difficult to adaptively handle trajectory clusters with significant density differences. Therefore, the research focuses on the advantages and disadvantages of AIS data and DBSCAN, and makes improvements to ultimately construct an AIS Data and DBSCAN-based Ship Route Planning Model (ADSRP).To assess the viability of integrating the aforementioned techniques, the study poses the following question: Can the amalgamation of density clustering and metric learning enable achieving a trajectory deviation of less than 1 nmi while concurrently cutting computational requirements by over 60%? Subsequently, the study undertakes the task of validating this proposition. The research aims to solve the problem of delayed dynamic environmental response in traditional ship route planning by utilizing the multidimensional spatiotemporal characteristics of AIS shipping big data and the density adaptive clustering ability of DBSCAN algorithm. The innovation of the research lies in deeply coupling the DBSCAN algorithm with AIS data characteristics, constructing a dynamic parameter framework, and achieving multi-scale density clustering of ship trajectory endpoints. Meanwhile, a trajectory similarity weighting mechanism is designed to enhance the robustness of trajectory clustering through AIS data-driven DBSCAN noise filtering and cluster merging strategies.

The research is divided into four sections. The first section introduces the current research on the logic and algorithms of ship route planning worldwide, and clarifies the shortcomings of existing research. The second section starts from AIS data processing, Douglas-Peucker (DP) algorithm, etc., establishes accurate and efficient trajectory feature extraction methods, and combines DBSCAN and Siamese Neural Networks (SNN) to construct trajectory planning methods, establishing an ADSRP model. The third section provides numerical examples and practical application analysis of the proposed model to verify its reliability. The fourth section provides a comprehensive summary and analysis of the article.

2 Related works

In the process of intelligentization and greening of the global shipping industry, ship route planning faces dynamic environmental challenges. Traditional methods rely on static data, which makes it difficult to cope with real-time risks, prompting researchers to explore new methods. In response to the harsh environment and insufficient sea ice information in the Arctic ice zone, Wu et al. proposed an online interactive system based on big earth data, combined with reinforcement learning and synthetic aperture radar data, to improve route safety and accuracy through dynamic path calculation and real-time ice water classification [8].Zhao et al. improved the artificial fish swarm algorithm to address the low path efficiency, local optimization, and insufficient smoothness of autonomous surface vessels. They introduced directional operators to enhance search efficiency, designed probability weight factors to reduce the risk of local optima, and combined adaptive convergence strategies and smoothing mechanisms to enhance practicality in complex sea conditions [9]. Zhao et al. constructed a multi-objective meteorological routing framework to address issues such as high fuel consumption and high risks in ships. They used particle swarm optimization and non-dominated sorting techniques to optimize fuel consumption, risk, and route coordination models through mutation operations and elite selection [10].In response to the lack of dynamic obstacle avoidance and rule coordination in unmanned ships, Li et al. proposed a multi-layer adaptive search tree method that integrated the speed obstacle method to predict collision domains, and combined rolling window modeling and path optimization strategies to improve dynamic response capabilities [11].

In addition, in response to the problem of weak multi-objective collaboration and trajectory adaptability, Hongjie et al. designed a phased trajectory optimization framework, which usedDBSCAN clustering to extract turning points and generate a global route network, balancing safety and economy [12].In response to the lack of AIS data mining leading to planning disconnection, Gao et al. constructed a framework for ship navigable routes, using manifold distance to generate high-density search networks, and driving topology generation with real trajectories to reduce path deviation rates [13].Kim et al. proposed a nested trajectory reconstruction framework to address the low accuracy of AIS spatiotemporal bias and destination prediction. The framework extracted spatiotemporal features through multi-channel vectorization and attention mechanisms, and combined gradient dropout techniques to improve long-term prediction accuracy [14]. In response to the lack of neutron trajectories in traditional trajectory partitioning, Cui et al. developed a multi-attribute DBSCAN method that integrated parameters such as speed and heading to achieve adaptive partitioning and enhance anomaly detection capabilities in complex scenarios [15]. A comprehensive comparison of these methods is shown in Table 1.

Table 1: Recent advances in ship route planning

Methods	Model type	Datasets	Clustering algorithm	Key features	Results	References
Online Interactive Route Planning System (RouteView)	Reinforcement learning integrated with deep learning	Sea ice and meteorological data + synthetic aperture radar data	-	Reinforcement learning based dynamic planning of arctic routes +synthetic aperture radar with deep learning for real-time sea ice classification	Enhancing northeast waterway safety and improving information extraction methods.	[8]
Improved Artificial Fish Swarm Algorithm (IAFSA)	Population intelligence optimization algorithm	Simulation of navigation data in the south china sea	-	Directional operators, probabilistic weight factors, adaptive convergence and path smoothing.	Algorithm efficiency and path quality outperform existing methods, and model ship tests verify applicability.	[9]
Multi-Objective Particle Swarm Optimization Routing System (MOPSO-RS)	Multi-objective optimization algorithm	Container ship simulation data	-	Mutation manipulation and elite selection balancing convergence speed and diversity, pareto optimal solution recommendation criterion.	Realization of low-risk, low-fuel-consumption, short-time route planning.	[10]
Whole Process Route Planning (WPRP)	Trajectory clustering and network optimization	Vessel trajectory data	Hierarchical dbscan（hdbscan）	Trajectory smoothing (polynomial approximation with exponential kernel) → compression (douglas-peucker) → clustering (hdbscan) → road network construction.	The trajectory similarity is improved by 36.53%, the compression rate is >92%, the clustering profile coefficient is 0.9032, and the road network contains 299 key nodes.	[12]
High Density Searching Framework Extraction (HDS-FE)	Ais big data-driven network construction	Global ais data	-	Ship navigable route framework based on ais big data to support actual navigation distance calculation and route planning.	Solve the problem of disconnection between traditional route planning and real sailing habits, and enhance the feasibility of route application.	[13]

As can be seen from Table 1, although existing research has made progress, it mostly relies on static data and manual experience, lacks in-depth exploration of historical navigation patterns, and has low clustering efficiency and insufficient feature extraction for massive AIS data, making it difficult to meet the personalized needs of multiple ship types. Therefore, the study aims to optimize the efficiency of trajectory feature extraction and clustering through AIS data and DBSCAN, improve the robustness and dynamic adaptability of the model, and construct an ADSRP ship route planning model to provide efficient and accurate solutions for practical scenarios.

3 Methods and material

This section is divided into two parts. The first part provides a detailed explanation of AIS data categories, DP algorithms, and other methods, preprocesses AIS data, and extracts trajectory features. The second part combines DBSCAN and SNN to construct a clustering method for trajectory endpoints (starting and ending points), proposes an ADSRP model, improves the accuracy of trajectory planning and noise robustness of the model, and enhances its dynamic environmental adaptability.

3.1 AIS data preprocessing and trajectory feature extraction

Ship trajectory feature extraction is the core foundation of intelligent navigation. However, traditional feature extraction methods are difficult to effectively handle the nonlinear characteristics of ship motion, which can easily lead to the loss of key turning points and trajectory segmentation fractures. AIS data can provide high-resolution spatiotemporal sequences and multidimensional motion parameters, thus reducing trajectory distortion and key node omissions in feature extraction [16]. Therefore, the study establishes a trajectory feature extraction method based on AIS data. The study first preprocesses AIS data, as shown in Figure 1.

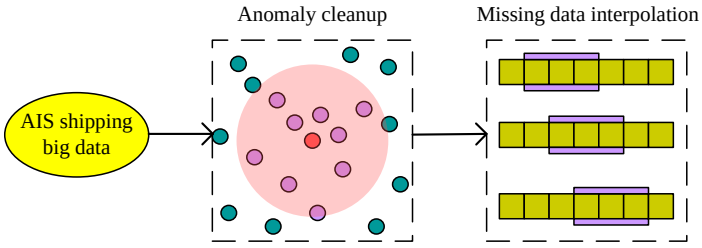


Figure 1: AIS data preprocessing flow

As shown in Figure 1, data preprocessing is divided into two stages: outlier cleaning and missing data interpolation. Among them, the outlier cleaning adopts the dynamic farthest point cleaning method in the neighborhood, which dynamically adjusts the neighborhood radius based on the real-time speed of the ship, and detects and removes outlier trajectory points through deviation threshold [17]. The study first performs dynamic radius calculation, based on the real-time ship speed v_t and time interval Δt, defines the neighborhood radius as shown in equation (1).

$$r_t = max(r_{min}, v_t * \Delta t * \alpha) \tag{1}$$

In equation (1), r_{min} is the minimum radius (set to 100m). α is the safety factor (set to 1.2), based on historical AIS data statistics, balancing noise filtering with trajectory fidelity. Afterwards, the spatial deviation between the current point p_t and its neighboring points $\{p_{t-1}, p_{t+1}\}$ is calculated, as shown in equation (2).

$$d_{dev} = max\left(\frac{\|p_t - p_{t-1}\|}{r_{t-1}}, \frac{\|p_t - p_{t+1}\|}{r_{t+1}}\right) \tag{2}$$

In equation (2), d_{dev} represents the spatial deviation of the current point p_t. When $d_{dev} > 2.0$, the point is marked as an outlier and cleared from the planning data. The dynamic neighborhood farthest point cleaning method can adapt to different navigation states, effectively balancing trajectory smoothness and keypoint preservation. In the stage of missing data interpolation, the sliding window linear interpolation method is studied, which is based on the spatiotemporal continuity of adjacent points within the time window, and calculates the position and motion parameters of missing points through linear weighting [18]. It first performs a window partitioning operation, with the missing points as the center, expanding the time window Δt (Δt is set to 10 minutes, the linear model needs only 2 neighbors in this window, reducing the interpolation time consuming) before and after.Moreover, the effective point set $\{p_{t-k},..., p_{t-1}, p_{t+1},..., p_{t+k}\}$ is extracted within the window. Afterwards, the interpolation weight w is calculated based on the time difference between the missing point t and the adjacent point, as shown in equation (3).

$$w = \frac{t - t_{t-1}}{t_{t+1} - t_{t-1}} \quad w \in [0,1] \tag{3}$$

In equation (3), t_{t-1} and t_{t+1} are the most recent valid timestamps before and after the missing point. Finally, linear interpolation is performed based on the effective point set $\{p_{t-k},..., p_{t-1}, p_{t+1},..., p_{t+k}\}$ and the interpolation weight w, as shown in equation (4).

$$p_t^{fill} = (1-w) * p_{t-1} + w * p_{t+1} \tag{4}$$

In equation (4), p_t^{fill} is the interpolation point, which refers to the latitude, longitude, speed, and heading data of a certain point. Dynamic neighborhood radius is adjusted by speed-driven adjustment (Eqs. 1-2) to ensure that the detection range is enlarged to avoid missing detection when sailing at high speeds, and the radius is contracted to suppress overfitting when sailing at low speeds. Sliding window interpolation (Eq. 3-4) fills in the missing based on spatio-temporal continuity and reduces the computational overhead compared with traditional cubic spline interpolation. After completing data preprocessing, the DP algorithmis used to extract key node features such as ship trajectory turning points and anchorage areas. Through high fidelity simplification, the key geometric features of the trajectoryare preserved, significantly reducing data redundancy, as shown in Figure 2.

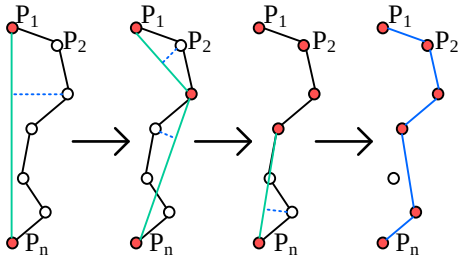


Figure 2: Feature extraction process of DP algorithm

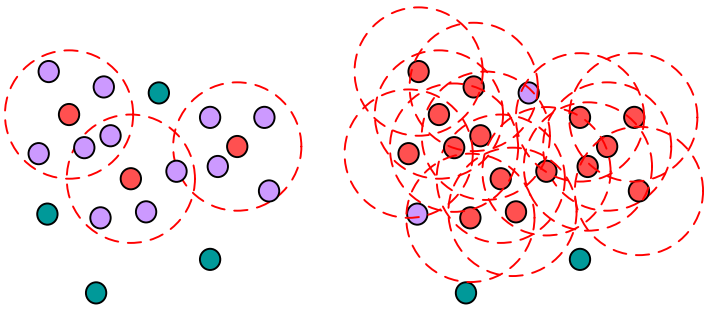


Figure 3: Clustering principle of DBSCAN

As shown in Figure 2, the black line is the original trajectory and the last blue line is the optimized and simplified trajectory. The red dots represent key feature nodes, while the white dots represent uncalculated or non key feature points. The study first inputs the preprocessed trajectory point sequence $P=\{p_1, p_2, ..., p_n\}$ and sets a simplified threshold ε (Optimized by grid search to ensure simplification error <5%). Afterwards, the algorithm connects the starting point p_1 and the ending point p_n to generate the head tail baseline, and calculates the vertical distance d_i from the midpoint p_i to the baseline. The algorithm selects the maximum distance point p_{max}, if $d_{max} > \varepsilon$, uses p_{max} as the feature point, divides the trajectory into $[p_1, p_{max}]$ and $[p_{max}, p_n]$, and recursively processes the sub segments [19]. It retains all feature points that satisfy $d_i \geq \varepsilon$ and generates a simplified trajectory P'. The DP algorithm simplifies the trajectory accuracy by controlling the geometric deviation threshold, while retaining key features such as turning points and anchoring areas, and reducing computational complexity.

3.2 DBSCAN-based trajectory endpoint clustering and establishment of adsrp model

The data preprocessing and trajectory feature extraction methods proposed in the study can improve the quality of trajectory data and accurately capture ship maneuvering behavior characteristics. However, traditional trajectory planning methods lack robustness to the heterogeneity of trajectory endpoint spatial distribution and multi-scale density changes, which can easily lead to misjudgment of trajectory boundaries and failure of anchorage identification. DBSCAN can adaptively identify multi-scale spatial clusters based on density accessibility, effectively distinguishing noise points from low-density anchorage areas. Therefore, the study uses DBSCAN to cluster and determine trajectory endpoints (starting and ending points), as shown in Figure 3.

As shown in Figure 3, DBSCAN conducts clustering based on density reachability. It begins by calculating the number of samples within the ε-neighborhood of each data point. If a point has at least MinPts samples in its neighborhood, it is designated as a core point. The algorithm then expands the neighborhood around this core point to group all density-connected points into the same cluster. Points that do not meet the criteria for core points or density-connected points are classified as noisy [20]. DBSCAN clustering of trajectory endpoints (origin/destination) corresponds to actual port coordinates. Port recognition accuracy measures the algorithm's capability to capture high-frequency berthing areas. High-precision clustering ensures that the origin/destination points of planned routes align with real port locations, providing a geographic reference for global route generation.The clustering process is as follows: First, the network inputs dataset $P=\{p_1, p_2, ..., p_n\}$, sets the neighborhood radius ò and the minimum number of neighborhood points $MinPts$ (set to 5 in the study).The process dynamically adjusts parameter ò based on real-time sea condition data (ò = 500 × [1 + 0.2 × (wind speed / 10)] meters). For each point p_i, it calculates the number of points in its ò neighborhood, as shown in equation (5).

$$N_{ò}(p_i)=\{p_j \in D \mid dist(p_i, p_j) \leq ò\} \quad (5)$$

In equation (5), if $|N_{ò}(p_i)| \geq MinPts$, p_i is marked as the core point. Afterwards, the neighborhood is expanded, starting from the unvisited core point p_c, the queue $Q=\{p_c\}$ is initialized, and p_c is assigned to cluster C_k. Each point p_q in Q is traversed, and if p_q is the core point, the unvisited points in its ò neighborhood are added to Q.The traversal update process is shown in equation (6).

$$Q \leftarrow Q \cup (N_{ò}(p_q) \setminus \{p_v\}) \quad (6)$$

In equation (6), p_q is a point in Q; $N_{ò}$ denotes a point in the ò neighborhood; p_v represents the visited point. p_q is marked as visited, and if p_q does not belong to any cluster, it is added to C_k. Other points that cannot be reached from any core point density are marked as noise [21]. Noise points are directly excluded after clustering and are not involved in trajectory similarity screening and route generation to avoid low-density abnormal trajectories interfering with the planning results, and their spatial distribution is used for subsequent route optimization in sparse regions. In addition, based on the endpoint clustering results, it is necessary to screen trajectory patterns with high spatiotemporal consistency to extract key routes and support the generation of

recommended routes. As a deep metric learning method in machine learning, SNN effectively measures the spatiotemporal similarity of variable length trajectories through shared weights and feature encoding, with strong noise resistance and adaptability to complex patterns [22]. Therefore, the study utilizes SNN to efficiently measure the spatiotemporal similarity of trajectories between DBSCAN clustering endpoints, accurately screen key routes to support recommendation generation, and its structure is shown in Figure 4.

As shown in Figure 4, SNN consists of a deep encoder (Transformer) with shared weights between two towers, a fully connected layer, and a similarity calculation module, which symmetrically processes input trajectory pairs. The encoder adopts a 4-layer Transformer with 512 hidden units, 8 multi-head attention heads, a dropout rate of 0.1, and GELU activation function. The 4-layer design was chosen to balance model depth and computational efficiency; the 512-dimensional hidden units match trajectory feature dimensions; the 8 attention heads capture multi-scale spatiotemporal relationships; and the 0.1 dropout prevents overfitting.The operation process of SNNis as follows: First, inputs the trajectory pair (T_A, T_B), where $T_A = \{x_1^A, ..., x_m^A\}$ and $T_B = \{x_1^B, ..., x_n^B\}$ .Each point $x_t \in R^d$ incorporates dynamic feature channels including position, speed, heading, meteorological, and ocean current parameters, enabling similarity computation to integrate real-time environmental constraints and achieve path generation combining historical patterns with dynamic feedback. Afterwards, feature encoding is performed, which is divided into positional encoding and self attention encoding. In the position encoding stage, SNN adds position encoding $P_t \in R^d$ to the trajectory points to generate an input vector, as shown in equation (7).

$$e_t = x_t + P_t \tag{7}$$

In equation (7), e_t is the encoded input vector. In the self attention encoding stage, SNN extracts global features through multi-head self attention layers, as shown in equation (8)

$$\begin{cases} H_A = \mathrm{TE}(E_A) \ \ H_A \in R^{m \times d} \\ H_B = \mathrm{TE}(E_B) \ \ H_B \in R^{n \times d} \end{cases} \tag{8}$$

In equation (8), E_A and E_B are input sequence matrices, and $E_A = [e_1^A, ..., e_m^A]$, $E_B = [e_1^B, ..., e_n^B]$. H_A and H_B are the encoded trajectory feature matrices. TE is the encoding operation. Afterwards, the encoded features are subjected to temporal average pooling to generate trajectory level vectors, as shown in equation (9).

$$\begin{cases} h_A = \dfrac{1}{m} \sum_{t=1}^{m} H_A^{(t)} \\ h_B = \dfrac{1}{n} \sum_{t=1}^{n} H_B^{(t)} \end{cases} \tag{9}$$

In equation (9), h_A and h_B are trajectory level feature vectors. Next, SNN calculates the cosine similarity between the two, as shown in equation (10).

$$s = \frac{h_A * h_B}{\|h_A\|_2 * \|h_B\|_2} \tag{10}$$

In equation (10), s is the cosine similarity score of the trajectory pair, which measures the similarity between two trajectories. The cosine similarity score $s \in [-1,1]$, where a larger s indicates a more similar trajectory, was set to a threshold of 0.85 in the study [23]. During this process, SNN uses triplet loss to train the model, as shown in equation (11).

$$\mathrm{L} = \max(0, s_{\mathrm{neg}} - s_{\mathrm{pos}} + \mathrm{margin}) \tag{11}$$

In equation (11), s_{neg} represents the similarity between positive sample pairs (similar trajectories). s_{pos} is the similarity between negative sample pairs (heterogeneous trajectories). margin is the similarity difference threshold (set to 0.5 in the study). In summary, the proposed ADSRP model architecture is shown in Figure 5.

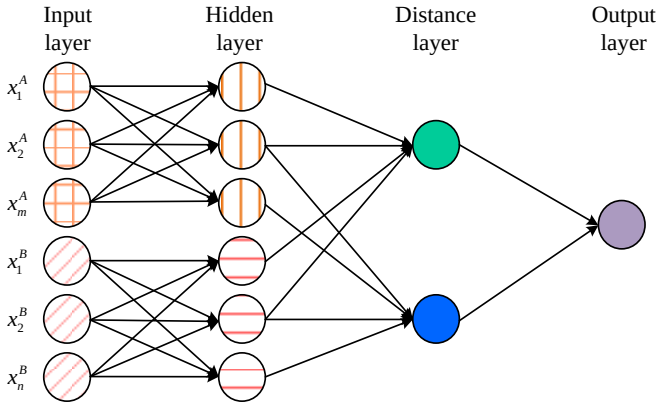


Figure 4: Network structure of SNN

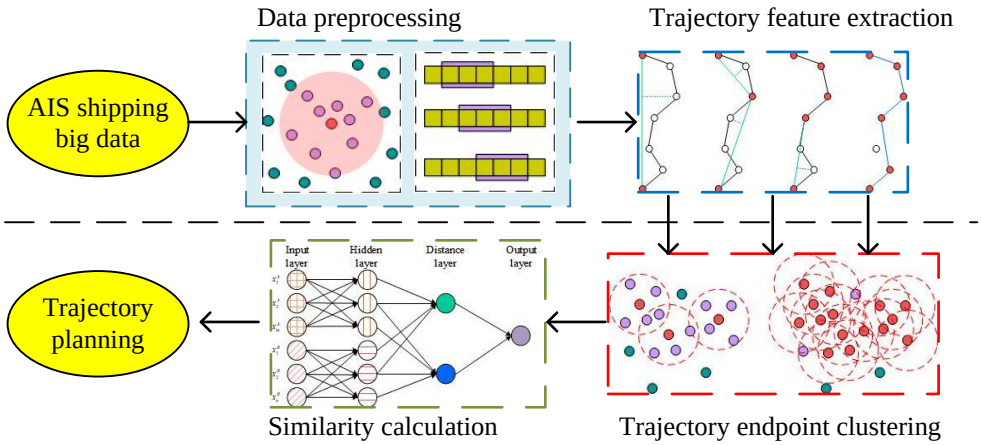


Figure 5: ADSRP model architecture

Table 2: Development environment and experimental parameters

Configurations and parameters		Details
Configurations	Hardware	NVIDIA RTX 6000 Ada 48GB
		AMD Threadripper 7960X
	Software	QGIS 3.28
		PyTorch 2.0
		scikit-learn 1.2
Parameters	Missing hours	15min
	Neighborhood radius(NR)	500m (port area)
	Encoder layers	4
	Learning rate	0.001
	Training round	50
	Confidence interval	95%

As shown in Figure 5, the ADSRP model first uses the DP algorithm to geometrically simplify the preprocessed AIS data trajectory, extract key node features such as turning points and anchoring areas, and solve the problem of nonlinear motion modeling.Afterwards, the model combines DBSCAN to perform density clustering on the trajectory endpoints, adaptively distinguishing noise points from multi-scale anchor areas.Finally, a symmetric deep network based on SNN is designed to screen key flight path patterns with high spatiotemporal consistency through trajectory feature encoding and cosine similarity measurement.By integrating a three-stage process — comprising fidelity compression via the DP algorithm, noise-resistant clustering using DBSCAN, and depth measurement through an SNN—this model attains both high accuracy and robust adaptability in route planning within complex and dynamic environments.

After filtering trajectories based on high spatiotemporal consistency, the model facilitates trajectory recombination by leveraging real-time environmental data (e.g., meteorological conditions and obstacle locations). It dynamically adjusts segment connection priorities using similarity weights derived from an SNN, integrates the A* algorithm to identify an optimal path that balances safety and efficiency across clustered regions, and employs receding horizon control to iteratively refine the path in dynamic environments.

4 Results

The study aimed to verify the superiority of the ADSRP ship route planning model through dual dimensional experiments of simulation and real-world scenarios.Simulation experiments verify the basic effectiveness and theoretical advantages of the algorithm in a controllable environment, while actual experiments evaluate dynamic adaptability based on complex channel data.The former focuses on the robustness verification of the core mechanism of the model, while the latter tests the practical application potential under multi-source interference.

4.1 Simulation operation experiment

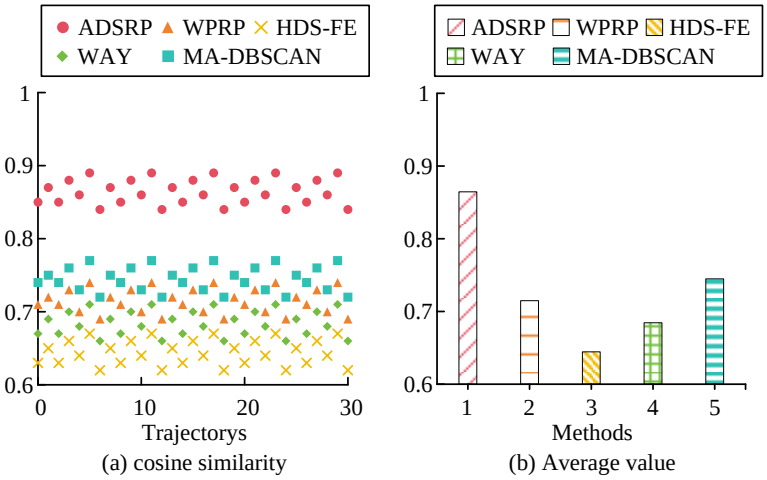
In the simulation operation experiment, the application environment of the ship route planning model was studied and adapted to the development environment and experimental parameters, as shown in Table2.

According to Table 2, the simulation application environment for route planning using the QGIS 3.28 platform was studied. SNN module was trained using PyTorch 2.0. DBSCAN clusteringwas implemented through scikit-learn 1.2. Other model parameters were subject to the settings in the research method. Moreover, the study used the Marine Cadastre AIS dataset as the training and testing sets (with a ratio of 8:2).This dataset contains AIS data for the entire year of 2021 in US waters, covering 1.2 million ship trajectories and a port area

trajectory density of 50 trajectories per square kilometer[24].The training and test sets were divided using stratified sampling based on vessel types (e.g., cargo ships/passenger vessels) and route areas (coastal/open-ocean) to ensure consistent class distribution between sets and prevent evaluation bias from regional or vessel-type differences. For vessel types and route frequencies with low occurrence (e.g., research vessels), temporal oversampling was applied, and a class-weighted loss function was introduced to reduce the impact of long-tail distribution on similarity measurement. All between-group comparisons (e.g., range deviation, fuel consumption) were performed using a two-sample t-test (normal distribution) with a significance level of $\alpha = 0.05$; confidence intervals (95% CI) were calculated by Bootstrap method.

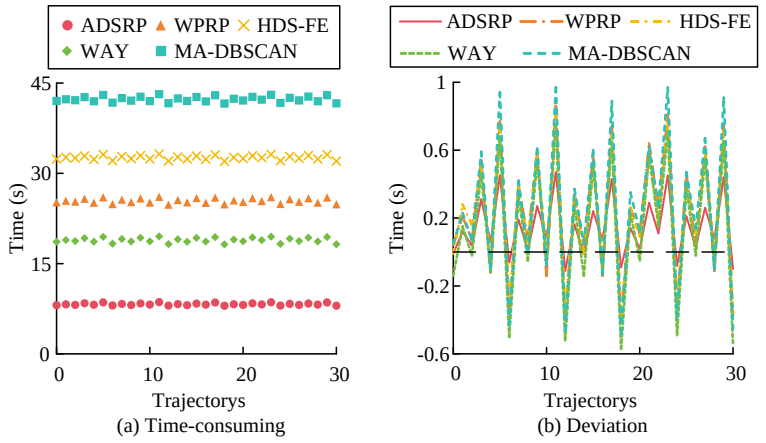
In addition, the study selected the methods used in references [12]-[15] as comparative models, including Whole Process Route Planning (WPRP), High Density Searching Framework Extraction (HDS-FE), Waypoint Analysis-based Destination Estimation (WAY), and Multi-Attribute DBSCAN Optimization (MA-DBSCAN). The ADSRP model proposed in the study was taken as the research object. The study first evaluated the feasibility of model trajectory planning by comparing the cosine similarity between trajectories planned by different methods and historical high-frequency trajectories, as shown in Figure 6.

From Figure 6 (a) and Figure 6 (b), the ADSRP model, based on DBSCAN and spatiotemporal feature fusion mechanism,achieved an average cosine similarityof 0.86in 30 experiments,significantly higher than comparisonmodels such as WPRP (0.71) and HDS-FE (0.64) ($p<0.001$), with a 95% confidence interval of [0.83, 0.89], covering historical high-frequency trajectory patterns.However, WPRP lost temporal continuity due to segmented clustering (range 0.13), while HDS-FE relied on kernel density estimation, resulting in spatial over smoothing. The cosine similarity deviation standard deviation of ADSRP was 0.05 (95% CI±0.10), with a significantly smaller fluctuation range than WAY (0.12) and MA-DBSCAN (0.07), attributed to its encoder's self attention weight dynamic allocation mechanism that suppresses AIS noise interference. There was a significant difference ($p<0.01$) between the ADSRP deviation distribution and the comparison model, with a confidence interval half width (0.025) of only 31.3% of WPRP. Subsequently, the study evaluated the computational efficiency of the model by comparing the time taken from input data to output planned routes using different methods, as shown in Figure 7.



(a) cosine similarity (b) Average value

Figure 6: Differences in trajectory cosine similarity



(a) Time-consuming (b) Deviation

Figure 7: Difference in time spent on planning routes

As shown in Figure 7 (a), the ADSRP model relied on the parallel multi-head self attention mechanism of SNN and took an average of 8.11 seconds (95% CI [7.89, 8.33]) in 30 experiments, which as 67.8% -80.7% lower than WPRP (25.24 seconds) and MA-DBSCAN (42.08 seconds)($p<0.001$).However, the complexity of MA-DBSCAN increased significantly due to the calculation of multi-attribute distance matrix (speed/heading/position), and HDS-FE relied on iterative kernel density search to further drag down efficiency.According to Figure 7 (b), the standard deviation of ADSRP time was 0.18s (95% CI ± 0.35), and the volatility was only 52.9% of the WAY model and 40% of the MA-DBSCAN.Moreover,there was a significant difference ($P<0.01$) between the deviation distribution of ADSRP and the comparison model, with a confidence interval half width of 0.22s, which was 57.7% narrower than WPRP (0.52s). This proved the stability of the SNN architecture in processing long sequence trajectories (>1000 points) and met the real-time threshold of port scheduling (<10s). Furthermore, to verify the influence of DBSCAN algorithm parameters (neighborhood radius ϵ, minimum point number MinPts) on model performance, ADSRP model parameter sensitivity experiments were conducted, as shown in Table 3.

According to Table 3, when ϵ=300m and MinPts=3, the excessively small neighborhood radius led to the fragmentation of the port trajectory into 34 clusters, with noise points accounting for 12.5% and a port recognition accuracy of only 72.3% (95% CI [68.1,76.5]). However, the combination of ϵ=700m and MinPts=7 caused neighboring ports to merge incorrectly due to the large neighborhood, resulting in an accuracy rate of 84.3% (CI [79.6, 89.0]).The optimal parameter combination of ϵ=500m and MinPts=5 achieved a balance between the number of clusters (15), the proportion of noise points (3.8%), and accuracy (93.6%, CI [90.5, 96.7]), and its performancewas significantly better than other combinations ($p<0.01$).Analysis showed that MinPts determined the strength of noise filtering by controlling the spatial density perception granularity: When ϵ<400m, the trajectory segmentation was too fine, and when MinPts<5, temporary anchor interference was introduced. Both would destroy the topological consistency of the origin and destination clustering, thereby reducing the reliability of route planning. Moreover, to investigate the effectiveness of the methodology module introduced in the study, the study conducted ablation experiments, the results of which are shown in Table 4.

In Table 4, removing the dynamic neighborhood cleaning module resulted in a decrease in model noise suppression (cosine similarity 0.71 ± 0.07, 17.4% lower than the full model, p=0.003) and an increase in range deviation to 1.82 ± 0.33 nautical miles. Removing the sliding window interpolation resulted in a distortion of the data's localized trend (deviation 1.95 ± 0.37 nautical miles, p=0.008). Without DP trajectory simplification, computational elapsed time increased to 9.8 ± 0.9 seconds (+20.9%) and voyage bias rose to 1.47 ± 0.29 nautical miles (p=0.012). Replacing DBSCAN clustering with K-means resulted in a significant reduction in port identification accuracy (deviation 1.24 ± 0.25 nautical miles, p=0.021). Fuel consumption increased to 467.5 ± 37.9L (+28.9%, p=0.004) when the SNN similarity metric was disabled. The full ADSRP model incorporated all modules to validate the need for multi-stage co-optimization with a cosine similarity of 0.86 ± 0.05, a range deviation of 0.98 ± 0.12 nautical miles and a low fuel consumption of 362.6L. For the feasibility verification of the cosine similarity threshold s in SNN, the study further conducted sensitivity experiments as shown in Table 5.

Table 3: The sensitivity of DBSCAN parameters

Parameter combinations			Number of clusters	Percentage of noise points (%)	Port recognition accuracy (%)	Accuracy (95% CI)
ϵ (m)	Minpts	Average number of ports				
300	3	36.1	34	12.5	72.3	[68.1, 76.5]
	5	29.7	28	8.9	78.5	[74.0, 83.0]
	7	22.2	21	6.3	82.1	[77.5, 86.7]
500	3	23.1	22	8.1	85.4	[81.2, 89.6]
	5	15.5	15	3.8	93.6	[90.5, 96.7]
	7	12.6	12	4.9	89.7	[85.4, 94.0]
700	3	16.9	16	10.7	80.2	[75.8, 84.6]
	5	9.5	9	6.5	87.9	[83.3, 92.5]
	7	7.4	7	5.1	84.3	[79.6, 89.0]

Table 4: Ablation experiments

Ablated Modules/Full model	Average cosine similarity	Range deviation (nmi)	Fuel consumption (L)	Planning time (s)	Memory usage (%)	p-value (vs. ADSRP)
Dynamic Neighborhood Cleaning	0.71 ± 0.07	1.82 ± 0.33	498.3 ± 38.7	7.2 ± 0.5	39.6 ± 2.8	0.003
Sliding window interpolation	0.68 ± 0.08	1.95 ± 0.37	527.4 ± 41.1	7.5 ± 0.7	42.1 ± 3.0	0.008
DP trajectory simplification	0.75 ± 0.06	1.47 ± 0.29	432.9 ± 35.6	9.8 ± 0.9	35.7 ± 2.5	0.012
DBSCAN clustering	0.78 ± 0.05	1.24 ± 0.25	398.2 ± 32.4	12.3 ± 1.1	31.9 ± 2.2	0.021
SNN similarity metric	0.72 ± 0.07	1.68 ± 0.31	467.5 ± 37.9	8.1 ± 0.8	37.4 ± 2.7	0.004
Full ADSRP model	0.86 ± 0.05	0.98 ± 0.12	362.6 ± 28.3	8.1 ± 0.7	27.5 ± 2.0	-

Table 5: Sensitivity experiments for threshold s

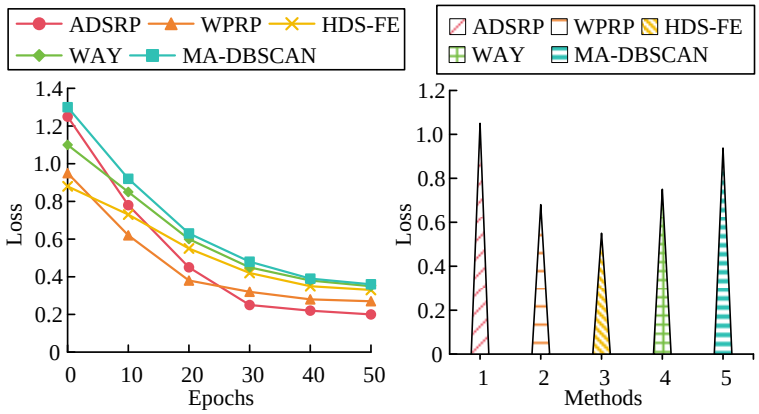
Threshold (s)	Percentage of high similarity routes (%)	Mean range deviation (nmi)	Average fuel consumption (L)	Planning time (s)	p-value (vs. s=0.85)
0.75	98.1 ± 1.2	1.54 ± 0.23	452.8 ± 39.5	6.5 ± 0.5	<0.001
0.8	92.3 ± 2.1	1.22 ± 0.18	402.5 ± 31.7	7.8 ± 0.6	0.004
0.85	78.5 ± 1.8	0.98 ± 0.12	362.6 ± 28.3	8.1 ± 0.7	-
0.9	58.6 ± 3.2	0.89 ± 0.09	340.1 ± 25.9	9.4 ± 0.9	0.018
0.95	34.2 ± 4.5	0.82 ± 0.07	328.7 ± 24.1	12.6 ± 1.3	0.033

In Table 5, when the threshold s=0.85, the range deviation (0.98±0.12 nm) and fuel consumption (362.6± 28.3L) were reduced by 19.7% and 9.9%, respectively, compared to s=0.80 (p=0.004), and the planning elapsed time was stabilized at 8.1 seconds. s=0.75 had a high coverage rate of 98.1% but the deviation spiked to 1.54 nm (p<0.001). s=0.95 had the lowest deviation (0.82±0.07 nm) but the elapsed time increased to 12.6 seconds and only 34.2% (p=0.033). 0.001). s=0.95 had the lowest deviation (0.82±0.07 nautical miles), but the elapsed time increased to 12.6 seconds and the coverage was only 34.2% (p=0.033). s=0.85 achieved the optimal balance of quality, efficiency, and economy, which validated its reliability as a recommended threshold. After that, in order to visualize, the convergence performance of the study model, the study conducted iterative training and the results are shown in Figure 8.

The combination of Figure 8(a) and Figure 8(b) shows that the training loss of each method decreased significantly with rounds, with ADSRP decreasing the most (1.25→0.2, Δ=1.05), followed by MA-DBSCAN (1.3 → 0.36, Δ =0.94). WPRP, HDS-FE and WAY converged at 0.68, 0.55, and 0.75, respectively, with ADSRP having the highest initial loss but fastest convergence after round 30 (0.25→0.2), and significantly better than WAY (0.45→0.35). The loss of ADSRP was stable after the 30th round (0.25 → 0.2), which was significantly better than that of WAY (0.45→0.35). MA-DBSCAN had the highest initial loss but fast convergence, and had the lowest loss at the 50th round (0.36). The stability and convergence efficiency of ADSRP validated its advantage as a recommended model.

4.2 Actual model performance experiment

Although simulation environments could idealize the control of variables, they could not reproduce the dynamic environmental disturbances in real sea areas.Actual model testing could verify the dynamic adaptability and anti-interference ability of the model through real ship interaction scenarios, ensuring the navigation safety and economy of the planned path.Therefore, the study selected Port A and Port B as the endpoints of the trajectory, and chose MA-DBSCAN and WPRP, which performed well in simulation experiments, as the comparative models, while ADSRP was still used as the research object.The embedded edge deployment testing was based on NVIDIA Jetson AGX Xavier (Ubuntu 18.04 LTS, ARMv8.2 architecture), Jetson Nano 4GB (Maxwell architecture with 128-core GPU), and STM32H743VIT6 microcontroller (ARM Cortex-M7 core, 480MHz), covering hardware verification needs from high-performance edge computing to low-power embedded scenarios.The experimental parameters were strictly kept consistent with the simulation run experiment, and the training set used the measured trajectory of Marine Cadastre AIS to ensure comparability of results and support the closed-loop of the model from theoretical verification to practical application.Firstly, the study used different methods to plan the trajectory between A and B, and compared the trajectory with historical high-frequency trajectories, as shown in Figure9.



(a) Training convergence plots (b) Amount of convergence

Figure 8: Comparison of model convergence performance differences

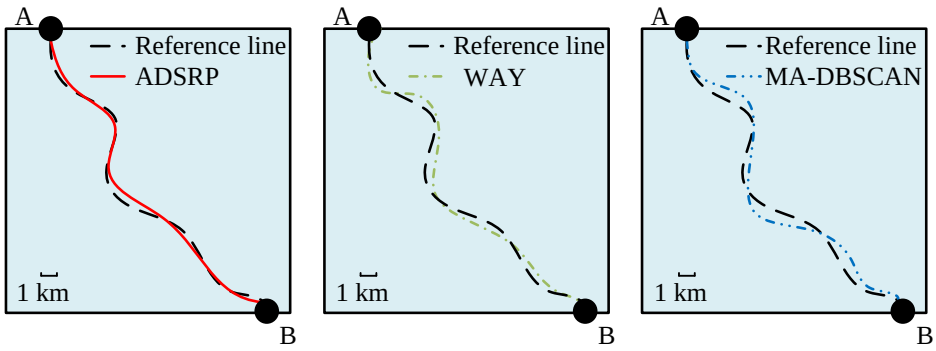
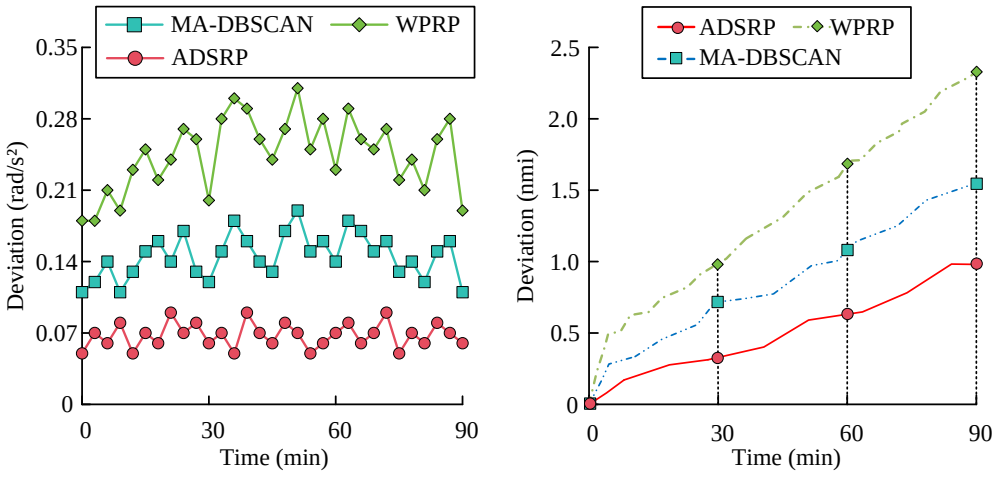


Figure 9: Comparison of different trajectories

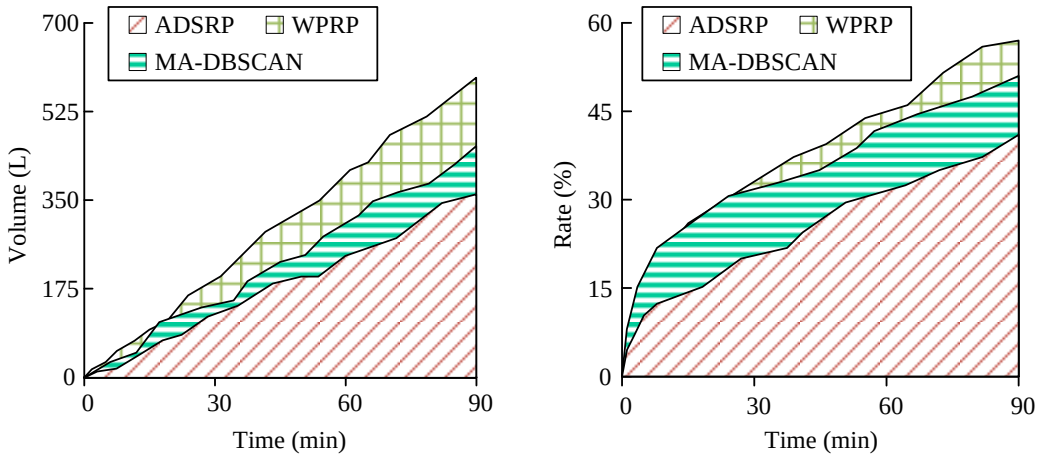
According to Figure 9, the ADSRP trajectory closely followed the dashed reference line throughout the entire process (maximum lateral deviation ≤ 10% of the 1km scale). However, due to probability estimation bias, WAY formed redundant curved detours in the nearshore area (with a lateral offset of 30% from the reference line). Although MA-DBSCAN conformed to the reference line at key turning points, it generated serrated trajectories (curvature radius ≤ 0.3km) due to multi-attribute segmentation clustering. ADSRP minimized range deviation while maintaining heading continuity (actual range was only 2.3% longer than the reference line), while the comparative method required frequent adjustment of speed due to path redundancy or abrupt changes (WAY range extension 8.7%, MA-DBSCAN extension 5.1%). To quantify the reliability of trajectory planning using different methods, a study was conducted to compare the path smoothness deviation and range deviation of different trajectories, as shown in Figure 10.

According to Figure 10 (a), ADSRP optimized heading continuity through spatiotemporal feature fusion, and its average path smoothness deviationwas 0.07 rad/s2 (95% CI [0.05,0.09]), significantly lower thanMA-DBSCAN (0.14rad/s2,P=0.002) and WPRP (0.24 rad/s2, P<0.001).This model utilized multi-head attention weighted speed heading coupling features at route turning points to suppress sudden turns and verify the enhancement effect of SNN structure on trajectory smoothness.According to Figure 10 (b), ADSRP constrained path global consistency through trajectory similarity threshold (s ≥ 0.85), resulting in a total range deviation of 0.98nmi (95% CI [0.92, 1.04]), which was 36.8% lowerthan MA-DBSCAN (1.55,p=0.013).Its encoder effectively captured the inertial characteristics of ship motion along the A-B diagonal main route, reducing redundant evasive maneuvers, and achieving a statistically significant difference in the cumulative deviation growth rate (0.011 nmi/min) of the voyage (p<0.01). Finally, the study compared the fuel consumption and hardware resource utilization of different trajectories to evaluate the actual deployment potential of different planning methods, as shown in Figure 11.



(a) Deviation of path smoothness (b) Course deviation

Figure 10: Actual trajectory planning reliability



(a) Fuel consumption (b) Memory footprint

Figure 11: Analysis of actual deployment potential

Table 6: Comparison of environmental validation of historical data

Year of data	Methods	Track endpoint recognition accuracy	Processing delay (ms)	Prediction error (nmi)
2021	ADSRP	93.20%	182±19	0.91±0.11
	MA-DBSCAN	84.50%	295±38	1.32±0.23
	WPRP	78.70%	258±31	1.49±0.30
2022	ADSRP	91.60%	189±22	0.95±0.13
	MA-DBSCAN	81.30%	308±42	1.38±0.26
	WPRP	73.40%	271±35	1.57±0.33
2023	ADSRP	89.80%	196±25	1.03±0.15
	MA-DBSCAN	78.10%	322±47	1.45±0.29
	WPRP	69.50%	285±39	1.68±0.37
Safety threshold		⩾85%	⩽200ms	⩽1.2 nmi

According to Figure 11 (a), ADSRP utilized trajectory similarity constraints ($s \geq 0.85$) and SNN feature fusion to dynamically optimize the ship's power curve. The average fuel consumptionwas 362.58L (95%CI [355.2369.9]), which was 20.8% and 38.8% lowerthan MA-DBSCAN (457.89L, $p=0.003$) and WPRP (592.98L, $p<0.001$), respectively. The speed heading coupling attention mechanism effectively suppressed redundant steering, with a fuel efficiency standard deviation of only 12.7L. As shown in Figure 11 (b), the average memory occupancy of ADSRPwas 27.5% (95%CI[25.1, 29.9]), significantly lower than that of MA-DBSCAN (36.5%,$p=0.008$) and WPRP (42.5%, $p<0.001$).DBSCAN in ADSRP could reduce intermediate state storage requirements, with a memory usage growth rate (0.43%/min) that was 14% lower than traditional clustering algorithms (0.5%/min), making it suitable for edge device deployment.In addition, the study used the historical AIS data from the endpoints of this port from 2021-2023 to validate the generalization performance across years and test the real-time processing capability, and the results are shown in Table 6.

In Table 6, the accuracy of ADSRP for trajectory endpoint identification in three years of data was 93.20%, 91.60%, and 89.80%, respectively, which was always above the 85% safety threshold, and the prediction error (0.91-1.03 nmi) was better than that of the comparison methods. The processing latency gradually increased from 182ms (2021) to 196ms (2023), which still met the ⩽ 200ms real-time requirement. The latency of both MA-DBSCAN and WPRP exceeded the thresholds (up to 322ms/285ms in 2023) and the prediction error broke through 1.45 nmi/1.68 nmi, which indicated that the performance of the traditional methods declined in the presence of interannual data significantly.

5 Discussion

In response to the insufficient performance of traditional trajectory planning methods, an ADSRP model was proposed to clean AIS shipping big data and implement trajectory endpoint clustering and similarity screening based on methods such as DBSCAN.Experimentsshowed that in simulation, the average cosine similarity of ADSRP reached 0.86 (95% CI[0.83, 0.89]), which was 12% -34% higher than the comparison model($p<0.001$), attributed to the accuracy of DBSCAN in port endpoint clustering (93.6% accuracy at ϵ=500m and MinPts=5) and the spatiotemporal feature fusion of SNN.In actual scenarios, the range deviation of ADSRP was only 0.98nmi (36.8% lower than MA-DBSCAN), fuel consumption was 362.58L (20.8% lower), and the average memory usage was 27.5% (24.7% lower than traditional methods),

verifying its dynamic adaptability and deployment potential.The result metrics demonstrated that range deviation and fuel consumption directly reflected the global optimality of generated routes, smoothness validated dynamic obstacle avoidance capability, while trajectory fit served only as auxiliary validation of historical pattern learning. Collectively, they proved the comprehensive optimization of generated paths in safety, economy, and dynamic adaptability. From the above, DBSCAN accurately recognized the port area based on density sensing and avoided the limitation of the number of preset clusters. Multi-attribute clustering fused speed/heading to enhance the feature correlation, but the high-dimensional distance computation exacerbated the complexity, and the sparse sea area was susceptible to the interference of redundant dimensions. The sliding window interpolation retained the local trend but ignored the nonlinear motion, which may introduce smoothing bias. DP algorithm compressed the trajectory through geometric thresholding, which required dynamic trade-offs between the retention of the key steering points and computational efficiency. SNN suppressed the noise interference by dynamically weighting the spatiotemporal features through multi-attention; and the shared weight encoding with ternary loss enhances the learning ability of the metrics. In practical applications, the model relied on AIS real-time/historical data, and real-time computing resources limited edge deployment.

6 Conclusion

In summary, A route planning framework for multi-scale density trajectories was developed, which solved the contradiction between traditional methods in terms of noise robustness, computational efficiency, and trajectory fit. The proposed method achieved some results in ship route planning, however, there are still limitations. First, DBSCAN's parameter sensitivity and neglect of multi-attribute correlations limited clustering robustness. Second, sliding-window interpolation induced nonlinear motion smoothing bias, and DP over-compression sacrificed critical steering points. Third, traditional models failed to capture spatiotemporal dependencies, and SNN's computational demanded hinder edge deployment. Fourth, reliance on AIS data quality, manual parameter tuning, and dynamic response lagged constrain practical applications.

Aiming at the above problems, the future work of the research lies in: First, develop a hybrid clustering framework integrating DBSCAN density-awareness with multi-attribute weighting and Bayesian-optimized parameter tuning (maximizing Silhouette Score), to improve the robustness of clustering in sparse sea area and high-dimensional data. Second, design LSTM-based adaptive interpolation for nonlinear patterns and curvature-constrained DP algorithms to balance key-point retention. Third, optimize SNN via lightweight attention modules and knowledge distillation while enhancing small-sample generalization through contrastive learning. Fourth, integrate meteorological/bathymetric data fusion and deploy edge-end lightweight ADSRP via federated learning for real-time distributed planning.

References

[1] Daoke Li. Fermat curve path planning method for ship trajectory tracking. Informatica, 48(8):193-206, 2024. https://doi.org/10.31449/inf.v48i8.5735

[2] Juan Li. Improved genetic algorithm enhanced with generative adversarial networks for logistics distribution path optimization. Informatica, 49(11):67-82, 2025. https://doi.org/10.31449/inf.v49i11.6961

[3] Lingxiao Wu, Yossiri Adulyasak, Jean-François Cordeau, and Shuaian Wang, Vessel service planning in seaports. Operations Research, 70(4): 2032-2053, 2022. https://doi.org/10.1287/opre.2021.2228

[4] Tianyang Zhao, Jiayu Qiu, Shuli Wen, and Miao Zhu. Efficient onboard energy storage system sizing for all-electric ship microgrids via optimized navigation routing under onshore uncertainties. IEEE Transactions on Industry Applications, 58(2):1664-1674, 2022. https://doi.org/10.1109/TIA.2022.3145775

[5] Madhushree Das, Arindam Roy, Samir Maity, Samarjit Kar, and Shatadru Sengupta. Solving fuzzy dynamic ship routing and scheduling problem through new genetic algorithm. Decision Making: Applications in Management and Engineering, 5(2):329-361, 2022. https://doi.org/10.31181/dmame181221030d

[6] Chan Woo Han, Sung Wook Lee, and Eun Seok Jin. Tracking of ARPA radar signals based on UK-PDAF and fusion with AIS data. Journal of Ocean Engineering and Technology, 37(1):38-48, 2023. https://doi.org/10.26748/KSOE.2022.046

[7] Eshan Bajal, Vipin Katara, Madhulika Bhatia, and Madhurima Hooda. A review of clustering algorithms: Comparison of DBSCAN and K-mean with oversampling and t-SNE. Recent Patents on Engineering, 16(2):17-31, 2022. https://doi.org/10.2174/1872212115666210208222231

[8] Adan Wu, Tao Che, Xin Li, and Xiaowen Zhu. Routeview: An intelligent route planning system for ships sailing through Arctic ice zones based on big Earth data. International Journal of Digital Earth, 15(1):1588-1613, 2022. https://doi.org/10.1080/17538947.2022.2126016

[9] Liang Zhao, Fang Wang, and Yong Bai. Route planning for autonomous vessels based on improved artificial fish swarm algorithm. Ships and Offshore Structures, 18(6):897-906, 2023. https://doi.org/10.1080/17445302.2022.2081423

[10] Wei Zhao, Hongbo Wang, Jianning Geng, Wenmei Hu, Zhanshuo Zhang, and Guangyu Zhang. Multi-objective weather routing algorithm for ships based on hybrid particle swarm optimization. Journal of

Ocean University of China, 21(1):28-38, 2022. https://doi.org/10.1007/s11802-022-4709-8

[11] Mengxia Li, Junmin Mou, Yixiong He, Xiaohan Zhang, Qinqiong Xie, and Pengfei Chen. Dynamic trajectory planning for unmanned ship under multi-object environment. Journal of Marine Science and Technology, 27(1):173-185, 2022. https://doi.org/10.1007/s00773-021-00825-x

[12] Hongjie Sha, Zhen Han, and Li'an Zhu. The whole process route planning algorithm based on AIS spatio-temporal big data analysis. Ships and Offshore Structures, 1(1):1-15, 2024. https://doi.org/10.1080/17445302.2024.2386883

[13] Miao Gao, Jinqiang Bi, Zhen Kang, Shuai Chen, Peiru Shi, Xi Zeng, and Anmin Zhang. Ship navigable route framework extraction using high-density searching from AIS big data. The Journal of Navigation, 1(1): 1-19, 2025. https://doi.org/10.1017/S0373463325000050

[14] Jin Sob Kim, Hyun Joon Park, Wooseok Shin, Dongil Park, and Sung Won Han. WAY: Estimation of vessel destination in worldwide AIS trajectory. IEEE Transactions on Aerospace and Electronic Systems, 59(5):5961-5977, 2023. https://doi.org/10.1109/TAES.2023.3269729

[15] Xiaofeng Xu, Deqaing Cui, Yun Li, and Yingjie Xiao. Research on ship trajectory extraction based on multiattribute DBSCAN optimisation algorithm. Polish Maritime Research, 28(1):136-148, 2021. https://doi.org/10.2478/pomr-2021-0013

[16] Yiqi Zhang, Yuan Chang, Changbo Wang, Jimmy C. H. Fung, and Alexis K. H. Lau. Life-cycle energy and environmental emissions of cargo ships. Journal of Industrial Ecology, 26(6):2057-2068, 2022. https://doi.org/10.1111/jiec.13293

[17] Weihua Xu, Kehua Yuan, and Wentao Li. Dynamic updating approximations of local generalized multigranulation neighborhood rough set. Applied Intelligence, 52(8):9148-9173, 2022. https://doi.org/10.1007/s10489-021-02861-x

[18] Changbo Hou, Guowei Liu, Qiao Tian, Zhichao Zhou, Lijie Hua, and Yun Lin. Multisignal modulation classification using sliding window detection and complex convolutional network in frequency domain. IEEE Internet of Things Journal, 9(19):19438-19449, 2022. https://doi.org/10.1109/JIOT.2022.3167107

[19] Lixiang Zhang, Yian Zhu, Jie Ren, Wei Lu, and Ye Yao. A method for detecting abnormal behavior of ships based on multi-dimensional density distance and an abnormal isolation mechanism. Mathematical Biosciences and Engineering, 20(8):13921-13946, 2023. https://doi.org/10.3934/mbe.2023620

[20] Praphula kumar Jain, Mani Shankar Bajpai, and Rajendra Pamula. A modified DBSCAN algorithm for anomaly detection in time-series data with seasonality. International Arab Journal of Information Technology, 19(1):23-28, 2022. https://doi.org/10.34028/iajit/19/1/3

[21] Ahmed Arafa, Nawal El-Fishawy, Mohammed Badawy, and Marwa Radad. RN-SMOTE: Reduced Noise SMOTE based on DBSCAN for enhancing imbalanced data classification. Journal of King Saud University Computer and Information Sciences, 34(8):5059-5074, 2022. https://doi.org/10.1016/j.jksuci.2022.06.005

[22] Kavita Bhosle, and Vijaya Musande. Evaluation of deep learning CNN model for recognition of devanagari digit. Artificial Intelligence and Applications, 2023, 1(2):98-102, 2023. https://doi.org/10.47852/bonviewAIA3202441

[23] Nicolás Serrano, and Alejandro Bellogín. Siamese neural networks in recommendation. Neural Computing and Applications, 35(19):13941-13953, 2023. https://doi.org/10.1007/s00521-023-08610-0

[24] David L. Young, Brandan Scully, and Katherine F. Chambers. Resilience-focused analysis of the United States maritime transportation system using automatic identification system data. Transportation Research Record, 2676(7):726-742, 2022. https://doi.org/10.1177/03611981221082561