

Article

MESTR: A Multi-Task Enhanced Ship-Type Recognition Model Based on AIS

Nanyu Chen ¹, Luo Chen ^{1,2,*}, Xinxin Zhang ¹ and Ning Jing ¹

¹ College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China; chennanyu@nudt.edu.cn (N.C.); zxx@nudt.edu.cn (X.Z.); ningjing@nudt.edu.cn (N.J.)

² Key Laboratory of Natural Resources Monitoring and Supervision in Southern Hilly Region, Ministry of Natural Resources, Changsha 410007, China

* Correspondence: luochen@nudt.edu.cn

Abstract: With the rapid growth in maritime traffic, navigational safety has become a pressing concern. Some vessels deliberately manipulate their type information to evade regulatory oversight, either to circumvent legal sanctions or engage in illicit activities. Such practices not only undermine the accuracy of maritime supervision but also pose significant risks to maritime traffic management and safety. Therefore, accurately identifying vessel types is essential for effective maritime traffic regulation, combating maritime crimes, and ensuring safe maritime transportation. However, the existing methods fail to fully exploit the long-term sequential dependencies and intricate mobility patterns embedded in vessel trajectory data, leading to suboptimal identification accuracy and reliability. To address these limitations, we propose MESTR, a Multi-Task Enhanced Ship-Type Recognition model based on Automatic Identification System (AIS) data. MESTR leverages a Transformer-based deep learning framework with a motion-pattern-aware trajectory segment masking strategy. By jointly optimizing two learning tasks—trajectory segment masking prediction and ship-type prediction—MESTR effectively captures deep spatiotemporal features of various vessel types. This approach enables the accurate classification of six common vessel categories: tug, sailing, fishing, passenger, tanker, and cargo. Experimental evaluations on real-world maritime datasets demonstrate the effectiveness of MESTR, achieving an average accuracy improvement of 12.04% over the existing methods.

Keywords: AIS; ship-type recognition; multi-task learning; anomaly detection



Received: 13 March 2025

Revised: 30 March 2025

Accepted: 31 March 2025

Published: 3 April 2025

Citation: Chen, N.; Chen, L.; Zhang, X.; Jing, N. MESTR: A Multi-Task Enhanced Ship-Type Recognition Model Based on AIS. *J. Mar. Sci. Eng.* **2025**, *13*, 715. <https://doi.org/10.3390/jmse13040715>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous growth in global maritime trade, the volume of shipping traffic has significantly increased, leading to heightened concerns regarding maritime safety issues [1–3]. Some vessels evade regulations by falsely reporting or tampering with ship-type information to engage in illegal activities such as poaching, smuggling, and piracy. These actions severely impact maritime traffic management, international trade security, and the ecological environment [4]. Therefore, identifying the type of ship accurately is crucial for strengthening maritime supervision, maintaining maritime order, and preventing illegal activities [5]. To this end, some studies have employed image recognition or deep learning techniques to determine the true categories of ships [6–8]. However, image-based methods are costly in terms of obtaining data and are susceptible to meteorological conditions, lighting environments, and other factors, making it difficult to ensure stability. Moreover, as forgery techniques continue to evolve, some vessels disguise their appearance

characteristics through illegal painting or structural modifications, further reducing the discriminative ability of vision-based methods in practical applications and making them less effective against increasingly sophisticated ship-type forgery tactics.

In recent years, the widespread adoption of the Automatic Identification System (AIS) has to some extent standardized maritime traffic management and provided crucial data support for maritime supervision. However, the accuracy and reliability of ship attribute information, which primarily relies on manual reporting by shipowners, are challenging to ensure [9]. Some vessels, when committing illegal activities, tamper with key information such as ship type to conceal their true intentions. For example, as shown in Figure 1a, a fishing vessel falsely reports itself as a “Cargo” vessel to evade regulation and engages in illegal fishing activities in a no-fishing zone. Compared to the manually reported attributes that are easily tampered with, trajectory data are relatively difficult to forge [10], and the spatiotemporal features can reflect the true operational patterns and behavioral characteristics of ships. As shown in Figure 1, there are significant spatiotemporal differences in the trajectories of fishing, cargo, and passenger vessels. Therefore, some studies have attempted to identify ship types based on trajectory data to enhance the reliability of classification [11–18]. However, the existing deep learning methods have not fully exploited the long-term dependencies and complex motion patterns in trajectory data, limiting the model’s classification ability when identifying different types of ships.



Figure 1. Schematic diagram of the spatiotemporal characteristics of different ship trajectory types. The blue lines represent the trajectories, and the red areas indicate the fishing ban zones. To ensure clarity, we have simplified and abstracted the trajectory representation while preserving key motion patterns.

To address this, this paper proposes a multi-task enhanced ship-type recognition method aimed at fully utilizing the spatiotemporal features of trajectory data to improve the model’s generalization ability and identification accuracy. Specifically, the main contributions of this paper are as follows:

- We propose a multi-task enhanced ship-type recognition method based on the Transformer model, utilizing self-attention mechanisms to capture long-range dependency associations in trajectory sequences.
- To enhance the model’s deep feature extraction of complex ship movement patterns, we combine two training tasks, *trajectory segment mask prediction* and *ship-type prediction*, to further improve the model’s recognition capability.
- We propose a motion-pattern-aware segment masking strategy, which masks local trajectory segments with complex dynamic features to more effectively extract detailed spatiotemporal features of the trajectory, thereby improving the model’s adaptability and recognition ability for different trajectory patterns.

- We have tested on real datasets and compared with baseline methods to verify the effectiveness of MESTR.

The remainder of this paper is organized as follows: Section 2 introduces related work. Section 3 describes the proposed method, MESTR. Section 4 presents experiments conducted with MESTR. Section 5 provides a detailed discussion. Finally, Section 6 concludes the paper and offers insights on future work.

2. Related Work

Ship-type recognition plays a key role in maritime safety and traffic management. In recent years, the research in this area has centred around two data sources: visual images of ships and spatiotemporal trajectory data from the AIS.

2.1. Image-Based Ship-Type Recognition

Image-based ship classification uses data from optical cameras, infrared (IR) sensors, and synthetic aperture radar (SAR) images. For example, Huang et al. [6] proposed a multi-feature learning framework (MFL) that combines Gabor transform, MS-CLBP, Fisher vectors, and BOVW+SPM to simultaneously extract the global and local features of ship images for ship classification, optimizing the classification results through feature-level and decision-level fusion strategies. Leclerc et al. [7] adopted a transfer learning strategy, applying pre-trained deep convolutional neural networks (CNNs) such as Inception and ResNet to ship image classification. By fine-tuning on smaller-scale ship datasets, they achieved the recognition of different ship types. Bentes et al. [8] proposed a CNN-based ship classification method specifically for maritime targets in TerraSAR-X SAR images. The study constructed a complete SAR target detection and classification pipeline, first using the constant false alarm rate (CFAR) algorithm to detect targets and then using a CNN for classification. They also proposed a multi-input resolution CNN model that uses SAR images of different resolutions as input to extract richer features and improve classification accuracy. Shi et al. [19] proposed a multi-feature fusion convolutional neural network (ME-CNN) for ship classification, combining 2D discrete fractional Fourier transform (2D-DFrFT), Gabor filtering, and complete local binary patterns (CLBP) to enhance the extraction of ship edges, contours, local textures, and global rotation-invariant features. They used logarithmic opinion pooling (LOGP) for decision-level fusion, integrating the classification results of multiple CNN models to improve robustness and accuracy. However, although these methods perform well in visual recognition, they still have some obvious limitations. For example, the cost of data acquisition is high, especially for the collection and annotation of high-quality remote sensing images, which require significant resources. To mitigate the burden of manual labeling, many recent studies have adopted contrastive learning strategies to improve model performance with limited supervision. For instance, Chen et al. [20] proposed an asynchronous contrastive-learning-based method for effective fine-grained visual classification, addressing the highly “imbalanced fineness” and “imbalanced appearances” of ships among subclasses. Dong et al. [21] proposed a multiscale contrastive learning network (MSCL-Net) for ship classification, utilizing a channel spatial attention module (CSAM) to extract the most similar channel features and leveraging spatial similarity to enhance them, thereby overcoming the challenge of significant interclass similarity and intraclass difference. However, these vision-based methods remain susceptible to external factors such as adverse weather and varying lighting conditions, which can significantly degrade classification performance. Moreover, the complex and dynamic background of the marine environment—particularly in crowded ports or scenes with overlapping ships—can introduce substantial visual interference, further increasing the difficulty of accurate recognition.

2.2. AIS-Based Ship-Type Recognition

Unlike images, AIS data are spatiotemporal sequence data. The early research mainly relied on manual feature extraction, such as speed and course changes, and used traditional machine learning algorithms (e.g., logistic regression and decision trees) for classification. For example, Zhong et al. [11] used satellite AIS data and a random forest algorithm to classify cargo ships, tankers, and fishing vessels. Baeg et al. [12] manually designed 39 features, including trajectory shape features, geographic features, and ship appearance features, and proposed an ink feature based on sketch recognition to describe the overall trajectory morphology of ships, improving the classification accuracy. Using trajectory information converted from AIS data, the study employed random forest and decision tree algorithms to classify fishing vessels, passenger ships, tankers, and cargo ships, validating the effectiveness of ink features in ship trajectory classification. Huang et al. [13] proposed an AIS-data- and machine-learning-based ship classification method, further subdividing cargo ship categories into bulk carriers, container ships, general cargo ships, and vehicle carriers. The study used tree structure, proximity, and regression models for classification and optimized the model through feature selection methods. Zhou et al. [22] identified ship behavior patterns in ports through behavior clustering. They used K-means clustering to classify ship behaviors, extracted the main behavioral features (e.g., trajectory and ground speed), and built a classifier based on ship static features (e.g., ship length and ship width) to predict ship behavior categories. However, these feature-engineering-based methods rely on manually designed features, often requiring extensive domain knowledge to construct effective feature sets, limiting the model's adaptability and generalization ability. Additionally, manual feature extraction methods often struggle to capture complex nonlinear temporal patterns, constraining the classification accuracy.

In recent years, deep learning methods have been widely applied in ship-type classification tasks due to their powerful data-driven capabilities. Unlike traditional feature-engineering-based methods, deep learning can automatically learn features from massive data, reducing the reliance on manually designed features. Graph neural networks (GNNs), especially graph convolutional networks (GCNs), have shown great potential in modeling the spatial dependencies and structural relationships inherent in ship trajectory data. For example, Li et al. [23] proposed a graph-based classification method that transforms AIS trajectory data into graph-structured representations. By leveraging GCNs to aggregate features from neighboring nodes, their model effectively classifies different vessel types. Similarly, Ye et al. [24] developed a GCN-based approach that integrates trajectory sequences and dependency relations to capture vessel behavior patterns more comprehensively. These methods demonstrate strong performance in modeling local spatial relations; however, GCNs often struggle with very long time series as their receptive fields are typically limited to a few graph layers, making it difficult to capture long-range temporal dependencies. To better address temporal modeling, especially for sequential AIS data, recurrent neural networks (RNNs), including gated recurrent units (GRUs) and long short-term memory networks (LSTMs), have been extensively explored. These models are designed to capture dynamic behavioral changes over time, such as variations in speed, heading, and movement patterns. For example, Fuscà et al. [14] proposed an LSTM-based ship classification method that extracts dynamic features such as speed, acceleration, and motion direction changes from AIS data to identify different types of ships. By modeling the time series behavior of ships with LSTM, the method overcomes the impact of AIS data noise and sensor errors. However, due to the reliance on recurrent structures, LSTM suffers from gradient vanishing or explosion issues during training, limiting its effectiveness in modeling long-time-series data.

In summary, the current methods have not adequately extracted the long-range sequential dependencies and complex movement patterns implicit in trajectory data, making it difficult to cope with the increasingly complex maritime environment. In recent years, Transformer [25] and its improved models have gradually been applied to time-series analysis tasks and have shown promising prospects. However, the traditional Transformer architecture still faces issues such as high computational complexity and insufficient focus on key features when dealing with AIS trajectory data. To further enhance the model's temporal modeling capabilities and computational efficiency, this paper employs a multi-task learning approach combined with an adaptive masking strategy to achieve stable ship-type recognition.

3. Methods

3.1. Concept Definitions

Definition 1. *Spatiotemporal Trajectory.* A spatiotemporal trajectory containing n trajectory points is defined as $Tr_{1:n} = \{p_1, p_2, \dots, p_i, \dots, p_n \mid p_i = (x_i, t_i), n \in \mathbb{N}^*\}$. Here, p_i denotes the i -th trajectory point. The geospatial coordinate of p_i is represented by $x_i = (lat_i, lng_i)$, where lat_i and lng_i correspond to the latitude and longitude, respectively. The timestamp t_i indicates the time at which the i -th trajectory point was recorded, with the condition that $t_1 < t_2 < \dots < t_n$.

An example of a spatiotemporal trajectory is shown in Figure 2. It consists of five trajectory points, each represented by a pair of latitude and longitude coordinates along with a timestamp.

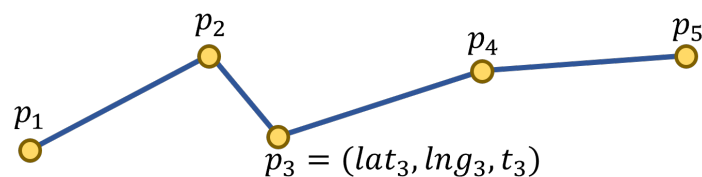


Figure 2. Illustration of spatiotemporal trajectory.

Definition 2. *Ship-Type Recognition Based on Spatiotemporal Trajectory.* Given a ship's spatiotemporal trajectory $Tr_{1:n}$, the objective of ship-type identification based on this trajectory is to accurately predict the ship's actual type. In this study, to ensure a balanced distribution of data, six common ship types are selected for analysis: "Tug", "Sailing", "Fishing", "Passenger", "Tanker", and "Cargo".

3.2. Model Structure

Transformer [25] is a deep learning model based on the attention mechanism, initially used for Natural Language Processing tasks. Its advantage lies in its ability to capture long-range dependencies in sequence data, enabling it to identify personalized features of different types of ships in the moving process. To enhance the model's ability to extract deep trajectory features, inspired by BERT [26], we designed a multi-task enhanced ship-type recognition model that includes a masked segment prediction task and a vessel-type prediction task. The overall architecture of the model is shown in Figure 3.

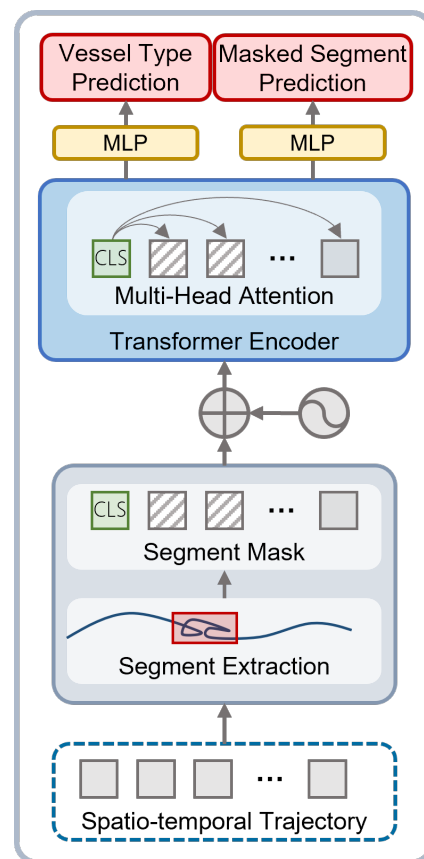


Figure 3. The MESTR architecture.

Unlike conventional ship-type prediction systems used in current shipboard or simulator-based applications, which typically rely on manually input metadata (e.g., vessel-reported AIS static information such as ship type, length, or draught) or rule-based trajectory interpretation, the MESTR framework introduces a fully data-driven approach. Traditional systems generally lack adaptive learning capabilities and are unable to capture hidden motion features or long-term behavioral dependencies. In contrast, MESTR leverages a multi-task Transformer architecture that jointly optimizes masked trajectory reconstruction and ship-type prediction, allowing the model to autonomously learn both global navigation patterns and fine-grained local movement cues from raw AIS trajectory data.

(1) Motion-Pattern-Aware Segment Masking

To enhance the model's ability to learn spatiotemporal features of ship trajectory data, we propose a motion-pattern-aware segment masking strategy. Unlike single-point masking methods, which tend to fit the spatial position of the trajectory through simple interpolation of adjacent trajectory points during the reconstruction process, thus failing to fully exploit the deep spatiotemporal patterns and motion characteristics of the trajectory, we adopt sub-trajectories as the masking units. This allows the model to learn richer motion patterns and dynamic changes, improving its predictive ability for missing trajectory data and enhancing its capability to capture ship behavior patterns.

During the navigation process, trajectory often exhibits complex maneuvering characteristics, which can provide important categorical information. In particular, high-curvature regions of the trajectory segment usually correspond to ship turning, circling, or other complex maneuvering behaviors. To accurately select these trajectory segments, we use a sliding window method to segment the trajectory, calculate the average curvature of each segment, and then set a masking threshold based on the third quartile (Q3) to filter

out high-complexity trajectory segments for masking. Given an input trajectory $Tr_{1:n}$, we employ a sliding window method to divide the trajectory into K segments. Each trajectory segment $Tr_{i:i+m-1}$ includes m consecutive trajectory points (segments with fewer than m points are truncated and not included in the calculation). The average curvature of each trajectory segment $Tr_{i:i+m-1}$ is shown in Equation (1).

$$\kappa = \frac{1}{m-2} \sum_{j=i+1}^{i+m-2} \frac{|v_{j-1} \times v_j|}{|v_{j-1}| |v_j| |p_{j-1} - p_j|} \quad (1)$$

Here, $v_{j-1} = x_j - x_{j-1}$ and $v_j = x_{j+1} - x_j$ represent the displacement vectors of adjacent trajectory points within the trajectory segment. $\times$ denotes the cross-product of vectors, and $|\cdot|$ represents the magnitude of a vector. A larger curvature value indicates that the trajectory segment contains more significant turning or circling operations, reflecting a more complex motion pattern. To reasonably select trajectory segments with higher maneuverability for masking, we base the curvature set $\mathcal{K} = \{\kappa_1, \kappa_2, \dots, \kappa_K\}$ and use the third quartile (Q3) as the masking threshold, i.e., $\kappa_{th} = Q3(\mathcal{K})$, selecting trajectory segments with curvature values greater than κ_{th} as candidate masking areas. Among these candidate trajectory segments, 50% are randomly selected for masking to enhance the model's focus on key trajectory patterns.

(2) Multi-Task Prediction

We append a special token, [CLS], to the masked trajectory sequence as a global representation of the trajectory sequence. By applying linear projection, we obtain feature embeddings and introduce trainable positional encodings to preserve the sequential information of trajectory points. This yields the input tensor $E_{mask} \in \mathbb{R}^{(n+1) \times d}$ for the encoder, where d represents the dimensionality of the model's hidden layer. The Transformer encoder utilizes a multi-head self-attention mechanism to compute the global relationships between trajectory points. The query (Q), key (K), and value (V) for each trajectory point are obtained through linear transformations, as shown in Equation (2), where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ are trainable weight parameters.

$$Q = W_Q E_{mask}, \quad K = W_K E_{mask}, \quad V = W_V E_{mask} \quad (2)$$

The attention scores are calculated as shown in Equation (3). Here, $\frac{1}{\sqrt{d_k}}$ serves as a scaling factor to ensure the stability of the gradients.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

The results from multiple attention heads are concatenated, as shown in Equation (4).

$$\text{MultiHead}(Q, K, V) = [\text{head}_1; \dots; \text{head}_h] W_C \quad (4)$$

The output of the multi-head self-attention is then passed through a feed-forward neural network (FFN) and subjected to residual connections and layer normalization to obtain the final output $E_L \in \mathbb{R}^{(n+1) \times d}$.

(3) Loss Function

In the vessel-type prediction task, we use the [CLS] vector E_{CLS} after encoding in $E_L \in \mathbb{R}^{(n+1) \times d}$ as the global trajectory representation. It is passed through the Softmax function to predict the ship-type distribution. For N ship categories, let the model's output

for the u -th category be $\hat{y}_u$ and the true category label be y_u . We calculate the loss for the vessel-type prediction task using the cross-entropy loss function, as shown in Equation (5):

$$L_{VTP} = - \sum_{u=1}^N y_u \log \hat{y}_u \quad (5)$$

In the masked segment prediction task, let the number of masked points be M . The model's output for the w -th masked position is $\hat{x}_w = (\hat{lat}_w, \hat{lon}_w)$, and the corresponding true trajectory point coordinates are $x_w = (lat_w, lon_w)$. To more accurately capture spatial features, we use a distance loss function with spatial awareness to calculate the loss for the masking prediction task, as shown in Equation (6):

$$L_{MSP} = \frac{1}{M} \sum_{w=1}^M \text{dis}_H(x_w, \hat{x}_w) \quad (6)$$

Here, $\text{dis}_H(\cdot, \cdot)$ represents the Haversine distance between two points, as shown in Equation (7). In this paper, $\alpha = 0.5(lat_w - \hat{lat}_w)$, $\beta = 0.5(lon_w - \hat{lon}_w)$, and Earth's radius $r_{eq} = 6378.137$ km.

$$\text{dis}_H(x_w, \hat{x}_w) = 2r_{eq} \times \arcsin(\sqrt{\sin^2 \alpha + \cos(\hat{lat}_w) \cos(lat_w) \sin^2 \beta}) \quad (7)$$

We combine L_{VTP} and L_{MSP} with weights ω to form the final loss function for MESTR, as shown in Equation (8):

$$L_{MESTR} = \omega \cdot L_{VTP} + (1 - \omega) \cdot L_{MSP} \quad (8)$$

In summary, MESTR leverages a Transformer encoder enhanced with a motion-pattern-aware segment masking strategy to better capture features from complex navigational behaviors. By employing a joint learning approach for trajectory segment reconstruction and ship-type classification, MESTR effectively models both global and local spatiotemporal patterns, leading to more accurate and robust vessel classification.

4. Experiments

4.1. Experimental Setup

4.1.1. Dataset and Metrics

We used open-source AIS data provided by the Danish Maritime Authority (DMA) (<https://www.dma.dk/safety-at-sea/navigational-information/ais-data>, (accessed on 1 March 2025)). The study area is a rectangular region ranging from (55.3762° N, 10.0852° E) to (57.89° N, 13.1133° E), with a diagonal distance of approximately 181.2049 nmi and an area of about 1.516×10^4 nmi² (based on WGS 84 (EPSG:7030)). The time range is from 1 July 2023 to 31 August 2023. The visualization of the dataset is shown in Figure 4.

Data preprocessing consists of four steps:

- Spatial filtering and outlier removal: Remove trajectory points outside the study area. Eliminate drift points in the trajectory based on a maximum speed threshold of 30 knots. Remove sampling points with missing or abnormal values in the fields "MMSI", "longitude", "latitude", "time", and "shiptype". Linear interpolation is then applied to fill in the missing segments, and the processed trajectories are subsequently grouped by MMSI.
- Trajectory segmentation: First, segment the trajectory at points where the time interval between consecutive records exceeds 1 h. Then, segment the trajectory with a maximum length of 10 h.

- Downsampling: Reduce the trajectory sampling rate to 10 min.
- Length filtering: Remove trajectories with lengths less than or equal to 3 h.

We randomly selected 5000 trajectories from the processed data and divided them into training, validation, and test sets in a 6:2:2 ratio. The overall situation is shown in Table 1.

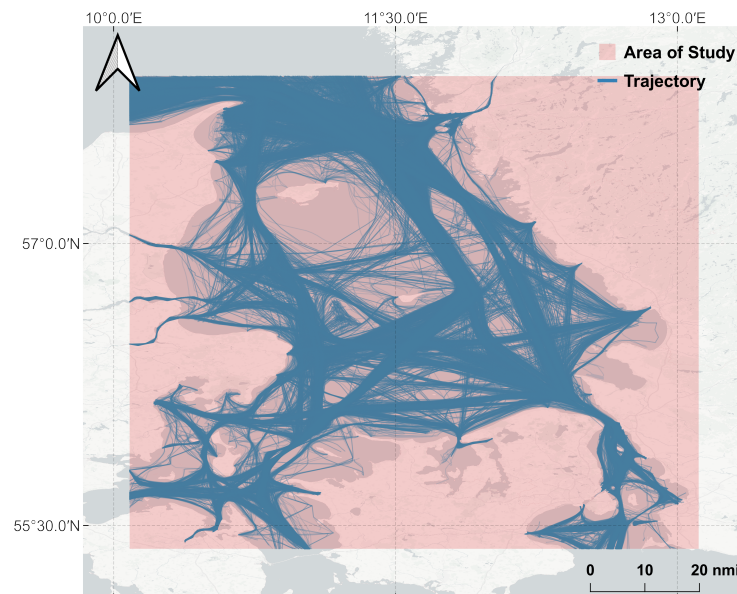


Figure 4. Visualization of the dataset.

Table 1. Dataset summary.

Dataset	Trajectory Points	Number of Trajectories	Number of Ships
Training	101,839	3000	752
Validation	32,091	1000	319
Testing	31,722	1000	362

We employed the commonly used accuracy metric to evaluate the overall recognition performance and used precision, recall, and F1-score to evaluate the model's recognition performance for each ship type. The calculation formulas are shown in Equations (9)–(11).

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (11)$$

Here, *TP* (true positive) represents the number of samples predicted as a certain category and actually belonging to that category. *FP* (false positive) represents the number of samples predicted as a certain category but actually belonging to other categories. *FN* (false negative) represents the number of samples actually belonging to a certain category but incorrectly predicted as other categories. In the ship-type recognition task, these metrics reflect the model's category recognition ability from different perspectives.

4.1.2. Baseline

To validate the effectiveness of the proposed method, we selected three common deep learning models as comparison methods: MLP, GRU, and Transformer, and made appropriate structural adjustments for trajectory data characteristics. Specifically,

- MLP [27]: A multi-layer perceptron (MLP) is used to encode trajectories and predict ship types. This method does not consider the temporal features of trajectories and relies solely on the static information of each trajectory point for classification. We employed a 3-layer MLP, with each layer consisting of 64 neurons, using ReLU as the activation function.
- vanilla GRU [28]: GRU is a variant of the recurrent neural network (RNN) that simplifies the structure of the LSTM (long short-term memory) network. It controls information updates and forgetting through gating mechanisms, improving computational efficiency and reducing model complexity. We used a 3-layer GRU, with each layer containing 64 units.
- vanilla Transformer [25]: The Transformer models global trajectory information through self-attention mechanisms, overcoming the difficulty of RNN structures in capturing long-distance dependencies. We employed an 8-layer Transformer encoder, with each layer including 8 attention heads, a hidden layer dimension set to 64, and GELU as the activation function.

All the above methods use the same data preprocessing methods and are tested under the same experimental conditions to ensure the fairness and comparability of the experimental results.

4.1.3. Implementation Details

In the local masking part, to fully leverage the Transformer encoder's capability in capturing long-range dependencies within vessel trajectories while simultaneously avoiding excessively large masking intervals—which may hinder the model's ability to learn effective movement patterns—we set the sliding window size to $m = 8$. We use a weight $\omega = 0.5$ to balance the multi-task loss. The dataset comprises six vessel categories: "Tug", "Sailing", "Fishing", "Passenger", "Tanker", and "Cargo", resulting in a vessel-type parameter of $N = 6$. We train MESTR via the AdamW optimizer [29]. To prevent overfitting, we employ an early stopping mechanism, which stops training if the validation loss does not improve after a given patience of 5. The remaining parameter settings are shown in Table 2.

The hyperparameters were selected based on a combination of prior work and empirical tuning. We first referred to commonly used settings in Transformer-based models to define a reasonable search space. Then, we conducted a grid search over key parameters, including the learning rate, dropout rate, hidden size, and number of layers. For example, the learning rate was searched in the range of $[1 \times 10^{-5}, 1 \times 10^{-3}]$, and dropout values from 0.0 to 0.3 were evaluated. The batch size and number of attention heads were chosen based on computational constraints and prior studies. The final settings were determined based on validation performance, aiming for a balance between model accuracy and training stability.

The training of MESTR was conducted on a single NVIDIA GeForce RTX 3090 GPU with 24 GB memory. The NVIDIA GeForce RTX 3090 GPU is manufactured by NVIDIA Corporation, which is located in Santa Clara, California, USA. The CUDA version was 12.1, and the operating system was Ubuntu 22.04. Thanks to the parallelism of the Transformer architecture, each epoch took an average of 4.2 min to complete. The peak GPU memory usage was approximately 16 GB. All experiments were implemented using PyTorch 2.3.0 and executed on a cloud server equipped with an Intel® Xeon® Platinum 8362 2.80 GHz

CPU and 45 GB of RAM. The Intel® Xeon® Platinum CPU is manufactured by Intel Corporation, which is located in Santa Clara, California, USA. The computational cost is moderate and feasible for most modern deep learning environments.

Table 2. Model hyperparameters.

Parameter	Value
Batch size	32
Epoch	100
Dropout	0.1
Hidden size	64
Hidden layer	8
Number of heads	8
Learning rate	1×10^{-4}

4.2. Result Analysis

As shown in Table 3, MESTR outperforms all comparison methods in overall classification performance, achieving an accuracy of 67.00%, which is 12.04% higher than the second-best method, Transformer. This result indicates that MESTR has stronger feature extraction capabilities in trajectory classification tasks, enabling it to more accurately learn the movement patterns of different ship types and effectively improve classification accuracy. Further analysis with Figure 5 shows that MESTR performs more stably in terms of precision, recall, and F1-score, maintaining good classification performance across different ship types. In particular, MESTR achieves a precision of 80.26% for the fishing category, far exceeding other methods, indicating its stronger ability to extract fine-grained features for targets with complex trajectory patterns. Additionally, MESTR shows superior performance in the passenger, sailing, and tug categories, demonstrating its ability to effectively model feature differences in different types of trajectory data and achieve more accurate classification.

Table 3. Accuracy comparison.

Method	MLP	GRU	Transformer	MESTR
Accuracy	43.20	52.70	59.80	67.00

Bold indicates the best performance.

In contrast, MLP performs poorly overall (43.20%), indicating that simple fully connected neural networks cannot effectively model the spatiotemporal dependencies of trajectories, making it difficult for the model to learn key features in trajectory sequences. GRU improves classification accuracy to some extent (52.70%) by modeling the temporal dynamics of trajectories through gating mechanisms, but it still has limitations in capturing complex trajectory patterns, leading to suboptimal classification performance for some categories. The Transformer model, which uses global attention mechanisms to effectively model long-range dependencies in trajectories, achieves an accuracy of 59.80% in the experiments. However, it still has some shortcomings in processing local trajectory features, resulting in less than ideal recognition accuracy for some categories.

To further explore the performance of different methods in predicting different ship types, we constructed confusion matrices for all methods and visualized the results. From Figure 6, we can observe that traditional methods struggle to accurately distinguish between tanker and cargo ships due to their similar trajectory characteristics during navigation. However, MESTR, through the mask prediction task, guides the model to reconstruct key local trajectory segments, enabling it to more effectively capture the local spatiotemporal

details of trajectories and reveal subtle differences in trajectory patterns between different ship types, ultimately improving the classification accuracy of tanker and cargo ships.

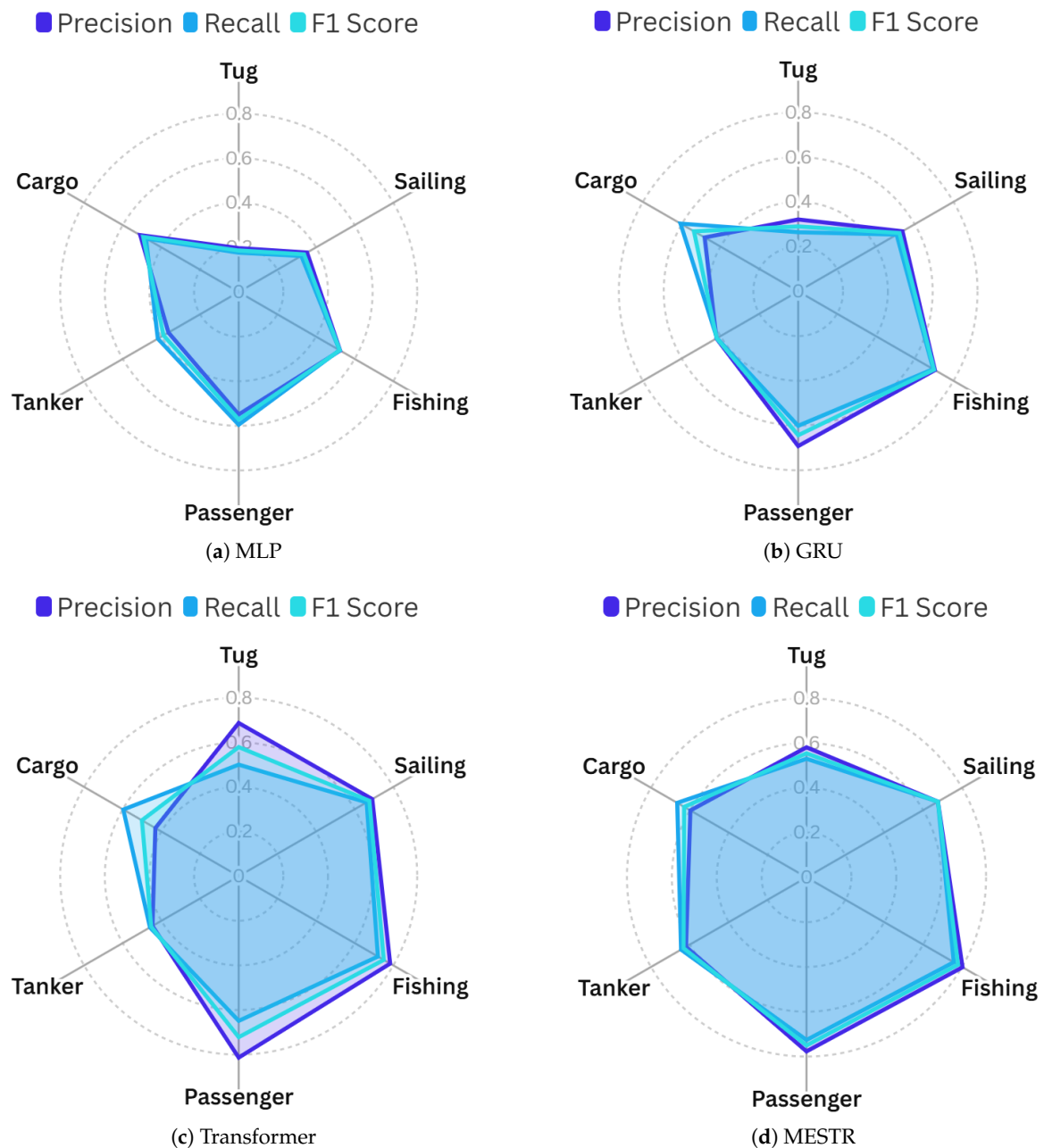


Figure 5. Precision, recall, and F1-score radar charts. The radial axis represents normalized score values ranging from 0 to 1 (unitless).

In contrast, MLP performs poorly in the experiments due to its inability to effectively model the temporal dependencies of trajectories. GRU relies on gated recurrent units for temporal modeling, but its local dependency is strong, which may lead to the attenuation of historical information when processing long-span trajectory information, making it difficult for the model to fully utilize global trajectory patterns to distinguish between tanker and cargo ships. Additionally, the vanilla Transformer, while superior to GRU in modeling long-range dependencies due to its global attention mechanism, has weaker attention to key local trajectory features, making it difficult to extract sufficient information for effective differentiation when faced with ship categories that have similar global spatiotemporal

distributions but slightly different local trajectory details. Therefore, these two methods perform worse than MESTR in the classification task of tanker and cargo ships.

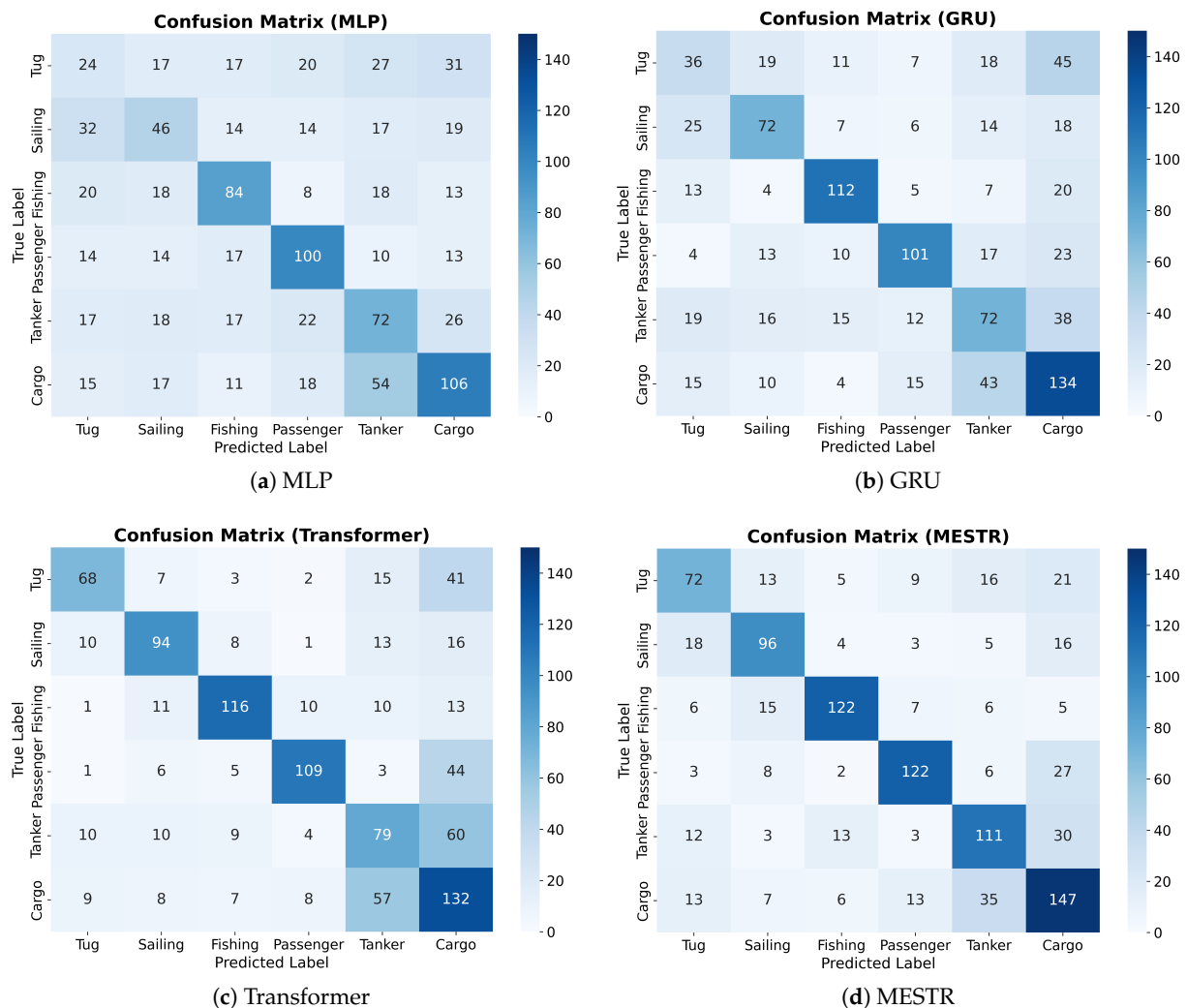


Figure 6. Confusion matrix comparison.

4.3. Visualization Analysis

We visualized the ship-type recognition results of different models, as shown in Figure 7. The performance of each method varies across different categories, with MESTR achieving the best recognition results. MLP has many misclassifications, such as misclassifying fishing as passenger, indicating its difficulty in learning the temporal dependencies of trajectories and its limitations in relying solely on static features for classification. GRU improves recognition in some categories through sequence modeling, such as correctly identifying fishing, but still confuses cargo and tanker, indicating its limitations in learning long-span trajectories. Transformer performs well in classifying fishing and passenger but still misclassifies cargo and tanker, reflecting that, while its global attention mechanism improves spatiotemporal feature learning, it is still insufficient in distinguishing local trajectory details. MESTR performs more stably across all categories, especially in successfully distinguishing cargo and tanker, validating that its motion-pattern-aware segment masking strategy and multi-task learning approach effectively enhance the model's ability to learn key trajectory patterns, making it the best performer in complex trajectory classification tasks.

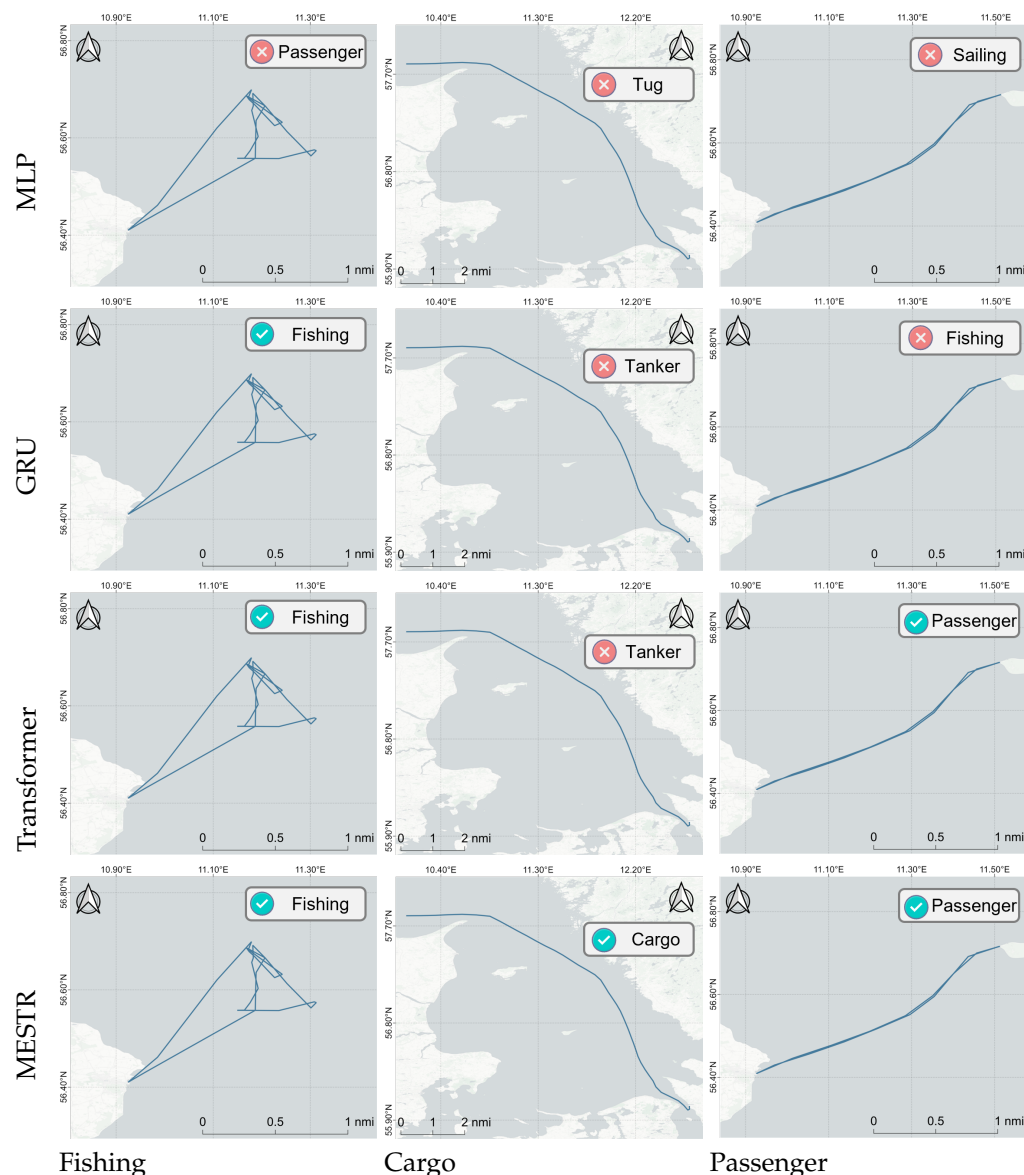


Figure 7. Visual analysis of ship-type identification results.

4.4. Ablation Study

To further measure the impact of different components on MESTR's performance, we conducted ablation experiments. As shown in Figure 8, the accuracy of without MPASM (removing the motion-pattern-aware segment masking strategy) drops to 63.5%, which is 3.5% lower than the complete MESTR, indicating that MPASM helps the model to focus on key movement features by masking high-maneuverability trajectory segments, improving classification performance. Without this strategy, the model may tend to learn global trajectory patterns while ignoring key local trajectory changes, affecting classification accuracy. The accuracy of without MSP (removing the masked segment prediction) further drops to 60.2%, which is 6.8% lower than MESTR, indicating that MSP plays a key role in enhancing the model's fine-grained feature extraction ability. Without this module, the model cannot strengthen its understanding of local trajectories through the trajectory reconstruction task, leading to a decline in classification performance. In summary, MPASM helps to guide the model to focus on key trajectory regions, while MSP enhances the model's ability to learn trajectory details through reconstructing masked trajectory segments. Together, they enable MESTR to achieve the best performance in ship classification tasks.

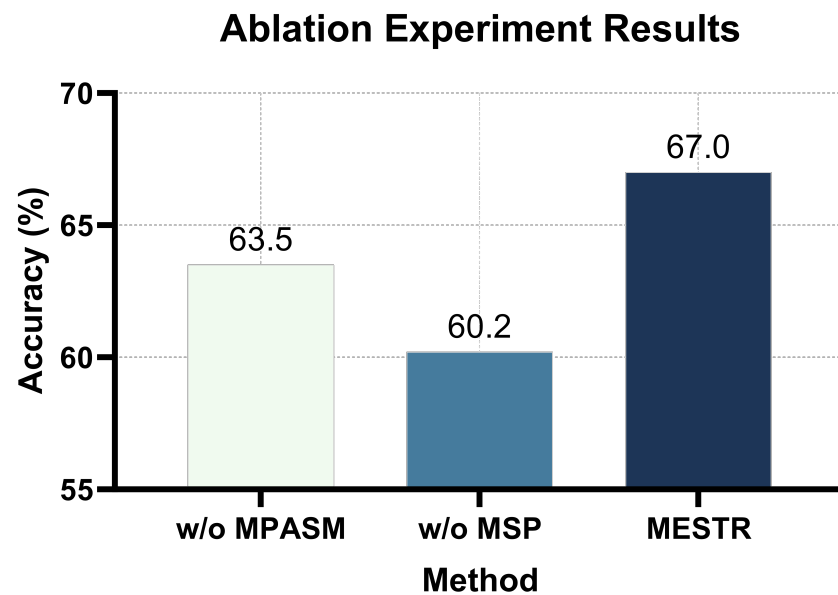


Figure 8. Ablation experiment results.

5. Discussion

5.1. Comparison with Existing Studies

Compared to the existing vessel-type classification methods, MESTR demonstrates advantages in both modeling strategies and classification performance. Traditional AIS-based identification approaches typically rely on extensive handcrafted feature engineering, which exhibits limitations when dealing with complex and nonlinear motion patterns. Although recent advancements have introduced deep learning methods such as recurrent neural networks to improve temporal sequence modeling, these methods still suffer from memory decay and insufficient capability to capture long-term dependencies.

In comparison, MESTR incorporates a Transformer-based architecture, leveraging the multi-head attention mechanism to more effectively capture long-range dependencies within vessel trajectories. Additionally, MESTR adopts a multi-task learning framework that jointly optimizes vessel-type prediction and trajectory segment reconstruction tasks, significantly enhancing the model's ability to learn both global patterns and local dynamic features. Notably, the proposed motion-pattern-aware trajectory segment masking strategy guides the model to focus on key trajectory segments with high maneuverability, thereby improving its capability to distinguish complex motion behaviors. In real-world applications, even in the presence of noise or missing data, the combination of these effective design choices and a flexible preprocessing pipeline enables MESTR to maintain robust performance.

The experimental results indicate that MESTR achieves an average improvement of 12.04% in overall classification accuracy compared to the existing baseline models. In particular, it performs exceptionally well in distinguishing vessel types with similar movement patterns, such as “oil tankers” and “cargo ships”. This demonstrates that MESTR not only enhances classification accuracy but also provides a robust and scalable behavior recognition approach for AIS data.

5.2. Limitations

Despite the strong performance of MESTR in the experiments, certain limitations remain. One lies in the fixed nature of the current trajectory segment masking strategy, which does not take into account dynamic environmental factors such as sea conditions or wind speed. Additionally, the current model is limited to six common vessel types.

Expanding to more fine-grained classifications, such as distinguishing container ships from ro-ro ships, may require modifications to the model architecture and augmentation of training samples. Furthermore, applying MESTR in real-time monitoring systems poses challenges. In addition to latency and hardware constraints, MESTR is designed for offline processing and is not suitable for online learning, which is often needed for real-time adaptation to dynamic vessel behaviors.

6. Conclusions

We proposed a multi-task enhanced ship-type recognition model (MESTR) based on AIS trajectory data. By combining two training tasks, *trajectory segment mask prediction* and *ship-type prediction*, MESTR helps the model to simultaneously learn the global patterns and local dynamic features of trajectories, improving its ability to distinguish between different ship types. Additionally, to enhance the model's attention to key trajectory patterns, we designed a motion-pattern-aware trajectory segment masking strategy that adaptively masks high-maneuverability trajectory segments, guiding the model to learn spatiotemporal detail changes in trajectories and improving recognition accuracy for ship types with similar shapes. We tested MESTR on real AIS trajectory data and the experimental results show that the method outperforms the traditional methods in ship-type recognition tasks, especially in distinguishing between ship types with similar movement patterns (e.g., cargo and tanker ships). Through visualization analysis and ablation experiments, we further validated the effectiveness of the trajectory segment mask prediction task and the motion-pattern-aware strategy.

In future research, we plan to further optimize the trajectory segment masking strategy to adaptively adjust based on environmental factors (e.g., sea conditions and wind speed). Additionally, we aim to expand the model's classification capability by incorporating a broader range of vessel types, including finer-grained categories, which will require structural adjustments and the inclusion of more diverse training samples. Furthermore, to support real-time applications, future work will explore lightweight and incremental learning approaches to improve adaptability and deployment efficiency in resource-constrained environments.

Author Contributions: L.C., N.J. and N.C. Methodology: N.C. Writing—original draft: N.C. Writing—review and editing: L.C., X.Z. and N.C. Supervision: N.J. Funding acquisition: L.C. and N.J. Statement: All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Innovation Science Fund of NUDT (grant no. 22-ZZCX-058).

Data Availability Statement: The data used in this study are publicly available from Danish Maritime Authority <https://www.dma.dk/safety-at-sea/navigational-information/ais-data>, (accessed on 1 March 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wolsing, K.; Roepert, L.; Bauer, J.; Wehrle, K. Anomaly detection in maritime AIS tracks: A review of recent approaches. *J. Mar. Sci. Eng.* **2022**, *10*, 112. [CrossRef]
2. Chen, X.; Hu, R.; Luo, K.; Wu, H.; Biancardo, S.A.; Zheng, Y.; Xian, J. Intelligent ship route planning via an A search model enhanced double-deep Q-network. *Ocean Eng.* **2025**, *327*, 120956. [CrossRef]
3. Wang, D.; Jing, Y. Ship collision risk analysis in port waters integrating GRA algorithm and BPNN. *Transp. Saf. Environ.* **2025**, *tdaf012*. [CrossRef]

4. Kessler, G.C.; Zorri, D.M. AIS Spoofing: A Tutorial for Researchers. In Proceedings of the 2024 IEEE 49th Conference on Local Computer Networks (LCN), Normandy, France, 8–10 October 2024; pp. 1–7. [\[CrossRef\]](#)
5. Zhang, B.; Ren, H.; Wang, P.; Wang, D. Research progress on ship anomaly detection based on big data. In Proceedings of the 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 16–18 October 2020; pp. 316–320. [\[CrossRef\]](#)
6. Huang, L.; Li, W.; Chen, C.; Zhang, F.; Lang, H. Multiple features learning for ship classification in optical imagery. *Multimed. Tools Appl.* **2018**, *77*, 13363–13389. [\[CrossRef\]](#)
7. Leclerc, M.; Tharmarasa, R.; Florea, M.C.; Boury-Brisset, A.C.; Kirubarajan, T.; Duclos-Hindié, N. Ship classification using deep learning techniques for maritime target tracking. In Proceedings of the 2018 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 10–13 July 2018; pp. 737–744. [\[CrossRef\]](#)
8. Bentes, C.; Velotto, D.; Tings, B. Ship classification in TerraSAR-X images with convolutional neural networks. *IEEE J. Ocean. Eng.* **2017**, *43*, 258–266. [\[CrossRef\]](#)
9. Yang, D.; Wu, L.; Wang, S. Can we trust the AIS destination port information for bulk ships?—Implications for shipping policy and practice. *Transp. Res. Part Logist. Transp. Rev.* **2021**, *149*, 102308. [\[CrossRef\]](#)
10. Zheng, Y. Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol. (TIST)* **2015**, *6*, 1–41. [\[CrossRef\]](#)
11. Zhong, H.; Song, X.; Yang, L. Vessel classification from space-based ais data using random forest. In Proceedings of the 2019 5th International Conference on Big Data and Information Analytics (BigDIA), Kunming, China, 8–10 July 2019, pp. 9–12. [\[CrossRef\]](#)
12. Baeg, S.; Hammond, T. Ship Type Classification Based on The Ship Navigating Trajectory and Machine Learning. In Proceedings of the ACM IUI Workshops, Sydney, Australia, 27–31 March 2023. [\[CrossRef\]](#)
13. Huang, I.L.; Lee, M.C.; Nieh, C.Y.; Huang, J.C. Ship classification based on ais data and machine learning methods. *Electronics* **2023**, *13*, 98. [\[CrossRef\]](#)
14. Fuscá, D.; Rahimli, K.; Leuzzi, R. Identification of vessel class with LSTM using kinematic features in maritime traffic control. *Int. J. Comput. Electr. Autom. Control. Inf. Eng.* **2022**, *16*, 1–4. Available online: <https://publications.waset.org/10012385/identification-of-vessel-class-with-lstm-using-kinematic-features-in-maritime-traffic-control> (accessed on 15 January 2024).
15. Han, X.; Zhou, Y.; Weng, J.; Chen, L.; Liu, K. Research on fishing vessel recognition based on vessel behavior characteristics from AIS data. *Front. Mar. Sci.* **2025**, *12*, 1547658. [\[CrossRef\]](#)
16. Sánchez Pedroche, D.; Amigo, D.; García, J.; Molina, J.M. Architecture for trajectory-based fishing ship classification with AIS data. *Sensors* **2020**, *20*, 3782. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Duan, H.; Ma, F.; Miao, L.; Zhang, C. A semi-supervised deep learning approach for vessel trajectory classification based on AIS data. *Ocean Coast. Manag.* **2022**, *218*, 106015. [\[CrossRef\]](#)
18. Meyer, R.; Kleynhans, W. Vessel classification using AIS data. *Ocean Eng.* **2025**, *319*, 120043. [\[CrossRef\]](#)
19. Shi, Q.; Li, W.; Tao, R.; Sun, X.; Gao, L. Ship Classification Based on Multifeature Ensemble with Convolutional Neural Network. *Remote Sens.* **2019**, *11*, 419. [\[CrossRef\]](#)
20. Chen, J.; Chen, K.; Chen, H.; Li, W.; Zou, Z.; Shi, Z. Contrastive learning for fine-grained ship classification in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–16. [\[CrossRef\]](#)
21. Dong, S.; Feng, J.; Fang, D. A novel multi-scale contrastive learning network for fine-grained ocean ship classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 9989–10005. [\[CrossRef\]](#)
22. Zhou, Y.; Daamen, W.; Vellinga, T.; Hoogendoorn, S.P. Ship classification based on ship behavior clustering from AIS data. *Ocean Eng.* **2019**, *175*, 176–187. [\[CrossRef\]](#)
23. Li, T.; Xu, H.; Zeng, W. Ship classification method for massive AIS trajectories based on GNN. In *Journal of Physics: Conference Series, Volume 2025, Proceedings of the 2021 3rd International Conference on Artificial Intelligence and Computer Science (AICS), Beijing, China, 29–31 July 2021*; IOP Publishing: Bristol, UK, 2021; Volume 2025, p. 012024. [\[CrossRef\]](#)
24. Ye, L.; Chen, X.; Liu, H.; Zhang, R.; Zhang, B.; Zhao, Y.; Zhou, D. Vessel Type Recognition Using a Multi-Graph Fusion Method Integrating Vessel Trajectory Sequence and Dependency Relations. *J. Mar. Sci. Eng.* **2024**, *12*, 2315. [\[CrossRef\]](#)
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv*, **2017**. [\[CrossRef\]](#)
26. Jawahar, G.; Sagot, B.; Seddah, D. What does BERT learn about the structure of language? In Proceedings of the ACL 2019-57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019. [\[CrossRef\]](#)
27. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [\[CrossRef\]](#)

28. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555. [[CrossRef](#)]
29. Loshchilov, I.; Hutter, F. Fixing weight decay regularization in adam. *arXiv* **2017**, arXiv:1711.05101. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.