A machine learning method for the recognition of ship behavior using AIS data

Downloaded from: https://research.chalmers.se, 2024-12-19 14:43 UTC

Citation for the original published paper (version of record):
Ma, Q., Lian, S., Zhang, D. et al (2025). A machine learning method for the recognition of ship behavior using AIS data. Ocean Engineering, 315. http://dx.doi.org/10.1016/j.oceaneng.2024.119791

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

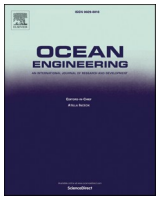
(article starts on next page)

Ocean Engineering 315 (2025) 119791

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng



Research paper

A machine learning method for the recognition of ship behavior using AIS data

Quandang Ma [a], Sunrong Lian [a], Dingze Zhang [b], Xiao Lang [c], Hao Rong [d], Wengang Mao [c], Mingyang Zhang [e,*]

[a] Hubei Key Laboratory of Inland Shipping Technology, School of Navigation, Wuhan University of Technology, Wuhan, China
[b] School of Information Engineering, Wuhan University of Technology, China
[c] Department of Mechanics and Maritime Sciences, Chalmers University of Technology, Gothenburg, Sweden
[d] Centre for Marine Technology and Ocean Engineering (CENTEC), Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, Lisboa, Portugal
[e] Department of Mechanical Engineering, School of Engineering, Aalto University, Espoo, Finland

ARTICLE INFO

Keywords:
Maritime traffic safety
Ship behavior recognition
AIS data processing
Clustering algorithm
Machine learning

ABSTRACT

The efficiency of maritime traffic management and the safety of ship navigation have become top priorities. This study introduces a ship behavior recognition method that utilizes the Extreme Gradient Boosting (XGBoost) classification model, in conjunction with the Sparrow Search Algorithm (SSA), to enhance proactive maritime traffic management. The method leverages Automatic Identification System (AIS) data as its primary source and involves several critical steps. Initially, the AIS data is preprocessed, and ship behaviors are encoded. Subsequently, the encoded behaviors are clustered using spectral clustering to create a labeled dataset. Then, this dataset is employed to train and validate the SSA-XGBoost classification algorithm for identifying ship behaviors. Finally, an example analysis is performed in the Yangtze River. The results indicate that the proposed method can accurately and swiftly identify ship behaviors, achieving an accuracy of 97.28%, precision of 96.97%, recall of 97.43%, and an F1 score of 97.19%, surpassing the performance of the existing algorithms. The findings have the potential to aid maritime supervision authorities in promptly assessing ship navigation statuses and provide a valuable reference for developing ship scheduling decisions.

1. Introduction

Maritime transportation is crucial to global trade and the economy, facilitating the international movement of goods and supporting complex global supply chains (Svanberg et al., 2019; Verschuur et al., 2022; Liu et al., 2024; Zhang et al., 2024b). However, the sharp growth in trade volume and the increasing number of ships have heightened the need for efficient maritime traffic management and enhanced safety in ship navigation (Ma et al., 2023; Fu et al., 2023). Current maritime traffic management depends on traditional Vessel Traffic Service (VTS) systems, which are crucial for navigation monitoring. However, the growing shipping volumes and increasing number of vessels pose challenges for VTS systems in processing real-time data and responding promptly. These systems often depend on manual operations, complicating their ability to manage complex navigational environments and dynamic traffic conditions. Introducing machine learning-based systems can offer maritime authorities enhanced data analytics for real-time decision-making, optimizing resource allocation, and improving navigational safety and efficiency. By understanding ship behavioral patterns and dynamics, maritime authorities can strengthen the supervision of navigational risk hotspots, accurately assess traffic conditions and risks, and improve overall traffic management capabilities (Rong et al., 2024). Thus, research on ship behavior identification is essential for proactive maritime traffic management.

Ship behavior refers to the manner in which a ship maneuvers and the principles governing its movement under the direction of the crew for navigation and avoidance purposes. Current research on ship behavior typically focuses on two scales: macro and micro. Macro-analysis examines movement patterns of ship fleets on a global scale, while micro-analysis focuses on individual ship behaviors on a local scale (Zhou et al., 2023, 2024; Wang et al., 2024). Studying the collective behavior of ship groups facilitates the identification of universal patterns and governing principles.

Research on ship behavior recognition has primarily employed three

* Corresponding author. Otakaari 4, 02150, Koneteknikka 1, Espoo, Finland.
E-mail address: mingyang.0.zhang@aalto.fi (M. Zhang).

https://doi.org/10.1016/j.oceaneng.2024.119791
Received 16 August 2024; Received in revised form 2 November 2024; Accepted 12 November 2024
Available online 23 November 2024
0029-8018/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

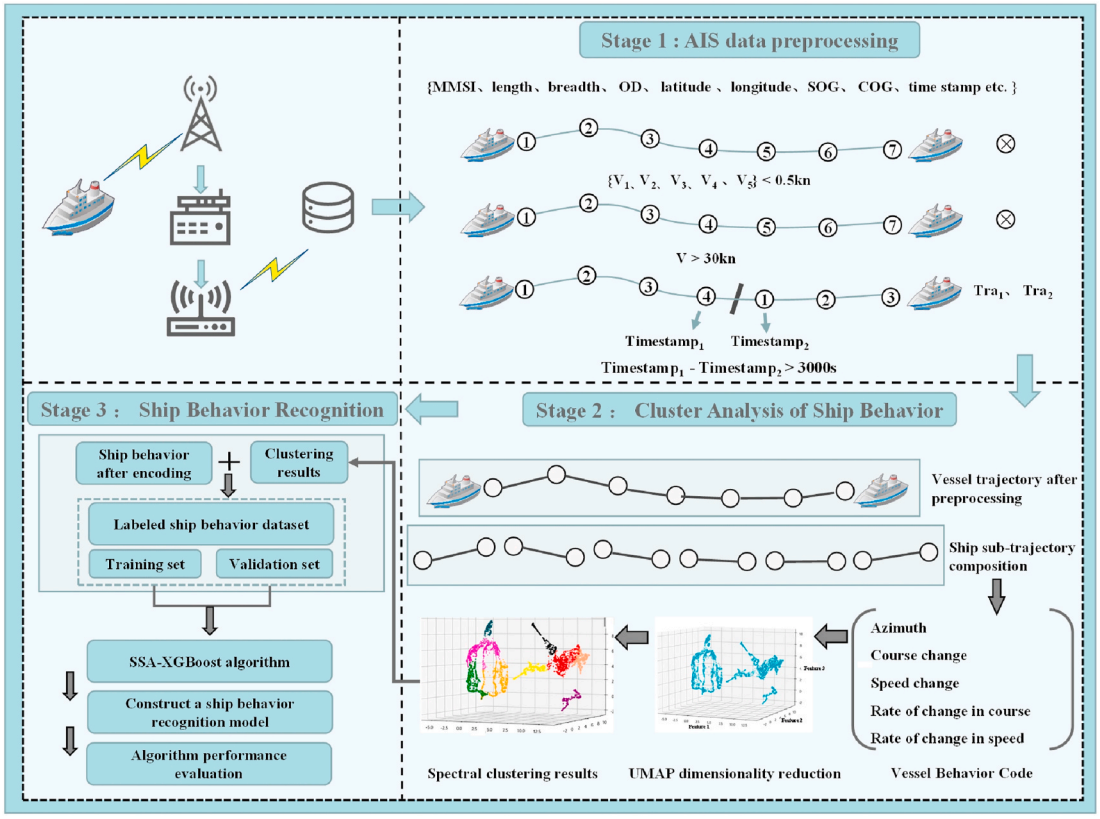


Fig. 1. Ship behavior identification flowchart.

model types: (a) probabilistic statistical models, (b) unsupervised learning models, and (c) supervised learning models.

Probabilistic statistical models have a clear mathematical foundation and theoretical support to identify ship behaviors. For instance, Castaldo et al. (2014) used Dynamic Bayesian Networks to automatically identify anomalies in port environments. Tang et al. (2020) developed a probabilistic directed graph model to detect ship states based on historical AIS data and node state features. Carlson et al. (2021) used a multinomial Hidden Markov Model (HMM) to classify early hostile behaviors by encoding the rate of change in a ship course. Murray et al. (2022) employed Gaussian mixture models to classify ship trajectories and predict behaviors. However, these models often rely on simplifying assumptions that may not fully capture the complexity of ship behavior using simplified mathematical method, especially given the high-dimensional nature of ship behavior data, which includes multi-dimensional features such as position, Speed Over Ground (SOG), and Course Over Ground (COG).

Unsupervised learning models can automatically extract knowledge from large amounts of unlabeled data for the recognition of ship behaviors. For example, Wang et al. (2021) improved the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to develop Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) for clustering ship trajectories, facilitating maritime supervision in complex waters. Wei et al. (2024) introduced a multidimensional Dynamic Time Warping (DWT) metric to measure trajectory similarity and used DBSCAN for clustering analysis. Liu et al. (2023) proposed an unsupervised method based on AIS data for motion behavior extraction and voyage pattern mining. Despite their ability to discover patterns from unlabeled data, these models are sensitive to data quality, and noise or outliers can significantly impact their performance.

Supervised learning models, trained using labeled data, typically achieve high recognition accuracy. For example, Huang et al. (2019) combined K Nearest Neighbor (KNN) and Local Outlier Factor (LOF) algorithms to detect ship behavior. Kim et al. (2015) processed ship trajectory data using Support Vector Machines (SVM), optimizing parameters through K-fold cross-validation and grid search. However, traditional machine learning methods often struggle with complex maritime traffic data (Dong et al., 2020), such as unbalanced or high-dimensional datasets, limiting their ability to capture multidimensional features and underlying structures.

Currently, the methods used for maritime traffic management (VTS) often rely on manual monitoring and decision-making when dealing with high traffic and complex channel environments, which can lead to lagging response times. They often lack the ability to flexibly adapt to complex or dynamic environments, such as narrow waterways, bends or harbor areas, which may not be able to adequately identify and handle all potential risks. In order to effectively utilize the ship behavioral features embedded in ship AIS data for ship behavior identification and avoid potential risks, this paper propose the use of integrated learning algorithms in supervised learning to address this issue. Integrated learning is widely used and effective in the field of big data mining, building classification models and pattern recognition research (Qu et al., 2019; Tan et al., 2020; Li et al., 2020). The reason for this is that integrated learning is good at solving data recognition or classification problems with explicit features, and it can improve the efficiency of the whole system. We utilize the Extreme Gradient Boosting (XGBoost) classification algorithm and optimize its hyperparameters using the Sparrow Search Algorithm (SSA) to enhance recognition accuracy. This study introduces a integrated learning algorithm based on Uniform Manifold Approximation and Projection (UMAP) and spectral clustering. The method involves encoding, dimensionality reduction, clustering, and visualization of ship behavior, considering parameters such as azimuth, speed change, course change, speed change rate, and course change rate. This approach helps in understanding and elucidating ship maneuvering behavior patterns embedded in AIS data. A recognition method combining SSA and XGBoost is proposed. SSA optimizes

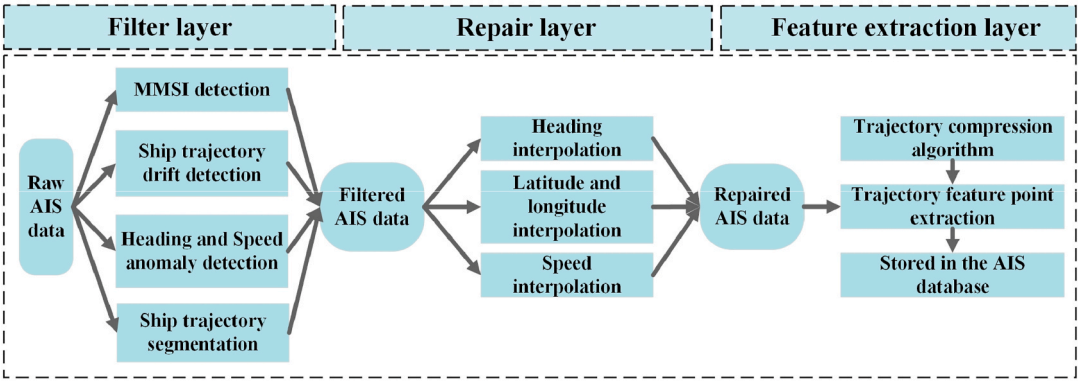


Fig. 2. AIS data preprocessing flowchart.

XGBoost parameters, eliminating manual parameter setting influences. This method achieves high-precision recognition of ship behavior, enhancing machine learning efficiency in ship collision avoidance, route planning, and abnormal behavior detection.

This paper is organized as follows: Section 2 details the methods, including ship AIS data preprocessing, ship behavior clustering algorithm, ship behavior pattern recognition algorithm, and recognition performance evaluation. Section 3 validates the proposed method with a case study conducted in the Yangtze River and analyzes the relevant conclusions. Section 4 summarizes the conclusions of this paper and outlines future research directions.

2. Methodology

The objective of this study is to uncover the behavioral patterns of ships as recorded in AIS data and to identify the behaviors and maneuvering patterns of ships in maritime environments. This research aims to provide crucial theoretical support for intelligent supervision and navigation safety within the context of big data and intelligent shipping. Additionally, it holds significant implications for enhancing the efficiency of maritime traffic supervision and reducing the risk of ship navigation. The flowchart illustrating the ship behavior recognition process proposed in this paper is presented in Fig. 1. This process comprises three main stages.

- **Stage 1:** AIS Data Preprocessing: Raw AIS data are filtered, interpolated, and compressed. The processed trajectories are then stored in the AIS database.
- **Stage 2:** Cluster Analysis of Ship Behavior: Ship behaviors are encoded based on the preprocessed AIS data. UMAP and spectral clustering algorithms are employed to reduce dimensionality and cluster the ship behaviors.
- **Stage 3:** Ship Behavior Recognition: The encoded ship behaviors, along with the clustering results, serve as the data source for the SSA-XGBoost algorithm. This process constructs a ship behavior recognition model and facilitates the identification of ship behaviors.

2.1. Stage 1: AIS data preprocessing

The AIS is essential for monitoring the maritime environment, ship transport, ship management, and other related activities (Fu et al., 2017; Li et al., 2018; Emmens et al., 2021). AIS can send and receive critical ship-related information, including Maritime Mobile Service Identity (MMSI), ship position, speed, course, and destination port (Bailey et al., 2008). This system enhances collision avoidance capabilities between ships and improves the safety and reliability of navigation. Additionally, AIS systems are widely used as a data source for maritime traffic analysis (Liu et al., 2020; Yu et al., 2020; Zhang et al., 2022a; Liang et al., 2022; Ma et al., 2024), collision risk prevention (Greidanus et al., 2016; Alessandrini et al., 2018; Liu et al., 2021, 2023a), ship behavior identification(Zhang et al., 2022b, 2024; Rong et al., 2022; Liu et al., 2023b), and various other research applications. It is worth noting that Liu et al. (2022, 2024) open the door to research questions on the automatic identification and analysis of icebreaker assistance operations in ice-covered waters, as well as the need for such assistance, through the integration of AIS data and ice data using multi-source data fusion method. However, AIS data transmission involves multiple nodes, such as satellite communication, AIS base stations, decoders, and routers, which can compromise the authenticity, completeness, and accuracy of the data. This may result in incorrect or missing key information,

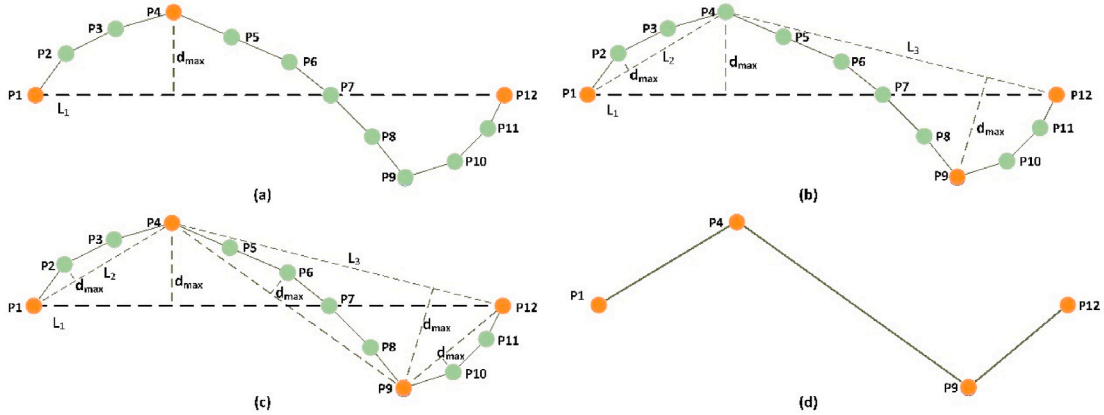


Fig. 3. Schematic diagram of the DP algorithm.

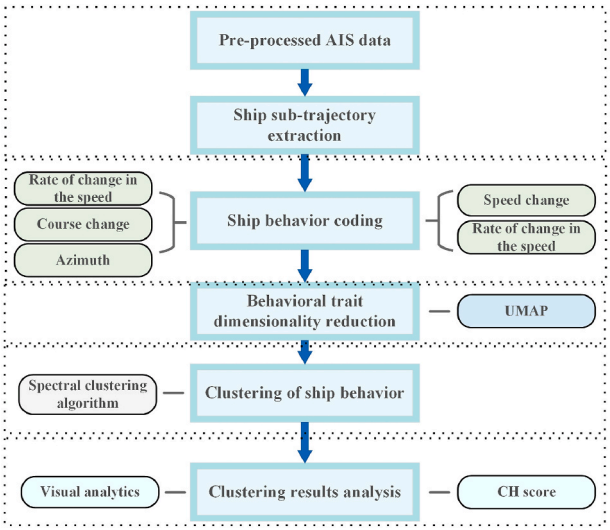


Fig. 4. Flowchart for clustering analysis of ship behavior.

including abnormal AIS data values, ship trajectory drift, round-trip sailing, and missing data (Pallotta et al., 2013; Yang et al., 2019).

To address these issues, this paper preprocesses the raw AIS data to obtain effective data for mining and analysis. The main process of AIS data preprocessing is depicted in Fig. 2. The preprocessing framework comprises three parts: a filter layer, a repair layer, and a feature extraction layer.

- Filter Layer: This layer performs preliminary screening of the raw AIS data and eliminates data with obvious errors, such as incorrect MMSI codes and positioning errors.
- Repair Layer: The filtered AIS data are interpolated to equalize the time intervals between each trajectory point and smooth the trajectory. This provides more accurate trajectory feature points for subsequent analysis.
- Feature Extraction Layer: This layer compresses the repaired AIS data and stores the compressed trajectories in the AIS database, ready for further analysis and mining.

Trajectory compression is mainly used to extract the key features of the trajectory, improve the computational efficiency, reduce the data storage space, etc. It is also a commonly used trajectory segmentation method (Tang et al., 2021). Douglas-Peucker (DP) algorithm (Zhao et al., 2019) is a more popular trajectory compression algorithm at the present stage. Its advantage lies in the fact that it can retain the feature points on the larger curvature patterns. It can be adapted to different application scenarios by modifying the threshold value. Therefore, in this paper, the DP algorithm will be used to compress and segment the repaired AIS data to form ship sub-trajectories.

The schematic diagram of the DP compression algorithm is shown in Fig. 3, now given the threshold ε, the original ship trajectory $Traj = \{P_1, P_2, P_3, \ldots, P_m, \ldots, P_{12}\}$, the core idea of the DP compression algorithm is to replace the original trajectory segment by the compressed approximation of the trajectory segment $\overline{P_1P_{12}}$ and to ensure that the perpendicular Euclidean distance $PED(P_m) < \varepsilon$ from the intermediate point P_m, which is at the maximum distance from $\overline{P_1P_{12}}$, to the straight line segment $\overline{P_1P_{12}}$.

If the condition $PED(P_m) < \varepsilon$ is not satisfied, the original problem is decomposed into two sub-problems, and the line segments $\overline{P_1P_m}$, $\overline{P_mP_{12}}$ are processed recursively with P_m as the splitting point. Taking Fig. 3 as an example, where the trajectory point P_4 is the farthest point from $\overline{P_1P_{12}}$ and $PED(P_4) > \varepsilon$, the recursive compression $\overline{P_1P_4}$, $\overline{P_4P_{12}}$. The recursion stops when the maximum perpendicular Euclidean distance of the intermediate nodes of all approximate trajectory segments are all less than a given error threshold ε. The approximate trajectory segment $\overline{P_1P_4}$ in Fig. 3 satisfies $PED(P_2) < \varepsilon$, $PED(P_3) < \varepsilon$, so line segment $\overline{P_1P_4}$ is retained and trajectory points P_2 and P_3 are deleted. The line segment $\overline{P_4P_{12}}$ is processed in the above way, and the approximate trajectory $Traj' = \{P_1, P_4, P_9, P_{12}\}$ is finally obtained.

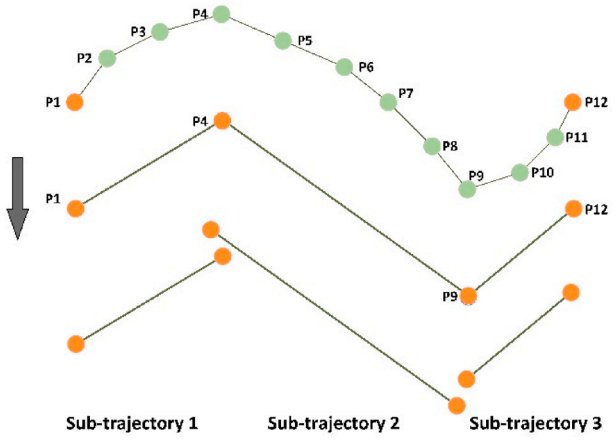


Fig. 5. Sub-trajectory generation graph.

2.2. Stage 2: cluster analysis of ship behavior

Most current studies on ship behavior cluster the overall ship trajectory without considering the changes in navigational state (e.g., speed, course). To intuitively analyze ship behavior and explore the underlying behavioral laws, this paper employs a clustering analysis method using UMAP and spectral clustering. This approach fully considers the dynamic changes in a ship state during navigation.

The process begins with encoding ship behavior to construct a mapping relationship between the ship behavioral characteristics and its sub-trajectories. The encoded ship behaviors are then subjected to visual analysis using the UMAP dimensionality reduction algorithm. Fig. 4 illustrates the flowchart of the proposed ship behavior clustering analysis method. This method can be divided into three main steps: Ship Behavior Coding: This step involves encoding various states of the ship, such as speed and course changes, to represent the ship behavior accurately. Behavior Feature Dimensionality Reduction: Using the UMAP algorithm, the high-dimensional behavior features are reduced to lower dimensions for better visualization and analysis. Ship Behavior Clustering: Spectral clustering is then applied to the dimensionally reduced data to group similar ship behaviors, facilitating a deeper understanding of ship navigation patterns.

This comprehensive approach allows for a more detailed and accurate analysis of ship behaviors, considering the dynamic nature of ship navigation. And the details are presented in the following sections.

2.2.1. Ship behavior coding

The intuitive embodiment of ship behavior is represented by the ship trajectory, which can be considered the path a ship takes within a specific environment. Viewing the trajectory merely as a sequence of data points ordered by time only utilizes the positional information of the ship and overlooks the dynamic changes in ship behavior. Therefore, analyzing ship behavior should include considering azimuth, course, and speed characteristics.

In this paper, we propose a ship behavior coding method to establish a mapping relationship between ship sub-trajectories and ship behaviors. A ship sub-trajectory is a segment of a ship's sailing path in a specific time period or a specific region. As shown in Fig. 5, this study marks the trajectory in segments by taking the start point, end point of the ship's trajectory and the characteristic trajectory points extracted by

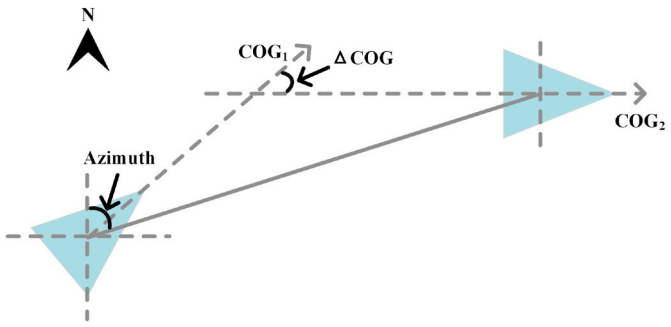


Fig. 6. Schematic diagram of ship behavior characteristics.

the DP algorithm as the initial points, and the trajectory between two adjacent initial points is a trajectory segment. Each sub-trajectory segment represents the smallest unit for storing ship behavior. Ship behavior is then extracted based on the information recorded in each sub-trajectory segment, a process we refer to as ship behavior coding. We propose selecting five elements as ship behavior features and encoding them, as shown in Fig. 6. The reason for selecting these five elements as ship behavior characteristics is that the azimuth angle from the starting point to the end point of the ship's sub-trajectory can characterize the change trend of the ship's position, and when the ship is sailing, the change of its rudder and order is reflected in the ship's behavior as the change of heading, the change of speed, the change of the ship's heading rate, and the change of the ship's speed rate.

This method enables a detailed and dynamic analysis of ship behaviors by considering critical navigational parameters and segmenting trajectories into manageable units for more precise behavior identification. More details, let a certain section of sub-trajectory be coded as $(Azimuth, \Delta COG, \Delta SOG, RC, RS)$, ΔCOG is the change in course, ΔSOG is the change in speed, RC is the rate of change in course, and RS is the rate of change in speed. Among them, the RC and RS are calculated as shown in Eq. (1) and Eq. (2), where $\Delta Time$ is the time interval between two points of the sub-trajectory.

$$RC = \frac{\Delta COG}{\Delta Time} \tag{1}$$

$$RS = \frac{\Delta SOG}{\Delta Time} \tag{2}$$

$\Delta Time$ and the parameter ε in the Douglas-Peucker algorithm are indirectly related in our trajectory simplification process. $\Delta Time$ refers to the time interval between consecutive points in the trajectory, while ε is the maximum distance between the original and simplified curves in the Douglas-Peucker algorithm. In our approach, the trajectory data is first temporally sampled using $\Delta Time$. This step reduces the number of points while maintaining a consistent time interval between points. The choice of $\Delta Time$ affects the granularity of the temporal representation of the trajectory. Subsequently, we apply the Douglas-Peucker algorithm with a specified ε value, which determines the spatial accuracy of the simplification. A smaller ε produces more accurate but less simplified trajectories, while a larger ε leads to a more simplified but possibly less accurate representation.

2.2.2. Behavioral trait dimensionality reduction

The proposed coding methods in Section 2.2.1 are used to encode all ship behaviors, creating a set of high-dimensional data to characterize these behaviors. However, using high-dimensional data increases the computational burden of the subsequent clustering algorithm and poses the risk of the dimensionality catastrophe during the clustering process (Verleysen et al., 2005). To intuitively display the distribution of ship behaviors, avoid the dimensionality catastrophe, and enhance the clustering algorithm's efficiency, it is necessary to reduce the dimensionality of the encoded ship behavior data. This reduction involves projecting the high-dimensional data into a lower-dimensional space using a downscaling algorithm (Sorzano et al., 2014).

Then, the UMAP algorithm (McInnes et al., 2018) among the stream learning is used to downscale the encoded ship behaviors and project the high-dimensional data to the three-dimensional space. The encoded set of ship behaviors is set as $M = \{m_1, m_2, ..., m_n\}$, the elements in this set are high-dimensional data, and the set of ship behaviors output by the UMAP algorithm is $Y = \{y_1, y_2, ..., y_n\}$, the elements in this set are three-dimensional data, and given a hyperparameter k, the set of neighbors of data point m_i in metric space d can be expressed as $\{m_{i1}, m_{i2}, ..., m_{ik}\}$, then the fuzzy topology of the high-dimensional data can be represented using an exponential probability distribution, as shown in Eq. (3).

$$P_{i|j} = exp\left(-\frac{d(m_i, m_j) - \rho_i}{\sigma_i}\right) \tag{3}$$

where, ρ_i is the distance from m_i to the first nearest data point, and σ_i is the diameter of m_i nearest neighbor data point.

At this time, $P_{i|j}$ is not a symmetric function, which has to be symmetrized, and the expression is shown in Eq. (4). After establishing the fuzzy topology in the high-dimensional space, the probability distribution needs to be constructed in the low-dimensional distribution, the expression of which is shown in Eq. (5), where a and b are hyperparameters.

$$P_{i|j} = P_{i|j} + P_{j|i} - P_{i|j}P_{j|i} \tag{4}$$

$$q_{i|j} = \left(1 + a\left(y_i - y_j\right)^{2b}\right)^{-1} \tag{5}$$

The UMAP algorithm wants data points with small differences to be as close together as possible in the low-dimensional projection space, while data points with large differences are as far away as possible in the low-dimensional projection space. Therefore, it is necessary to introduce gravitational and repulsive functions as shown in Eq. (6) and Eq. (7).

$$Attractive = P_{i|j}(X) log\left(\frac{P_{i|j}(X)}{q_{i|j}(Y)}\right) \tag{6}$$

$$Repulsive = \left(1 - P_{i|j}(X)\right) log\left(\frac{1 - P_{i|j}(X)}{1 - q_{i|j}(Y)}\right) \tag{7}$$

where, $P_{i|j}(X)$ is the weight of the data points in the high-dimensional space, and $q_{i|j}(Y)$ is the weight of the data points in the low-dimensional space. the UMAP algorithm firstly applies Attractive force to the points with small differences in the dataset, and applies Repulsive force to the points with large differences, and then gradually reduces the gravitational force and the repulsive force through the simulated annealing optimization algorithm. Finally, the simulated annealing optimization algorithm gradually reduces the gravitational force and repulsive force, and minimizes the loss function to find the optimal solution.

2.2.3. Clustering of ship behavior

This subsection introduces the ship behavior clustering method. The primary objectives for clustering ship behavior are: (1) to objectively discover distribution patterns of different ship behavior categories; (2) to deeply analyze these patterns, which, when combined with kernel density estimation, clearly highlight differences among behavior categories; (3) to categorize ship behavior data, providing high-quality datasets for further behavior identification studies.

Various clustering techniques, including K-means and DBSCAN, have been applied to AIS data. K-means often struggles with ring or concave datasets. DBSCAN, based on density reachability, may merge connected classes into one, and its results are highly sensitive to manually set

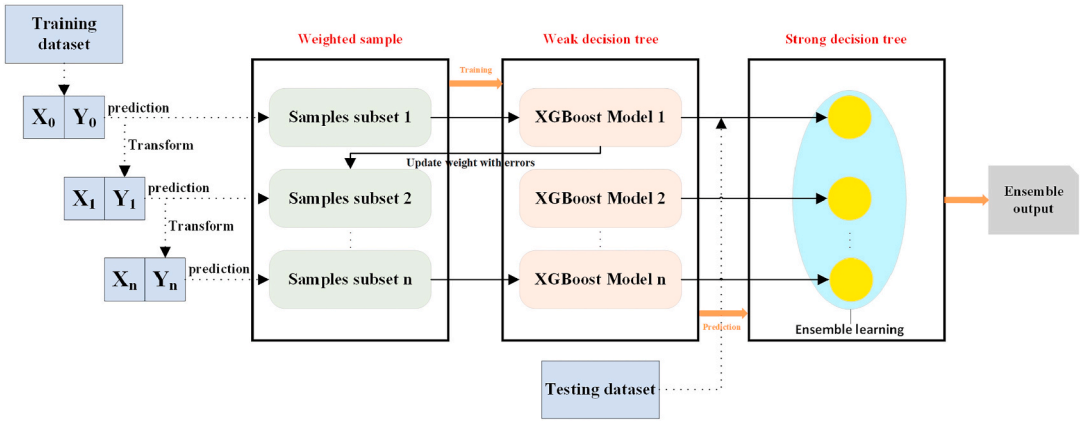


Fig. 7. The structure of Extreme Gradient Boosting (XGBoost) algorithm (Ma et al., 2021).

parameters like neighborhood radius and density threshold.

The spectral clustering algorithm (Von Luxburg et al., 2007), grounded in spectral graph theory, offers advantages over traditional methods. It has low time complexity, is insensitive to input data structure, and can identify non-convex datasets. Unlike K-means or DBSCAN, spectral clustering is suitable for spatial clustering irrespective of shape, achieving globally optimal solutions (Lin et al., 2019) without assuming specific data distribution features. This makes it increasingly popular (Ahn et al., 2016). Thus, this study employs spectral clustering for the analysis of dimensionally-reduced ship behavior data.

The core idea of spectral clustering is to consider the dataset as points in space, with points connected to each other by edges. The weight of each edge is determined by the distance between two points, if the distance between two points is farther then the weight of the edge is smaller, the closer the distance is then the weight of the edge is larger. Now suppose that the input dataset of the spectral clustering algorithm $X = \{x_1, x_2, ..., x_n\}$ contains n sample points, the number of clusters is k, and the output of the algorithm is the clusters $A_1, A_2, ..., A_k$, then the specific computational process of spectral clustering can be described as follows:

(1) Use Eq. (9) to compute the individual elements of the similarity matrix W, which is of size n^*n.

$$W = \begin{pmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nn} \end{pmatrix} \tag{8}$$

$$s_{ij} = s(x_i, x_j) = \sum_{i=1,j=1}^{n} exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma^2} \right) \tag{9}$$

(2) Use Eq. (10) to compute the degree matrix D, which is a diagonal matrix of size n^*n whose elements d_i on the diagonal are the sums of the elements in the i row of the similarity matrix W.

$$d_i = \sum_{j=1}^{n} s_{ij} \tag{10}$$

(3) Calculate the Laplace matrix $L = D - W$.
(4) Calculate the eigenvalues of the Laplace matrix, take the first k eigenvalues, and calculate the eigenvectors $u_1, u_2, u_3, ..., u_k$.
(5) Form the k eigenvectors into a matrix.
(6) Let $y_i \in R^k$ be the i row vector of U, where $i = 1, 2, 3..., n$.
(7) Use the K-means algorithm to cluster the new sample points $Y = \{y_1, y_2, y_3, ..., y_n\}$ into clusters $C_1, C_2, ..., C_k$.
(8) Output the clustering results $A_1, A_2, ..., A_k, A_i = \{j | y_j \in C_i\}$.

Since the spectral clustering algorithm requires manually setting the number of clusters, and this parameter can significantly impact the clustering results, it is essential to determine the optimal number of clusters. To achieve this, the output of spectral clustering with varying numbers of clusters is evaluated using the Calinski-Harabasz (CH) score, where a higher CH score indicates better clustering results.

This paper applies the spectral clustering algorithm to cluster ship behaviors. However, relying solely on the clustering results is insufficient for analyzing the distribution patterns of ship behaviors. The goal of cluster analysis as a descriptive data mining method extends beyond revealing the latent cluster structures in the data; it also involves exploring the underlying mechanisms that generate these structures. Therefore, to delve deeper into the information embedded in the spectral clustering algorithm's output, this paper employs kernel density estimation (Chen et al., 2017) to visualize and analyze the behaviors of different categories of ships concerning azimuth, SOG, and COG.

2.3. Stage 3: ship behavior recognition

In this section, we determine ship behavior characteristics by coding the ship behavior and propose an SSA-XGBoost-based method for ship behavior pattern recognition to address the challenge of recognizing or classifying data with distinct characteristics. This method integrates the clustering results and utilizes the Sparrow Search Algorithm (SSA) to optimize the hyperparameters of the XGBoost classification algorithm. By doing so, it effectively learns the features of different categories of ship behaviors and identifies ship behaviors based on these learned features.

2.3.1. XGBoost

XGBoost, proposed by Chen and Guestrin (2016), is an efficient gradient boosting decision tree algorithm. Its core principle involves using the concept of Boosting, where multiple weak learners are integrated into a strong learner. This is achieved by using multiple trees to make collective decisions. The result of each tree represents the difference between the target value and the prediction results of all previous trees, and the cumulative results of all trees provide the final prediction. This approach improves the overall model performance. XGBoost is composed of multiple Classification and Regression Trees (CARTs), making it a highly flexible and versatile tool for addressing classification and regression problems. The structure of the algorithm is illustrated in Fig. 7.

XGBoost is an additive model consisting of K base models and its model expression can be expressed as:

$$\hat{y}_i = \phi(X_i) = \sum_{k=1}^{K} f_k(X_i), f_k \in F \tag{11}$$

where, $\hat{y}_i$ is the predicted value of the i-th sample, X_i is the data of the i-th sample, K is the number of regression tree models, f_k denotes the k-th tree, $f_k(X_i)$ is the score of the i-th sample in the k-th tree, and F is the set of regression tree models.

XGBoost is similar to most machine learning models in that its objective function consists of two parts: the total sample loss, and the canonical term, which can be expressed as:

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \Omega(f_i) \tag{12}$$

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} {\omega_j}^2 \tag{13}$$

where, $L(\phi)$ is the objective function, $l(y_i, \hat{y}_i)$ is the loss function, which is used to calculate the difference between the actual value y_i and the predicted value $\hat{y}_i$ and $\Omega(f)$ is a regularization term used to control the model complexity and prevent the model from being overfitted. T represents the number of leaves in the decision tree model, ω_j is the weight of leaf nodes in the j-th tree, γ is a hyperparameter that controls the number of leaf nodes and λ is the L2 regularization hyperparameter used to control the weight of the leaf nodes

To train the error function, a forward distribution algorithm is used, where a new function is added to the model in each iteration round and the newly generated tree fits the residuals predicted by the tree model in the previous round. The forward distribution algorithm can be expressed as

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{14}$$

where, $\hat{y}_i^{(t)}$ is the predicted value of the i-th sample in the t-th round, and f_t is the t-th tree, and the current residuals are fitted. The forward distribution algorithm is substituted into the objective function, and the iteratively updated objective function can be expressed as:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \tag{15}$$

where, $\hat{y}_i^{(t-1)}$ is the predicted value of the ith sample in the $t-1$ round, $f_t(x_i)$ is the prediction value of the i-th sample by the t round fitting tree, and $l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right)$ is the loss function, which is used to measure the error between the true value y_i and the updated predicted value. $\Omega(f_t)$ are regularized terms that control the complexity of the new tree f_t.

In order to facilitate optimization, the loss function is approximated by the second-order Taylor expansion

$$l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) \approx \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] \tag{16}$$

$$g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right) \tag{17}$$

$$h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right) \tag{18}$$

where, g_i is the first order derivative of the loss function and h_i is the second order derivative of the loss function.

The objective function can be approximated as:

$$L^{(t)} \approx \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{19}$$

Since $l\left(y_i, \hat{y}_i^{(t-1)}\right)$ is independent of $f_t(x_i)$ and can be ignored, the objective function can be further reduced to

$$L^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{20}$$

2.3.2. Sparrow search algorithm

Sparrow Search Algorithm (Xue et al., 2020) is a group intelligence optimization algorithm derived from simulating the foraging process of a sparrow flock, in which the sparrows in the flock are mainly classified into producers, scroungers and vigilant sparrows.

The producer is the dominant position in the sparrow group, the proportion of the group is generally 10–20%, responsible for finding food for the whole group and provide the direction of the food and the area where the food is available; the scrounger will monitor the producer all the time, once the producer finds the food, the scrounger will immediately follow the producer to seize the food; the vigilant sparrows is responsible for monitoring the area around the foraging area. When a predator is present around the foraging area, the Vigilant sparrows will give an immediate warning, and when the warning signal exceeds the alarm value, the entire population will move to the next foraging site under the leadership of the producer. The mathematical model of SSA starts by assigning a matrix of sparrows' positions as follows:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ \vdots \\ X_i \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & \cdots & \cdots & x_{1j} \\ x_{21} & x_{22} & \cdots & \cdots & \cdots & x_{2j} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1} & x_{i2} & \cdots & \cdots & \cdots & x_{ij} \end{bmatrix} \tag{21}$$

where, x_{ij} denotes the position of the i-th sparrow in dimension j. The adaptation value F_X of all sparrows can be expressed as follows:

$$F_X = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ \vdots \\ f_i \end{bmatrix} = \begin{bmatrix} f[x_{11} & x_{12} & \cdots & \cdots & \cdots & x_{1j}] \\ f[x_{21} & x_{22} & \cdots & \cdots & \cdots & x_{2j}] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f[x_{i1} & x_{i2} & \cdots & \cdots & \cdots & x_{ij}] \end{bmatrix} \tag{22}$$

In the sparrow search algorithm, the sparrow with a higher fitness value is the producer, which can indicate the foraging location for all the scroungers, and the producer is able to obtain a larger foraging search range compared to the scroungers, discover potentially excellent solutions, and pass them on to the scroungers. The producer position is represented by Eq. (23):

$$X_{ij}^{(t+1)} = \begin{cases} X_{ij}^{(t)} e^{\left(\frac{-i}{\alpha t_{max}}\right)}, & R_2 < ST \\ X_{ij}^{(t)} + Q.L, & R_2 \geq ST \end{cases} \tag{23}$$

where, $X_{ij}^{(t)}$ is the position of the j-th dimension of the i-th sparrow in generation t. α is a constant that controls the step size of the exploration. t_{max} is the maximum number of iterations, R_2 is a random number in the interval $(0,1)$. Q is a constant that is used to control the movement step size. L is a random number that obeys a normal distribution and is used to add random perturbations. ST is the threshold that determines whether or not a predator will appear. When $R_2 < ST$ means that the sparrow conducts a broad search, and the position decays exponentially, controlling the range of exploration. When $R_2 \geq ST$, it means that the sparrows are affected by the warning signal and move randomly to avoid the threat of predators.

Based on the solutions provided by the producer, as well as their own search experience, the scroungers refine the search space, aiming to find better solutions in the search space, and pass them on to the vigilant sparrows. The scrounger position is represented by Eq. (24):

$$X_{ij}^{(t+1)} = \begin{cases} Q \cdot e^{\left(\frac{X_{worst}^{(t)} - X_{ij}^{(t)}}{i^2}\right)} & i > \frac{n}{2} \\ X_p^{(t+1)} + \left|X_{ij}^{(t)} - X_p^{(t+1)}\right| \cdot A^{+} & \text{Otherwise} \end{cases} \tag{24}$$

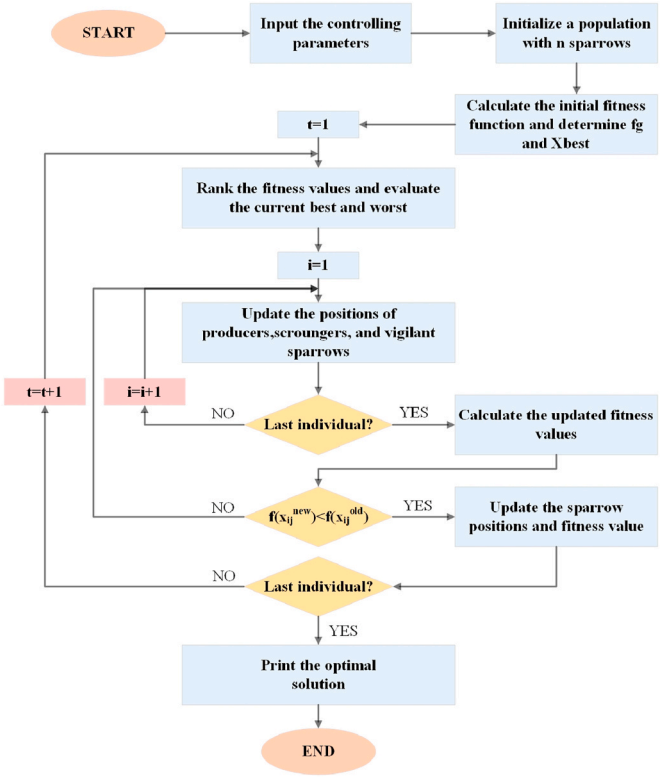


Fig. 8. the flowchart of Sparrow Search Algorithm (SSA).

where, $X_{worst}^{(t)}$ indicates the position with the worst fitness among all sparrows in the t-th iteration of the population. $X_P^{(t+1)}$ is the current optimal location or finder position. A^+ is a random number that obeys a normal distribution and is used to adjust the search step size. When $i > \frac{n}{2}$, the scrounger is located in the second half of the population. At this point, they use an exponential decay strategy to move to the least fit position. Otherwise indicate that the scrounger is located in the first half of the population. At this point, they tend to be in the optimal position.

Vigilant sparrows are the control centre of the sparrow search algorithm and are responsible for supervising and coordinating the activities of producers and scroungers. It is assumed that vigilant sparrows aware of danger make up between 10% and 20% of the population, and that the locations of these sparrows are updated according to the following Eq. (25):

$$X_{ij}^{(t+1)} = \begin{cases} X_{bj}^{(t)} + \beta\left(X_{ij}^{(t)} - X_{bj}^{(t)}\right) & f_i \neq f_g \\ X_{ij}^{(t)} + K\cdot\left(\dfrac{X_{ij}^{(t)} - X_{wj}^{(t)}}{|f_i - f_w| + e}\right) & f_i = f_g \end{cases} \tag{25}$$

where, $X_{bj}^{(t)}$ refers to the current best position in generation t, $X_{wj}^{(t)}$ refers to the current worst position in the t generation, f_i is the fitness value of the i-th sparrow. f_g is the global optimal fitness value, and f_w is the fitness value of the sparrow with the worst fitness in the current population. β is a random number that controls the step size, usually between 0 and 1, and is used to introduce randomness. K is a constant that controls the step size and is used to adjust the step size for position updates. e is a constant with a small value to prevent the denominator from being 0. When $f_i \neq f_g$, it means that the Vigilant is moving closer to the current global optimal position. When $f_i = f_g$, the Vigilant sets its location update direction to be farther away from the least well-suited location. the flowchart of Sparrow Search Algorithm (SSA) is shown in Fig. 8.

2.3.3. XGBoost hyperparameter optimization

The XGBoost classification algorithm involves multiple hyperparameters, which often require manual tuning (see Fig. 9). However, manually set hyperparameters are rarely optimal. A well-chosen combination of hyperparameters can significantly enhance the performance of the XGBoost classification algorithm, while poorly set hyperparameters can lead to underfitting or overfitting.

This paper presents a ship behavior recognition method based on the SSA-XGBoost algorithm. The method optimizes the hyperparameters of

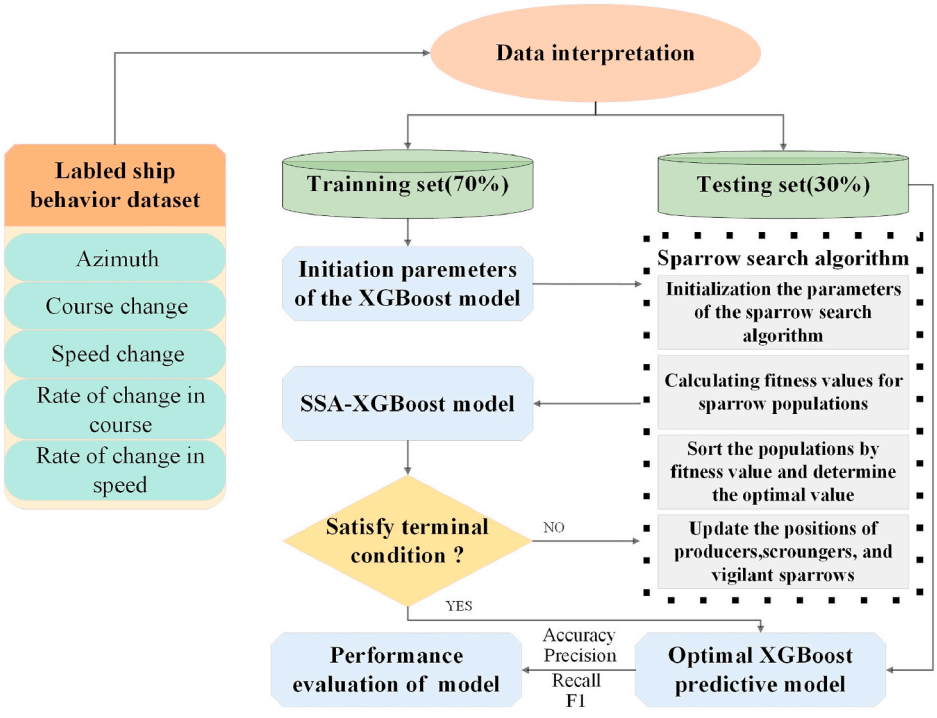


Fig. 9. The whole analysis process of the Sparrow Search Algorithm based on XGBoost.

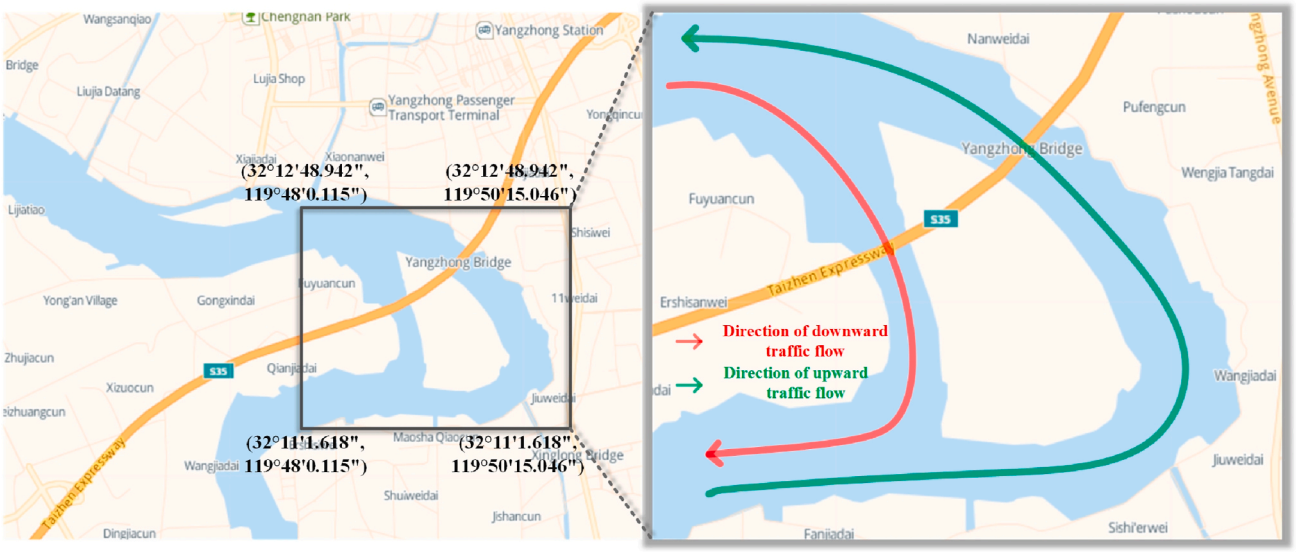


Fig. 10. Schematic of the study Watershed and direction of traffic flow.

XGBoost using the Sparrow Search Algorithm (SSA) to improve its performance. The specific implementation steps are summarized as follows.

- Determine Inputs and Outputs: Identify the inputs and outputs of the recognition model and create the training and test sets.
- Set Initial Parameters: Configure the initial parameters for XGBoost, including those involved in optimization, and set the population size, the maximum number of iterations, and the number of cross-validations for the Sparrow Search Algorithm.
- Calculate Fitness Values: Compute the fitness values of the sparrow population and rank these values to select the optimal value.
- Update Locations: Update the locations of the producer, scrounger, and vigilant sparrows.
- Check Termination Condition: Determine whether the number of iterations meets the termination condition. If not, repeat steps 3 and 4. If the termination condition is met, stop iterating, output the current optimal parameters, input the test set samples into the optimal model, and output the diagnostic results.

2.4. Evaluation index of methodological performance

In order to be able to objectively evaluate the performance of different integrated learning algorithms in solving the problem of ship behavior pattern recognition, this paper uses the following four metrics to compare the performance differences of different algorithms: *Accuracy*, *Precision*, *Recall*, F1 score, Receiver Operating Characteristic (*ROC*), Area Under ROC Curve(*AUC*). *Accuracy* is the ratio of the number of samples correctly classified by the classifier to the total number of samples. *Precision* is the proportion of all samples predicted to be in the positive category that are actually in the positive category. *Recall* is the proportion of samples that are correctly predicted to be in the positive category out of all samples that are actually in the positive category. F1 score is the reconciled average of precision and recall and is used as a combined measure of classifier performance. In the *ROC* curve, the horizontal coordinate represents the percentage of the number of samples that were correctly predicted as positive out of all the actual positive data samples. The vertical coordinate represents the percentage of the number of positive samples that were incorrectly classified as negative out of all the data samples that were actually negative. The *AUC* value represents the area below the *ROC* curve and takes values in the range [0,1]. The measure of a better classifier is the closer the *AUC* is to 1, i.e., the larger the *AUC* value, the better the model.

The formulas for the four evaluation indicators are shown in Eq. (26)–(29).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{26}$$

$$Precision = \frac{TP}{TP + FP} \tag{27}$$

$$Recall = \frac{TP}{TP + FN} \tag{28}$$

$$\text{F1 score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{29}$$

where, True Positive (TP) is the number of samples that are true positive and correctly classified as such. True Negative (TN) is the number of samples that are true negative and correctly classified as negative. False Positive (FP) is the number of samples that are true negative categories but are misclassified as positive. False Negative (FN) is the number of samples that are true positive categories but are misclassified as negative.

It is worth explaining that the ship behavior pattern recognition studied in this paper belongs to the multiclassification problem. Therefore, when calculating precision and recall, it is necessary to calculate and macro average (MA) for all categories; when calculating F1 score, it is necessary to calculate the precision and recall corresponding to all categories, and then calculate the F1 score corresponding to each category and macro average. The formulas for the three are shown in Eq. (30)–(32),

$$P_{macro} = \frac{1}{k}\sum_{i=1}^{k} P_i \tag{30}$$

$$R_{macro} = \frac{1}{k}\sum_{i=1}^{k} R_i \tag{31}$$

$$F1_{macro} = \frac{1}{k}\sum_{i=1}^{k} F1_i \tag{32}$$

where, P_{macro}, R_{macro} and $F1_{macro}$ are the precision rate, the recall rate and the macro-mean of the F1 score, respectively; k is the number of categories of the sample data; P_i is the precision rate of the i category; R_i is the recall rate of the i category; and $F1_i$ is the F1 score of the i category.

Fig. 11. Schematic diagram of ship trajectory before and after AIS data processing.

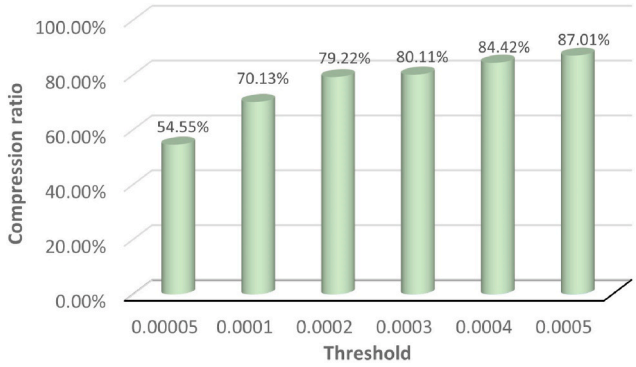


Fig. 12. Compression ratio of DP algorithm with different thresholds.

3. Case studies and experiments

This section outlines the methodology applied to the case study area and AIS data. Initially, the pre-processing steps for the AIS data are detailed. Subsequently, the processed data are utilized for ship sub-trajectory extraction, behavior coding, and dimensionality reduction and clustering of ship behaviors. Based on this, the spatial distribution characteristics and behavioral features of the ships are analyzed. Finally, the combined dataset, comprising ship behavior coding and clustering results, is input into the SSA-XGBoost algorithm for ship behavior identification. The performance of different algorithms in recognizing ship behavior patterns is compared and analyzed.

3.1. Experimental waters and data processing

The waters studied in this paper are located in the middle and lower reaches of the Yangtze River near the Yangzhong Bridge in Yangzhong City, Jiangsu Province, as shown in Fig. 10, where the area selected by the red rectangle is the case study area, and the specific latitude and longitude coordinates of the four vertices of the red rectangle are (119°48′0.1152″, 32°11′1.6188″), (119°48′0.1152″, 32°12′48.942″), (119°50′15.0468″, 32°11′1.6188″), (119°50′15.0468″, 32°12′48.942″).

The study area contains Yangzhong Bridge, Xisha Island and the navigation section is more curved, compared with the straight river section, the navigation behavior of ships in the study area changes more frequently. Therefore, this paper chooses this water as the research water, analyzes the ship navigation behavior in this water with practical significance and practical application value, can give the first time to sail into this water or unfamiliar with this water to provide navigation advice to the ship driver, and at the same time to assist the maritime authorities to develop regulatory programs.

In order to ensure the validity and reliability of the subsequent model data use, the ship AIS data collected in the case study area from November 1, 2021 to December 31, 2021 is now processed. The raw AIS data collected are now plotted as ship trajectory map and data processing is done. The results of data processing are shown in Fig. 11.

The last step of AIS data preprocessing is to extract the feature points of the ship trajectory, and the DP algorithm is introduced in section 2.1 of this chapter, and the parameter that affects the output results of this algorithm is mainly the threshold value. In order to find the threshold value suitable for the needs of the current study, the compression rate and output results of the DP algorithm under different threshold values are compared, as shown in Figs. 12 and 13.

As can be seen from Fig. 12, when the threshold is raised from 0.00005 to 0.0001, the trajectory compression rate increase is 15.58%; when the threshold is raised from 0.0001 to 0.0002, the trajectory compression rate increase is 9.09%. The compression rate increase brought about by these two threshold elevations is large, but the compression rate increase brought about by the subsequent threshold elevations decreases significantly, indicating that the trajectory compression rate begins to converge when the threshold is 0.0002. At this point, if the threshold continues to be raised, not only is the compression rate increase small, but the ship trajectory will also be distorted.

As can be seen in Fig. 13, when the threshold value is greater than or equal to 0.0003, the trajectory begins to appear distortion, and most of the trajectories in the southern part of the study waters are compressed into simple straight line segments, and a few of the trajectories appear to cross the land, losing the original trajectory characteristics. It shows that the threshold is set too high at this time, resulting in serious distortion of the compressed ship trajectory. By analyzing the above experimental results, this paper chooses the output results with a threshold value of 0.0002 as the data samples for subsequent research.

3.2. Clustering results of ship behavior

3.2.1. Ship behavior coding and dimensionality reduction

In this paper, we use preprocessed ship navigation data as a data source with the aim of extracting ship sub-trajectories and identifying the corresponding ship behaviors. Ship behavior usually refers to the dynamic activities and operation patterns exhibited by a ship during navigation. The analysis of subdivided trajectory segments can reveal how ships maneuver and navigate under different environmental and operational conditions. These behaviors may include steering, accelerating, decelerating, etc., and they are critical for navigation safety and

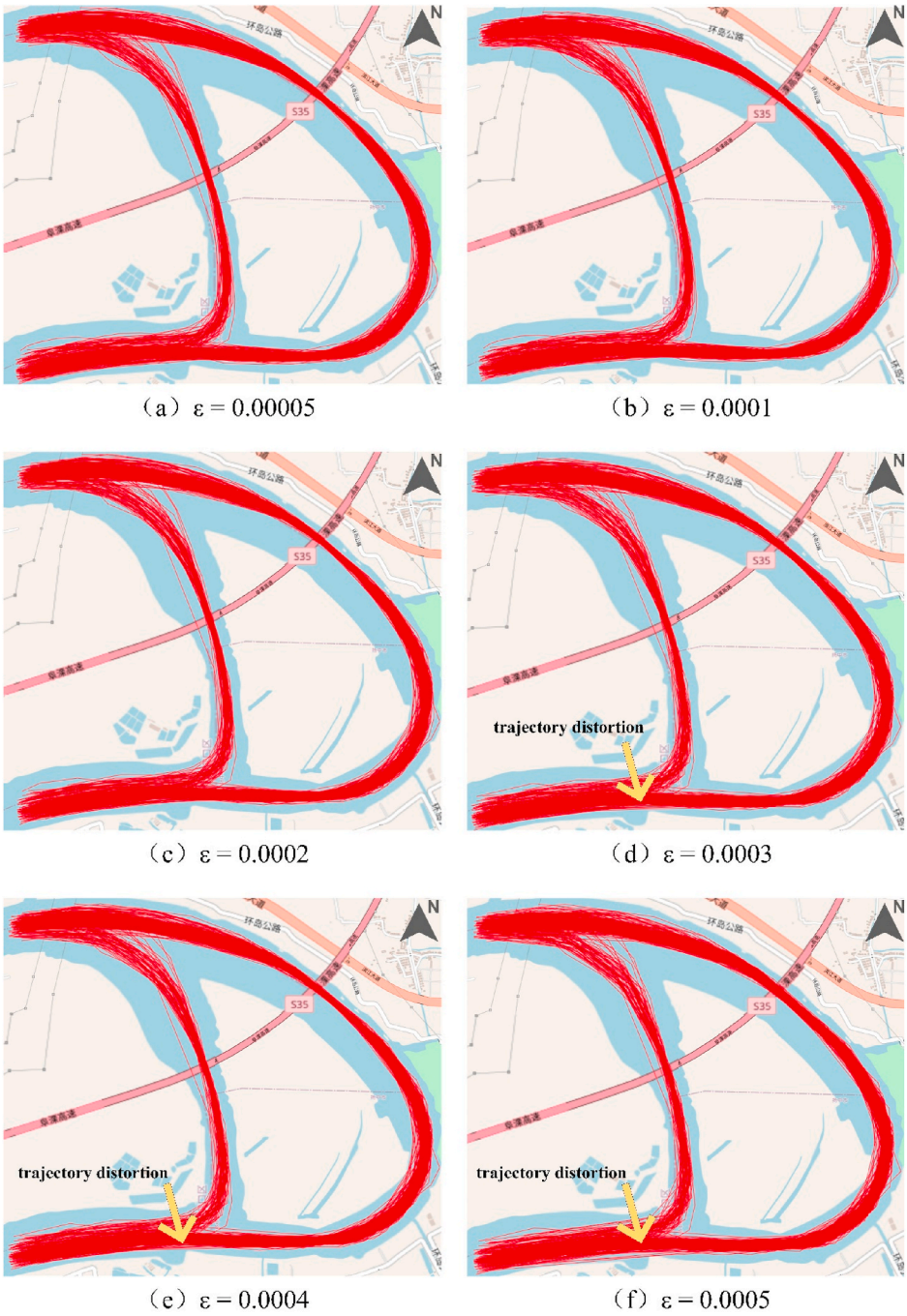


Fig. 13. Output of DP algorithm with different thresholds.

traffic management. However, since the original AIS data only contains latitude and longitude, speed over ground, course over ground, and time, which are relatively primitive information, it is difficult to describe the ship behavior, therefore, it is necessary to encode them, and the encoding includes the Azimuth, Δ*COG*, Δ*SOG*, *RC*, *RS*.

Changes in Azimuth can indicate a ship's steering maneuvers. By analyzing the fluctuation of the azimuth angle, it is possible to identify the steering behavior of a ship during a certain period of time, such as left turn, right turn and whether it is sailing in a straight line or not. Δ *COG* refers to the difference between a ship's turn from one heading to another. Large changes in heading may indicate that a vessel is engaged in behaviors such as avoiding, turning around, or turning in the channel. Frequent small changes may point to maneuvers that require precise maneuvering, such as navigating a narrow channel. *RC* is the rate of change of course per unit time. A high *RC* indicates that the ship is steering rapidly, which often occurs when avoiding obstacles or traversing complex waters. By monitoring the *RC*, it is possible to identify behaviors such as sharp turns, sustained turns, and traveling in a straight line at a constant speed. Δ*SOG* reflect the increase or decrease in a ship's speed over a period of time. Vessel acceleration is often associated with behaviors such as leaving port, avoiding danger, and changing course, while deceleration occurs mostly when berthing, anchoring, or preparing for a change of direction. *RS* indicates the rate of change of speed per unit of time. A high *RS* may indicate an emergency acceleration or deceleration maneuver. A steady *RS* usually indicates that the ship is engaged in a planned acceleration or deceleration of navigation rather than an emergency maneuver. By combining these metrics, we are able to construct a multi-dimensional ship behavior

Table 1
Example of ship behavior after coding.

MMSI	Azimuth	ΔCOG	ΔSOG	Rate of change in coures (RC)	Rate of change in speed(RS)
413973083	89.30174	12.67397	−0.22777	0.14106	−0.00253
413973083	100.66589	14.80551	−0.06465	0.12359	−0.00054
413973083	115.34473	11.62729	−0.00631	0.12941	−0.00007
413973083	125.14700	19.94029	0.02186	0.13316	0.00015
413973083	149.92980	16.90236	0.25968	0.18812	0.00289
413973083	158.89132	0.40843	0.64737	0.00227	0.00360
413973083	170.04237	15.97359	−0.63442	0.06667	−0.00265
413973083	196.75719	35.32902	−1.21015	0.23592	−0.00808
413973083	217.63242	38.83105	−0.41981	0.32413	−0.00350

model to better analyze and predict the actions a ship may take in different scenarios.

Taking the ship with MMSI No. 413973083 as an example, Table 1 shows the 9 ship behaviors coded for this ship, where each row of data describes one ship behavior, from row 1 to row 9, in the chronological order of the sub-trajectory starting point. From Table 1 and it can be seen that the ship azimuth gradually adjusted from 89.3° to 217.6° during the observed time period, and the course changes were all positive, indicating that it was making a right turn and accompanied by a change in speed. Through the above analysis, it can be seen that the coded ship sub-trajectories can better describe the ship behavior. Although coding can assist in analyzing the behavior of a single ship, it is difficult to mine the behavioral patterns of groups of ships, so it is necessary to code all the ship behaviors in the dataset.

After completing the coding process for the ship behaviors in the dataset, they need to be downscaled to project the coded high-dimensional ship behaviors to the three-dimensional space. In this paper, the UMAP algorithm is used to downscale the coded 4647 ship behaviors. The output of the UMAP algorithm is shown in Fig. 14, where each data point corresponds to a ship behavior and the three axes correspond to the three features after downsizing, respectively. As can be seen from Fig. 14, the use of the UMAP algorithm not only maintains the differences between individual ship behaviors, but also maintains the global structure of the dataset well.

3.2.2. *Clustering results and analysis*

In this paper, we use the spectral clustering algorithm to cluster the ship behaviors after dimensionality reduction and use the CH score to evaluate the advantages and disadvantages of the clustering results under different numbers of clusters. Fig. 15 shows the CH scores for 12 different numbers of clustering results, and from Fig. 15, it can be seen that the CH score is highest when the number of clusters is 9. Therefore, this paper selects the output with the number of clusters of 9 as the object of analysis, and Fig. 16 shows the distribution of this output in the three-dimensional space.

According to the clustering results of ship behaviors in Fig. 16, the ship trajectory map is drawn. Fig. 17 shows the overall display of the clustering results, and Fig. 18 shows the distribution pattern of different categories of ship behaviors on the map. Next, the kernel density is estimated for the speed change, course change, and azimuth in each category of ship behavior, and the processed results are shown in Fig. 19.

In response to the observation and analysis of ship behavior in the case study area, the following key regularities can be drawn from the observations in Figs. 18 and 19:

Although the geographical distribution of Cluster 1 and Cluster 4 ship behaviors is similar, and the peak speed change of both clusters is 0, the azimuth and course change values differ. For azimuth, Cluster 1 behaviors are primarily distributed at 85°, while Cluster 4 behaviors are

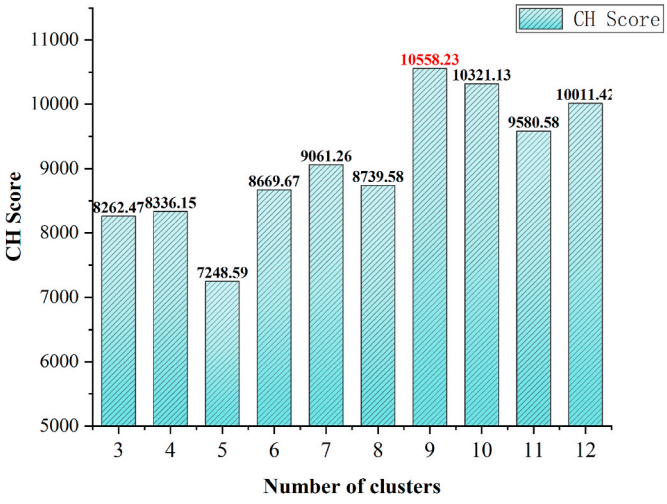


Fig. 15. CH scores for 12 different numbers of clustering results.

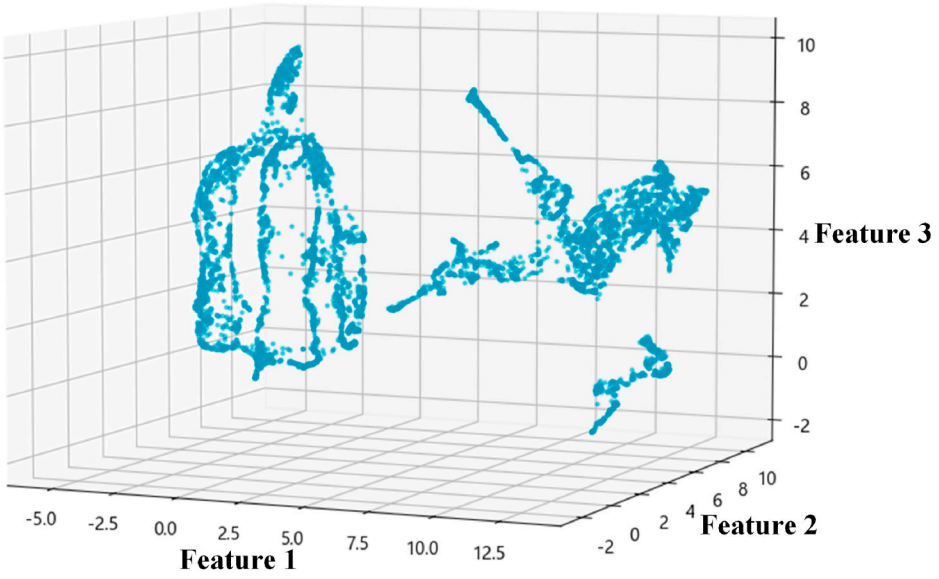


Fig. 14. UMAP downscaling results.

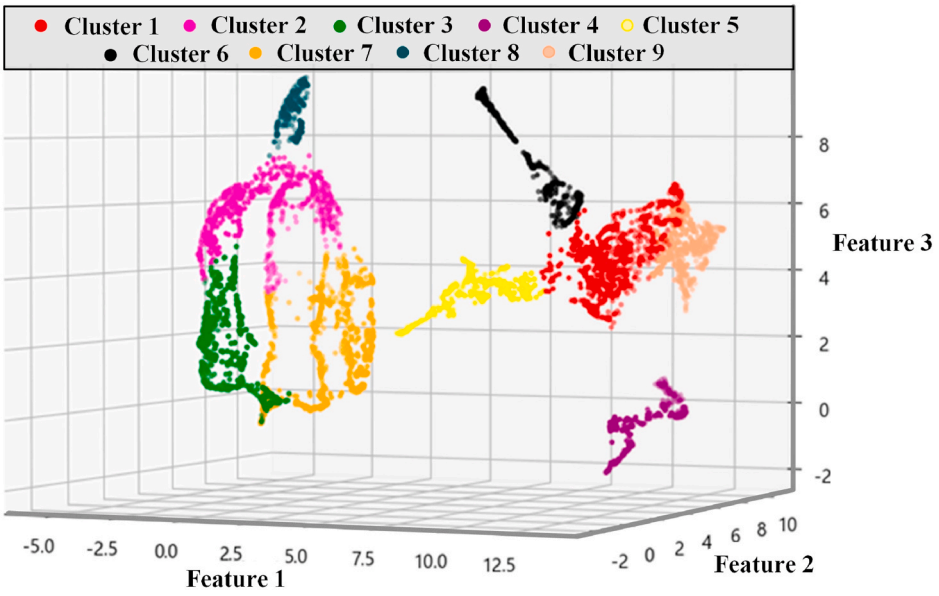


Fig. 16. Output of spectral clustering algorithm with number of clusters 9.

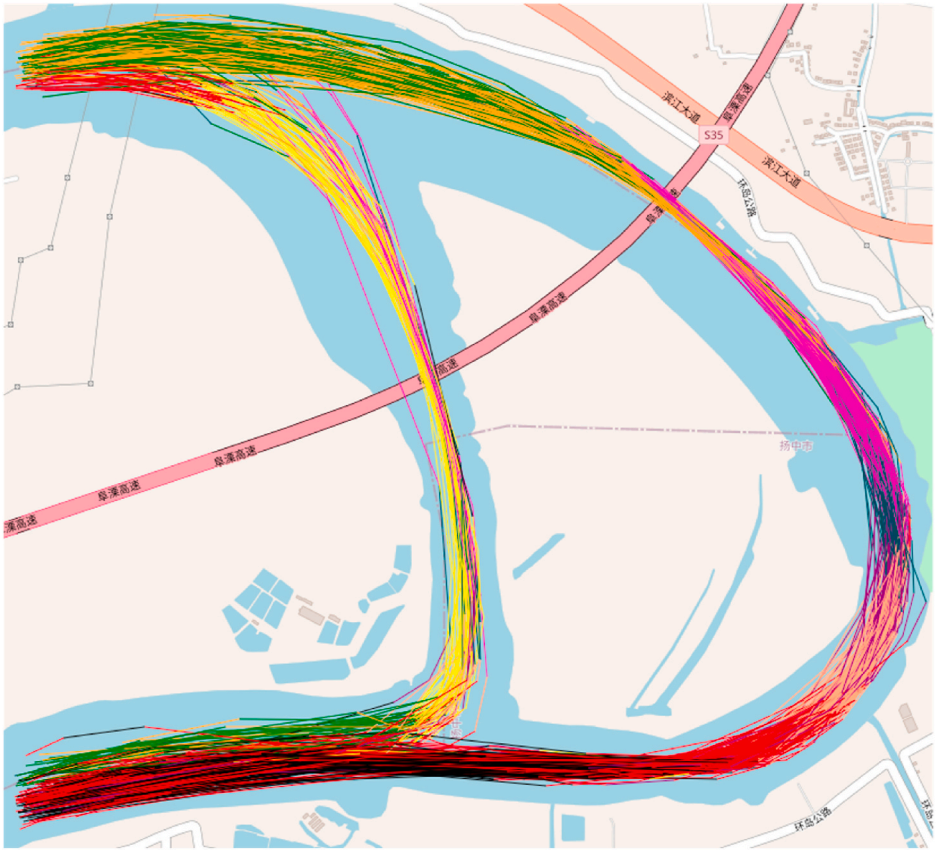


Fig. 17. Clustering results.

mainly at 10°. Regarding course change values, the peak for Cluster 1 is around −20, indicating a left steering attitude, whereas the peak for Cluster 4 is around 25, indicating right steering.

Although the geographical distribution of Cluster 2 and Cluster 8 ship behaviors is similar, their azimuth, course change, and speed change values show different distributions. Cluster 2 behaviors have an azimuth primarily at 320°, while Cluster 8 behaviors are mainly at 350°. The course change peaks for Cluster 2 and Cluster 8 are −12 and 25, respectively, meaning Cluster 2 behaviors are mostly accompanied by a small left turn, while Cluster 8 behaviors involve a right turn. For speed change, the peaks for Cluster 2 and Cluster 8 are 0 and -0.2, respectively, indicating that Cluster 2 behaviors involve relatively small speed changes, while Cluster 8 behaviors include slight deceleration.

The geographical distribution of Cluster 3 and Cluster 7 ship

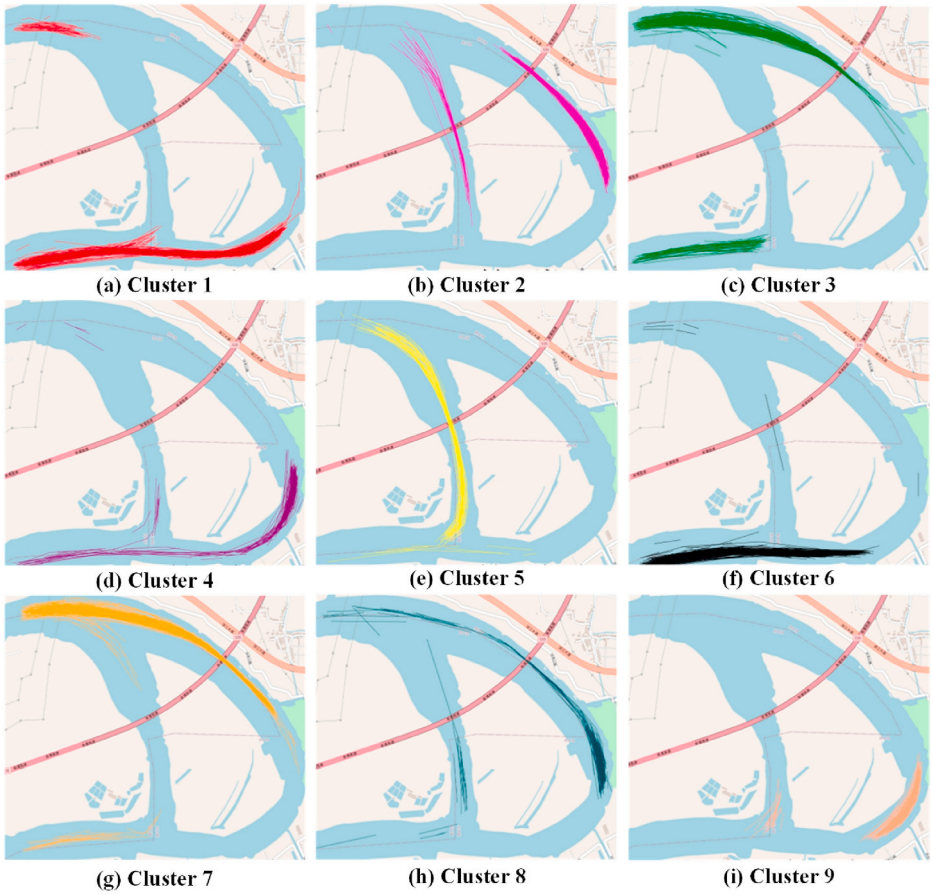


Fig. 18. Distribution of different categories of ship behavior.

behaviors is relatively similar, with the peak course change for both clusters at around −10. However, the azimuth and speed change values differ. Cluster 3 behaviors have an azimuth primarily at 280°, while Cluster 7 behaviors are mainly at 315°. The speed change peaks for Cluster 3 and Cluster 7 are 0.2 and −0.2, respectively, indicating that Cluster 3 behaviors are accompanied by slight acceleration, while Cluster 7 behaviors involve slight deceleration.

Cluster 5 ship behavior is geographically concentrated in the west channel and sailing downstream, with an azimuth mainly distributed at 155°. The peak course change is around 10, indicating slight right-hand steering, and the peak speed change is 0, indicating relatively small speed changes.

Cluster 6 ship behavior is geographically concentrated in the southern part of the waters, with an azimuth primarily at 90°. The peak course change is 0, indicating relatively small changes in course, and the peak speed change is 0.3, indicating slight acceleration.

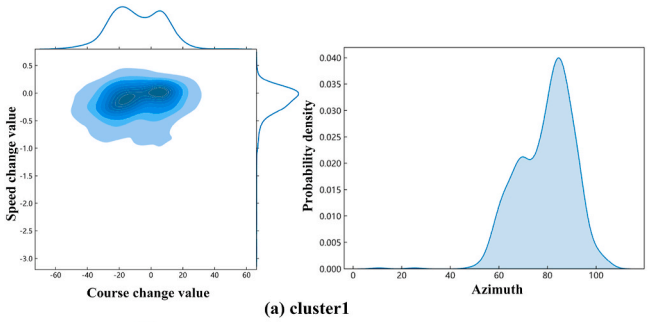
Cluster 9 ship behavior is geographically concentrated in the southwest part of the waters, with an azimuth primarily at 20°. The peak course change is around −20, indicating a larger left-turning behavior, and the peak speed change is 0, indicating relatively small speed changes.

The clustering results reveal the behavioral patterns of ship navigation in the case study area and align with the actual navigation situation. These analyses provide important references and support for ship pilots entering the waters for the first time, for the development of navigation aids, and for ship pilotage services. For ship pilots entering the waters for the first time, the results provide a visual reference of typical navigational paths and behavioral patterns, while each cluster represents a specific navigational strategy applicable to different navigational conditions. For navigational aids, the results can be used to optimize the arrangement of the navigational marking system, provide a data basis for the development of an intelligent navigation system, and help to predict the behavior of ships and provide real-time advice. The results can also be used to help develop more accurate pilotage plans and provide real-time advice for ship pilotage service. It helps to develop more accurate pilotage plans, and can also be used for pilot training to improve awareness and response to different navigational patterns. Although the clustering results provide a valuable reference for navigational patterns, multiple factors need to be considered in practical applications and used in conjunction with VTS and other navigation systems to ensure navigational safety.

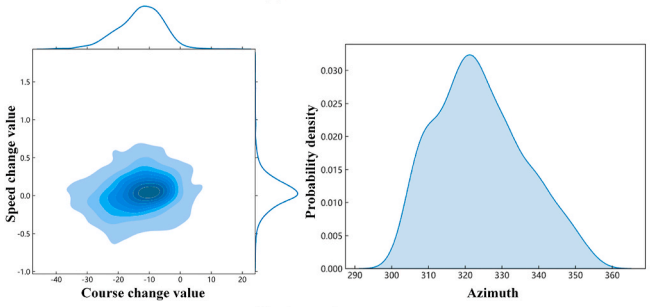
3.3. Recognition results and performance analysis

In this section, the clustering result with the highest CH score is used as the data source, and the Random Forest Algorithm, GBDT, XGBoost, and SSA-XGBoost are employed to identify ship behaviors. The performance differences of these algorithms in the problem of ship behavioral pattern recognition are compared and analyzed.

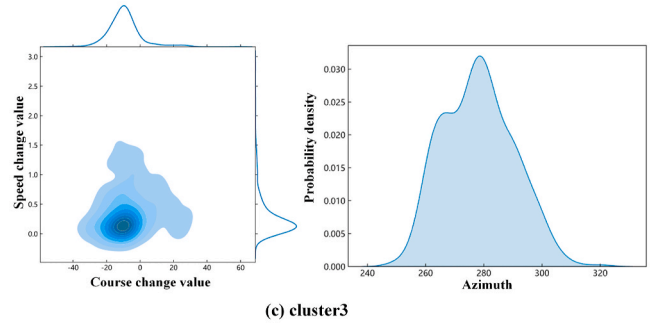
The hyperparameters of the XGBoost model that boost performance include the learning rate, the number of estimators (n_estimators), and the maximum depth (max_depth). The learning rate controls the magnitude of model parameter updates in each iteration; a smaller learning rate can make the model more stable but requires more iterations to reach the optimal solution. Therefore, the learning rate of the XGBoost model is selected using the Sparrow Search Algorithm in the range of 0.01–0.3. The number of estimators determines the number of rounds of model training; a larger number of iterations can make the model more accurate but may lead to overfitting. Consequently, the number of iterations' search range is set from 1 to 100. The depth of the tree controls the maximum depth of each tree; a greater depth can make the model better fit the training set but may also cause overfitting. Thus,



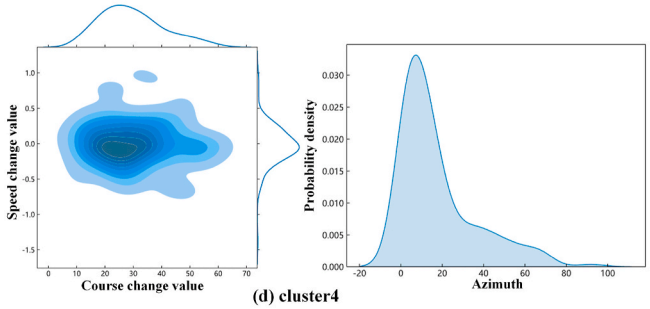
(a) cluster1



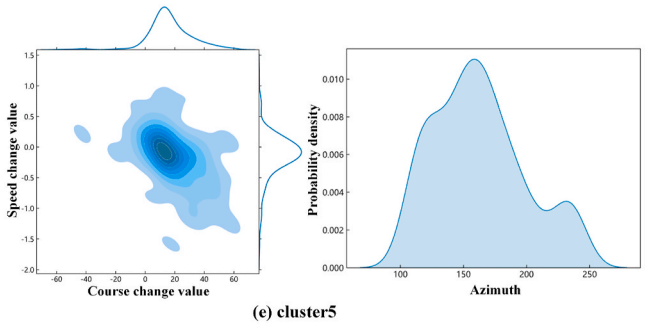
(b) cluster2



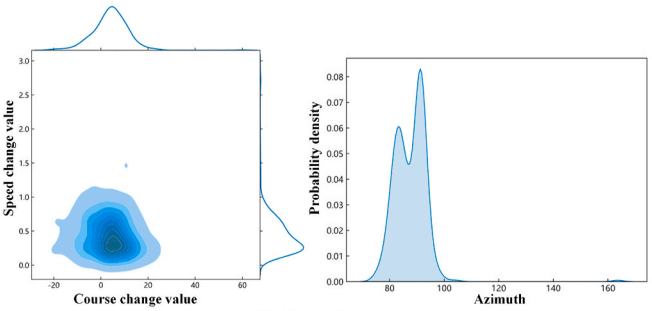
(c) cluster3



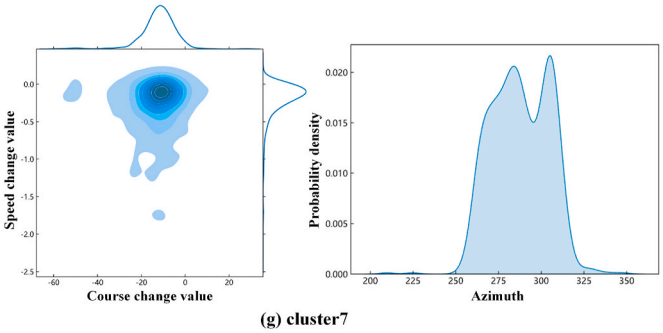
(d) cluster4



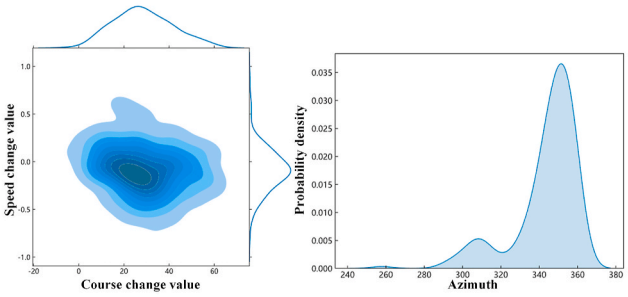
(e) cluster5



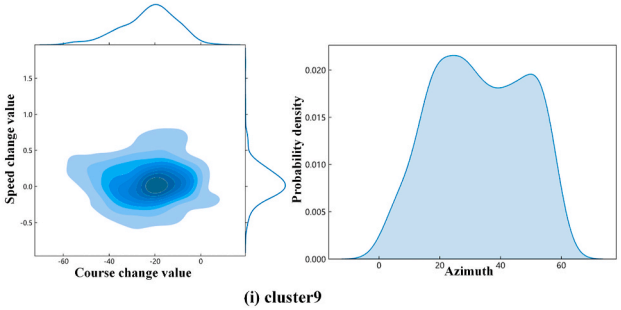
(f) cluster6



(g) cluster7



(h) cluster8



(i) cluster9

Fig. 19. Estimated probability densities of various types of ship behavioral characteristics.

Table 2
Parameter optimization settings.

Parameter name	Search Scope
learning rate	[0.01,0.3]
n_estimators	[1,100]
max_ depth	[3,15]

Table 3
Performance evaluation of different algorithms.

Algorithm name	Accuracy	Precision	Recall	F1 score
Random forest	0.8839	0.9161	0.8939	0.8839
GBDT	0.9039	0.9343	0.8751	0.8985
XGBoost	0.9425	0.9499	0.9384	0.9368
SSA-XGBoost	0.9728	0.9697	0.9743	0.9719

the depth of the tree is set to a range of 3–15 (Table 2).

By setting reasonable upper and lower limits for the boosting parameters, the search time and resource consumption caused by iteratively calculating the objective function values can be reduced, and search efficiency can be improved, avoiding XGBoost overfitting. Therefore, when optimizing XGBoost, it is particularly important to set reasonable upper and lower limits for the boosting parameters, which can improve search efficiency and enhance the generalization ability and performance of the SSA-XGBoost model. After optimization using the Sparrow Search Algorithm, the optimal parameter values obtained are a learning rate of 0.03, n_estimators of 20, and a max_depth of 12.

To intuitively compare the performance advantages and disadvantages of different algorithms in solving the ship behavior recognition problem, the recognition results of each algorithm are compared with the actual results using evaluation indices such as Accuracy, Precision, Recall, and F1 score, as shown in Table 3. According to the data in the table, the SSA-XGBoost algorithm achieves an accuracy of 97.28%, precision of 96.97%, recall of 97.43%, and F1 score of 97.19%. Compared to the other three algorithms, SSA-XGBoost has higher performance metrics and better performance in identifying ship behavior.

Different methods of dividing the training set and validation set can also impact model construction. To further verify the advantages of the SSA-XGBoost algorithm for the ship behavior problem, we conducted 10-fold cross-validation using the four algorithms. Cross-validation provides stable performance assessment through multiple training and testing. It can help identify cases where the model performs well on the training set but poorly on the unseen data, guiding the selection of the appropriate model complexity/hyperparameters and thus reducing the risk of overfitting. The results of 10 folder cross-validations of the four algorithms are shown in Fig. 20. The radar chart in Fig. 21 indicates that SSA-XGBoost performs the best in cross-validation compared to the other three algorithms. Meanwhile, the above comparative analysis shows that the ship behavior recognition model constructed by the SSA-XGBoost algorithm has better generalization capability, recognition accuracy, and overall performance. It indicates that the model performs consistently on different cross-validation sets, and that there is no significant overfitting or underfitting problem.

This Fig. 22 presents a comparative analysis of ROC curves for four machine learning models—Random Forest, GBDT, XGBoost, and SSA-XGBoost—evaluated across nine clusters. The results demonstrate a progressive improvement in model performance, with AUC values increasing from Random Forest to SSA-XGBoost. The SSA-XGBoost model stands out with the highest and most consistent AUC scores, reaching near-perfect values above 0.97, highlighting its superior predictive accuracy. This comparison underscores the effectiveness of

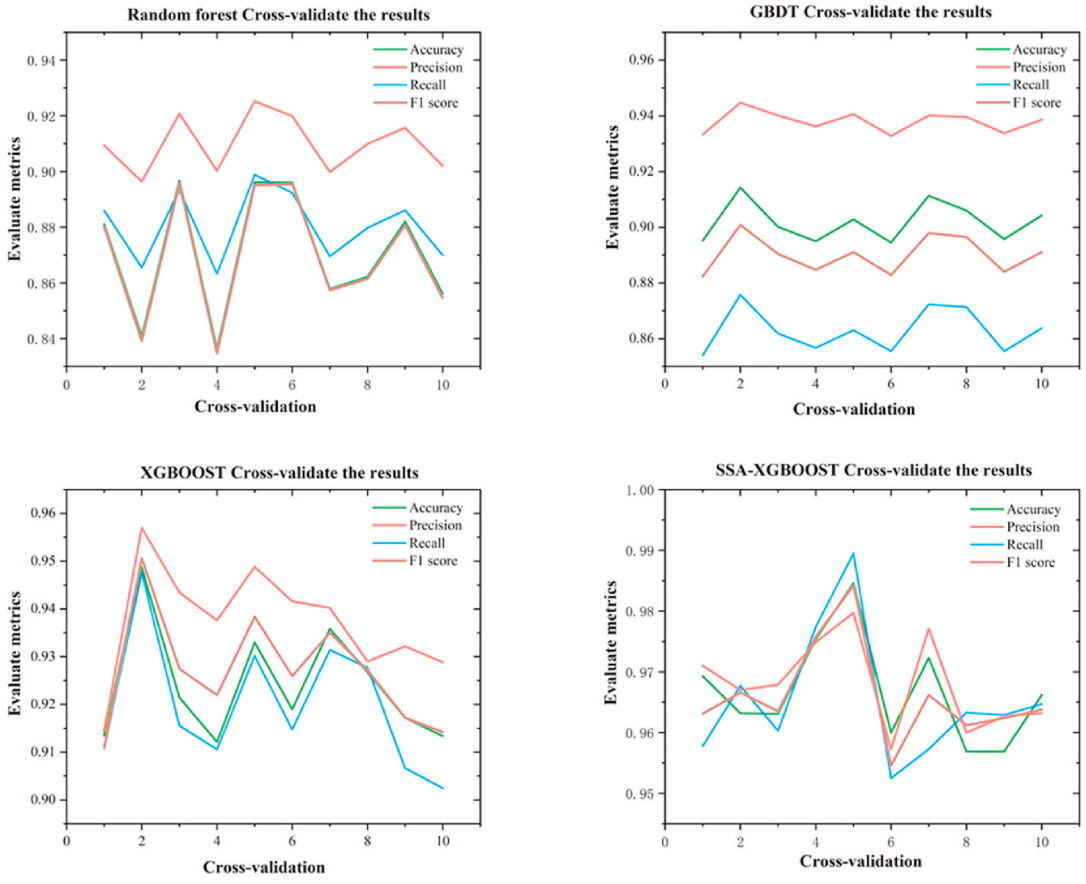


Fig. 20. Cross-validation results.

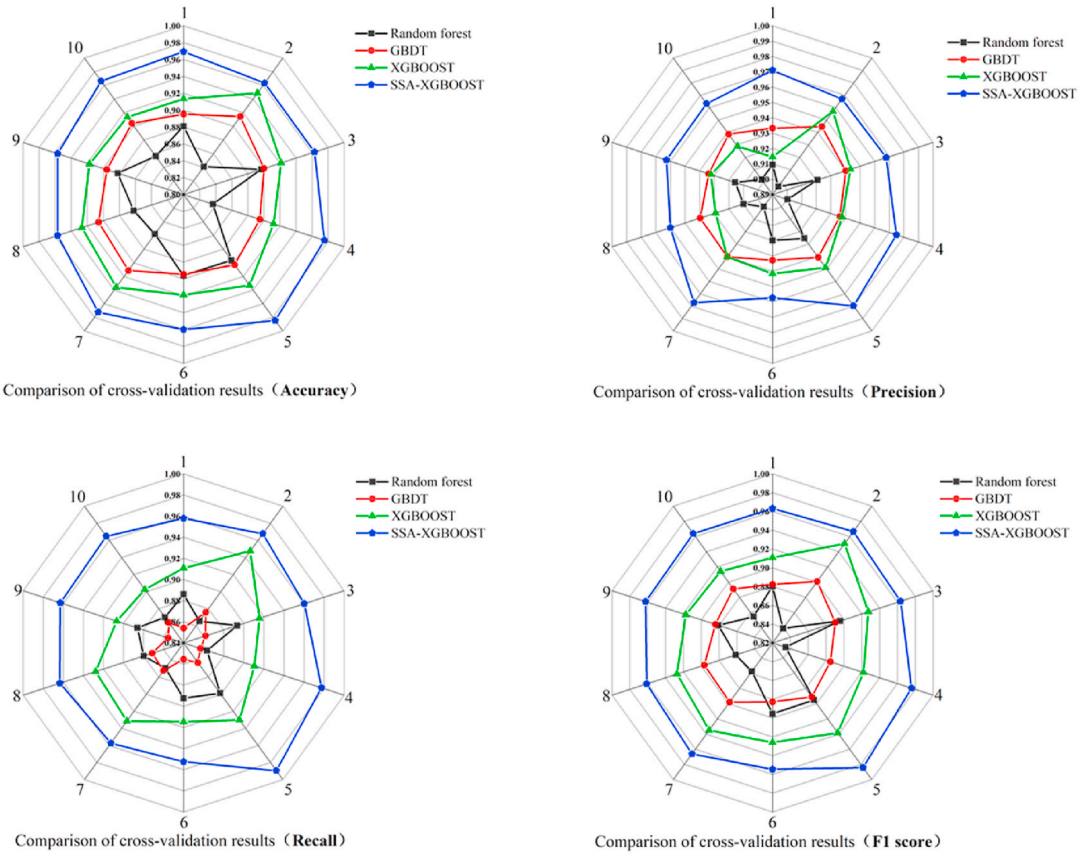


Fig. 21. Comparative analysis.

advanced optimization techniques, like SSA, in enhancing model performance across diverse data clusters.

4. Conclusion

In comparison to traditional Vessel Traffic Service (VTS) systems, machine learning-based methods significantly enhance the speed and accuracy of data processing while demonstrating adaptability to diverse navigational environments through dynamic learning. These capabilities empower maritime authorities to monitor potential risks more effectively and take timely actions, yielding notable improvements in both safety and operational efficiency. In this context, our study introduces a novel ship behavior pattern recognition method leveraging the SSA-XGBoost algorithm and AIS data.

The proposed method begins with the preprocessing of AIS data from the target study area, followed by encoding critical ship behavior features, such as azimuth, ΔCOG, ΔSOG, RC, and RS. We then apply the UMAP algorithm for dimensionality reduction and perform spectral clustering to facilitate an intuitive and comprehensive analysis of ship behavior patterns. The SSA-XGBoost algorithm subsequently maps the AIS data to these behaviors, ensuring efficient and accurate recognition.

The effectiveness of this approach was validated through a case study conducted near Yangzhong Bridge on the Yangtze River, covering data from November 1, 2021, to December 31, 2021. The results of spectral clustering closely corresponded to real-world ship behavior, and the SSA-XGBoost algorithm demonstrated superior performance compared to traditional models such as Random Forest, GBDT, and XGBoost. These outcomes provide a robust theoretical foundation for enhancing maritime traffic management, supporting the development of more effective strategies that improve traffic efficiency and reduce accident risks.

Compared to conventional VTS systems, the method proposed herein offers greater flexibility and precision in real-time data processing and the capture of multi-dimensional features. Traditional VTS systems often depend on manual monitoring and struggle to adapt swiftly to complex maritime conditions. In contrast, our machine learning-based approach enhances navigation safety and management efficiency by leveraging dynamic learning and data-driven decision-making. Moreover, this method excels at analyzing and recognizing patterns within high-dimensional datasets, identifying intricate ship behavior patterns that might be overlooked by human operators. By automating the recognition process, our system reduces reliance on human monitoring, yielding cost savings and more efficient resource utilization in traffic control centers.

Nevertheless, this study has certain limitations. Firstly, the clustering results may be influenced by external parameters such as wind, waves, and currents. If these factors are not adequately accounted for, the identified behavior patterns may lack accuracy or practical relevance. Future research should incorporate more comprehensive validation methods and consider additional environmental variables to ensure the robustness of clustering results under diverse conditions. Additionally, while the SSA-XGBoost algorithm demonstrates exceptional performance, its effectiveness depends on the quality of input data and appropriate hyperparameter configurations. In situations where data quality is suboptimal or hyperparameter settings are not well-tuned, the model's recognition capabilities may be compromised. Future work will focus on improving data processing robustness and exploring advanced machine learning techniques to optimize model performance across varying data quality scenarios in complex maritime environments.

CRediT authorship contribution statement

Quandang Ma: Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Sunrong Lian:** Writing –

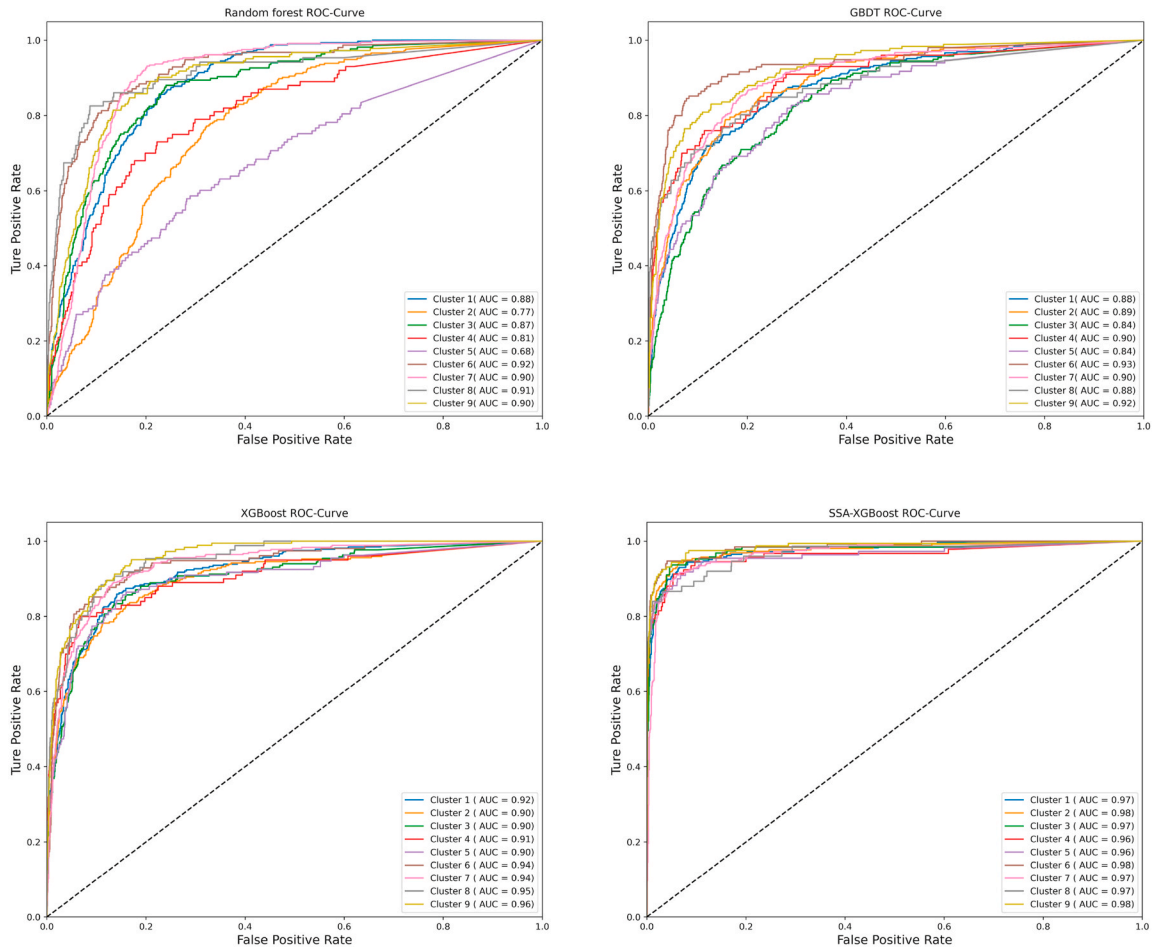


Fig. 22. AUC-ROC curve.

review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Dingze Zhang:** Writing – review & editing, Software, Methodology, Investigation. **Xiao Lang:** Writing – review & editing, Software, Methodology. **Hao Rong:** Writing – review & editing. **Wengang Mao:** Writing – review & editing, Visualization, Software. **Mingyang Zhang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation.

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.

Acknowledgments

This research is supported by the Natural Science Foundation of Hubei Province, China (2022CFB431).

References

Ahn, I., Kim, C., 2016. Face and hair region labeling using semi-supervised spectral clustering-based multiple segmentations. IEEE Trans. Multimed. 18 (7), 1414–1421.
Alessandrini, A., Mazzarella, F., Vespe, M., 2018. Estimated time of arrival using historical vessel tracking data. IEEE Trans. Intell. Transport. Syst. 20 (1), 7–15.
Bailey, N.J., Ellis, N., Sampson, H., 2008. Training and technology onboard ship: how seafarers learned to use the shipboard automatic identification system (AIS). Seafarers International Research Centre (SIRC). Cardiff University.
Carlson, L., Navalta, D., Nicolescu, M., Nicolescu, M., Woodward, G., 2021. Early classification of intent for maritime domains using multinomial hidden Markov models. Frontiers in Artificial Intelligence 4, 702153.
Castaldo, F., Palmieri, F.A., Bastani, V., Marcenaro, L., Regazzoni, C., 2014. Abnormal vessel behavior detection in port areas based on dynamic bayesian networks. In: 17th International Conference on Information Fusion (FUSION). IEEE, pp. 1–7.
Chen, T., Guestrin, C., 2016. Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, pp. 785–794.
Chen, Y.C., 2017. A tutorial on kernel density estimation and recent advances. Biostatistics & Epidemiology 1 (1), 161–187.
Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q., 2020. A survey on ensemble learning. Front. Comput. Sci. 14, 241–258.
Emmens, T., Amrit, C., Abdi, A., Ghosh, M., 2021. The promises and perils of Automatic Identification System data. Expert Syst. Appl. 178, 114975.
Fu, P., Wang, H., Liu, K., Hu, X., Zhang, H., 2017. Finding abnormal vessel trajectories using feature learning. IEEE Access 5, 7898–7909.
Fu, S., Gu, S., Zhang, Y., Zhang, M., Weng, J., 2023. Towards system-theoretic risk management for maritime transportation systems: a case study of the yangtze river estuary. Ocean Eng. 286, 115637.
Greidanus, H., Alvarez, M., Eriksen, T., Gammieri, V., 2016. Completeness and accuracy of a wide-area maritime situational picture based on automatic ship reporting systems. J. Navig. 69 (1), 156–168.
Huang, Y., Zhang, Q., 2019. Identification of anomaly behavior of ships based on KNN and LOF combination algorithm. In: AIP Conference Proceedings, 2073. AIP Publishing. No. 1.
Kim, J.S., Jeong, J.S., 2015. Pattern recognition of ship navigational data using support vector machine. International Journal of Fuzzy Logic and Intelligent Systems 15 (4), 268–276.
Li, G., Fang, S., Ma, J., Cheng, J., 2020. Modeling merging acceleration and deceleration behavior based on gradient-boosting decision tree. J. Transport. Eng., Part A: Systems 146 (7), 05020005.
Li, H., Liu, J., Wu, K., Yang, Z., Liu, R.W., Xiong, N., 2018. Spatio-temporal vessel trajectory clustering based on data mapping and density. IEEE Access 6, 58939–58954.
Liang, M., Liu, R.W., Zhan, Y., Li, H., Zhu, F., Wang, F.Y., 2022. Fine-grained vessel traffic flow prediction with a spatio-temporal multigraph convolutional network. IEEE Trans. Intell. Transport. Syst. 23 (12), 23694–23707.

Lin, X., 2019. A road network traffic state identification method based on macroscopic fundamental diagram and spectral clustering and support vector machine. Math. Probl Eng. 2019.
Liu, Z., Gao, H., Zhang, M., Yan, R., Liu, J., 2023. A data mining method to extract traffic network for maritime transport management. Ocean Coast Manag. 239, 106622.
Liu, C., Kulkarni, K., Suominen, M., Kujala, P., Musharraf, M., 2024. On the data-driven investigation of factors affecting the need for icebreaker assistance in ice-covered waters. Cold Reg. Sci. Technol. 221, 104173.
Liu, C., Liu, J., Zhou, X., Zhao, Z., Wan, C., Liu, Z., 2020. AIS data-driven approach to estimate navigable capacity of busy waterways focusing on ships entering and leaving port. Ocean Eng. 218, 108215.
Liu, C., Musharraf, M., Li, F., Kujala, P., 2022. A data mining method for automatic identification and analysis of icebreaker assistance operation in ice-covered waters. Ocean Eng. 266, 112914.
Liu, D., Rong, H., Soares, C.G., 2023b. Ship** route modelling of AIS maritime traffic data at the approach to ports. Ocean Eng. 289, 115868.
Liu, J., Zhang, J., Yang, Z., Wan, C., Zhang, M., 2024. A novel data-driven method of ship collision risk evolution evaluation during real encounter situations. Reliab. Eng. Syst. Saf. 249, 110228.
Liu, K., Yuan, Z., Xin, X., Zhang, J., Wang, W., 2021. Conflict detection method based on dynamic ship domain model for visualization of collision risk Hot-Spots. Ocean Eng. 242, 110143.
Liu, Z., Zhang, B., Zhang, M., Wang, H., Fu, X., 2023a. A quantitative method for the analysis of ship collision risk using AIS data. Ocean Eng. 272, 113906.
Ma, M., Zhao, G., He, B., Li, Q., Dong, H., Wang, S., Wang, Z., 2021. XGBoost-based method for flash flood risk assessment. J. Hydrol. 598, 126382.
Ma, Q., Du, X., Liu, C., Jiang, Y., Liu, Z., Xiao, Z., Zhang, M., 2024. A hybrid deep learning method for the prediction of ship time headway using automatic identification system data. Eng. Appl. Artif. Intell. 133, 108172.
Ma, Q., Zhou, Y., Zhang, M., Peng, Q., Fu, S., Lyu, N., 2023. A method for optimizing maritime emergency resource allocation in inland waterways. Ocean Eng. 289, 116224.
McInnes, L., Healy, J., Melville, J., 2018. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.
Murray, B., Perera, L.P., 2022. Ship behavior prediction via trajectory extraction-based clustering for maritime situation awareness. J. Ocean Eng. Sci. 7 (1), 1–13.
Pallotta, G., Vespe, M., Bryan, K., 2013. Vessel pattern knowledge discovery from AIS data: a framework for anomaly detection and route prediction. Entropy 15 (6), 2218–2245.
Qu, Y., Lin, Z., Li, H., Zhang, X., 2019. Feature recognition of urban road traffic accidents based on GA-XGBoost in the context of big data. IEEE Access 7, 170106–170115.
Rong, H., Teixeira, A.P., Soares, C.G., 2022. Ship collision avoidance behaviour recognition and analysis based on AIS data. Ocean Eng. 245, 110479.
Rong, H., Teixeira, A.P., Soares, C.G., 2024. A framework for ship abnormal behaviour detection and classification using AIS data. Reliab. Eng. Syst. Saf. 247, 110105.
Sorzano, C.O.S., Vargas, J., Montano, A.P., 2014. A survey of dimensionality reduction techniques. arXiv preprint arXiv:1403.2877.
Svanberg, M., Santén, V., Hörteborn, A., Holm, H., Finnsgård, C., 2019. AIS in maritime research. Mar. Pol. 106, 103520.
Tan, X., Cui, Z., Cao, Z., Min, R., 2020. Ship detection via superpixel-random forest method in high-resolution SAR images. In: Communications, Signal Processing, and Systems: Proceedings of the 2018 CSPS Volume II: Signal Processing 7th. Springer, Singapore, pp. 702–707.
Tang, C., Chen, M., Zhao, J., Liu, T., Liu, K., Yan, H., Xiao, Y., 2021. A novel ship trajectory clustering method for finding overall and local features of ship trajectories. Ocean Eng. 241, 110108.
Tang, H., Wei, L., Yin, Y., Shen, H., Qi, Y., 2020. Detection of abnormal vessel behaviour based on probabilistic directed graph model. J. Navig. 73 (5), 1014–1035.
Verleysen, M., François, D., 2005. The curse of dimensionality in data mining and time series prediction. In: International Work-Conference on Artificial Neural Networks. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 758–770.
Verschuur, J., Koks, E.E., Hall, J.W., 2022. Ports' criticality in international trade and global supply-chains. Nat. Commun. 13 (1), 4351.
Von Luxburg, U., 2007. A tutorial on spectral clustering. Stat. Comput. 17, 395–416.
Wang, L., Chen, P., Chen, L., Mou, J., 2021. Ship AIS trajectory clustering: an HDBSCAN-based approach. J. Mar. Sci. Eng. 9 (6), 566.
Wang, W., Huang, L., Liu, K., Zhou, Y., Yuan, Z., Xin, X., Wu, X., 2024. Ship behavior pattern analysis based on multiship encounter detection. ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civ. Eng. 10 (1), 04023045.
Wei, Z., Gao, Y., Zhang, X., Li, X., Han, Z., 2024. Adaptive marine traffic behaviour pattern recognition based on multidimensional dynamic time warping and DBSCAN algorithm. Expert Syst. Appl. 238, 122229.
Xue, J., Shen, B., 2020. A novel swarm intelligence optimization approach: sparrow search algorithm. Systems science & control engineering 8 (1), 22–34.
Yang, D., Wu, L., Wang, S., Jia, H., Li, K.X., 2019. How big data enriches maritime research–a critical review of Automatic Identification System (AIS) data applications. Transport Rev. 39 (6), 755–773.
Yu, Q., Liu, K., Chang, C.H., Yang, Z., 2020. Realising advanced risk assessment of vessel traffic flows near offshore wind farms. Reliab. Eng. Syst. Saf. 203, 107086.
Zhang, L., Chen, P.F., Luo, Y., Mou, J.M., Soares, C.G., 2024a. Identifying collision avoidance behaviour in AIS data from a heavy traffic area. In: Advances in Maritime Technology and Engineering. CRC Press, pp. 175–183.
Zhang, M., Taimuri, G., Zhang, J., Zhang, D., Yan, X., Kujala, P., Hirdaris, S., 2024b. Systems driven intelligent decision support methods for ship collision and grounding prevention: present status, possible solutions, and challenges. Reliab. Eng. Syst. Saf. 253, 110489.
Zhang, M., Zhang, D., Fu, S., Kujala, P., Hirdaris, S., 2022a. A predictive analytics method for maritime traffic flow complexity estimation in inland waterways. Reliab. Eng. Syst. Saf. 220, 108317.
Zhang, Z., Huang, L., Peng, X., Wen, Y., Song, L., 2022b. Loitering behavior detection and classification of vessel movements based on trajectory shape and Convolutional Neural Networks. Ocean Eng. 258, 111852.
Zhao, L., Shi, G., 2019. A trajectory clustering method based on Douglas-Peucker compression and density for marine traffic pattern recognition. Ocean Eng. 172, 456–467.
Zhou, C., Wen, K., Zhao, J., Bian, Z., Lu, T., Ko Ko Latt, M., Wang, C., 2024. Ontology-based method for identifying abnormal ship behavior: a navigation rule perspective. J. Mar. Sci. Eng. 12 (6), 881.
Zhou, Y., Daamen, W., Vellinga, T., Hoogendoorn, S.P., 2023. Ship behavior during encounters in ports and waterways based on AIS data: from theoretical definitions to empirical findings. Ocean Eng. 272, 113879.