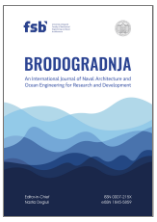
University of Zagreb
Faculty of Mechanical
Engineering and Naval
Architecture

journal homepage: www.brodogradnja.fsb.hr

Brodogradnja

An International Journal of Naval Architecture and
Ocean Engineering for Research and Development



Spatio-temporal prediction of vessel traffic flow based on GL-STFormer

Quandang Ma[1,2], Qihong Shao[1,2], Xu Du[1,2], Zhao Liu[1,2], Chi Zhang[3], Yongjin Guo[4], Mingyang Zhang[4]*

[1] *HubeiKey Laboratory of Inland Shipping Technology, School of Navigation, Wuhan University of Technology, Wuhan, China*
[2] *National Engineering Research Center for Water Transport Safety, Wuhan, China*
[3] *Department of Mechanics and Maritime Sciences, Chalmers University of Technology, Sweden*
[4] *State Key Laboratory of Submarine Geoscience, Shanghai Jiao Tong University, Shanghai, China*

ARTICLE INFO

Keywords:
AIS data
Vessel traffic flow prediction
Spatio-temporal features
CEEMDAN

ABSTRACT

Accurate prediction of vessel traffic flow is crucial for ensuring the safety of inland river shipping and enhancing the efficiency of traffic operations. Inland vessel traffic flow typically exhibits significant complexity and spatio-temporal dynamic characteristics. To address these challenges, this paper proposes a Global-Local Spatiotemporal Transformer (GL-STFormer) deep learning model. The Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) algorithm is utilized to decompose the original data into multi-feature inputs, effectively mitigating data non-stationarity. The model integrates Gated Recurrent Units (GRU) with a self-attention mechanism to extract temporal features of traffic patterns. The multi-head attention and local masking mechanisms of the Transformer model are employed to extract global and local spatial dependencies. Furthermore, the Whale Optimization Algorithm (WOA) is applied to optimize the model's hyperparameters. This study employs real-world Automatic Identification System (AIS) data from the Nantong waters of the Yangtze River for experimental validation. The results show that the proposed method significantly outperforms various baseline models in inland vessel traffic flow prediction. This study provides scientific support for precise traffic prediction and offers novel insights for the intelligent development of dynamic waterway traffic management.

1. Introduction

Inland waterway transportation is a vital component of modern integrated transportation systems, which can promote regional economic development and optimize global logistics networks [1-2]. With the continuous growth of vessel transportation, high-density inland waterways face dual challenges in navigation efficiency and safety management [3-6]. The Yangtze River, as a backbone channel of Chinese inland transportation network, directly impacts material circulation in its economic belt. However, the highly uneven spatial-temporal distribution of vessel traffic leads to significant congestion risks at key nodes. Dynamic changes in hydrological environments and frequent emergencies further increase navigation risk [7-9]. Thus, developing effective vessel traffic flow prediction models, which can enable intelligent waterway resource

[*] Corresponding author.
E-mail address: yongjinguo@sjtu.edu.cn

https://doi.org/10.21278/brod77309
Received 14 November 2025; Received in revised form 21 December 2025; Accepted 26 December 2025
Available online 12 January 2026
ISSN 0007-215X; eISSN 1845-5859

scheduling and enhance inland waterway system stability has practical values. Moreover, the widespread application of Automatic Identification System (AIS) provides high-resolution, high-frequency traffic information [10-11]. The AIS data records dynamic information of each vessel, including position, speed, and heading [12-13], which can support vessel traffic flow modeling and prediction. And the patterns including vessel movement periodicity, peak traffic trends, and abnormal flow fluctuations during specific time periods can be further identified.

In recent years, vessel traffic flow prediction has emerged as a significant branch of traffic flow prediction, which has gained widespread attention. As a time series prediction study, researchers have shifted their focus from temporal features to spatio-temporal characteristics through various modeling methods. Early research on vessel traffic flow prediction focused primarily on traditional statistical models. These models were based on linear time series modeling. Examples include Autoregressive (AR) models, Autoregressive Integrated Moving Average (ARIMA) models [14], and Kalman filtering [15]. These methods assume that historical data patterns can be predicted through linear combinations, which show limitations in handling non-linear dynamic patterns. In addition to traditional statistical models, machine learning and deep learning methods are widely used [16]. Traditional machine learning methods such as Support Vector Machine (SVM) [17], and Random Forest (RF) [18] were early applications. These methods have achieved certain effectiveness in capturing temporal patterns of traffic variations. However, when facing high-dimensional data or long-term prediction tasks, these methods demonstrate inherent limitations due to their restricted parameter space in modeling complex network traffic [19]. Deep learning technology has shown significant advantages in temporal feature analysis due to its powerful feature extraction capabilities. Recurrent Neural Networks (RNN) and their variants are widely used. RNNs model temporal dependencies through recurrent structures and hidden state updates. However, standard RNNs perform poorly in modeling long-term dependencies due to the vanishing gradient problem [20]. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) networks were introduced to address this issue [21-23]. They use gating mechanisms to achieve dynamic balance between long-term and short-term features. Several notable studies have been conducted in this field. For instance, Zhang et al. [24] proposed a BP neural network based on particle swarm optimization algorithm to predict vessel traffic flow in port areas. Dong et al. [25] developed a short-term vessel traffic flow prediction model that combines LSTM network with Dung Beetle Optimizer (DBO) algorithm. Chang et al. [26] proposed a combined CNN and Bi-GRU model for vessel traffic prediction in Qingdao Port's main channel.

However, modeling methods based solely on temporal features have limitations, such as ignoring of spatial distribution characteristics and potential interactions between different waterways. To capture the spatial features, researchers have turned to Graph Neural Networks (GNN). GNNs model spatial information through node-edge structures. They combine with time series prediction models to capture spatio-temporal features. These methods typically use adjacency matrices to describe spatial relationships between regions through physical waterway connections. Several researchers have made significant contributions in this area. For instance, Man et al. [27] proposed a spatiotemporal vessel traffic flow prediction model that combines the graph attention mechanism and bidirectional long short-term memory network to extract spatiotemporal features of vessel traffic patterns. Liang et al. [28] developed a spatio-temporal multi-graph convolutional network (STMGCN) that used distance graphs, interaction graphs, and correlation graphs to capture spatial information in inland waterways, significantly improving prediction accuracy. Ma et al. [29] introduced a Spatial-Temporal Attention Graph Convolution Network (STAGCN) model, which used an attention mechanism to compute dynamic adjacency matrices combined with graph convolution networks to capture Ship Time Headway (STH) patterns across different water areas. Lie et al. [30] proposed a semi-dynamic spatio-temporal graph neural network (SDSTGNN) model, using pre-defined adjacency matrices and adaptive matrices to construct semi-dynamic adjacency matrices, which enhanced the ability to capture spatial information. In summary, existing research methods have effectively integrated both temporal and spatial information, addressing to some extent the limitations associated with focusing solely on temporal features. However, several critical challenges remain:

1. Traditional single-model methods can only consider either temporal or spatial features. This leads to reduced prediction accuracy for spatio-temporal vessel traffic flow data.

2. Fully extracting spatial information from inland waterway networks remains challenging. Existing methods struggle to capture both global and local spatial information.

3. The complex characteristics of multi-waterway vessel flow data are often difficult to analyze effectively.

To address these issues, this paper proposes the Global-Local Spatiotemporal Transformer (GL-STFormer) model. It integrates three core modules: The Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) for signal decomposition, GRU and Self-Attention for temporal feature extraction, and Transformer for spatial feature extraction. Considering the inherent non-stationarity and non-linearity of vessel traffic flow, the CEEMDAN algorithm serves as the initial component [31]. Compared with EMD [32] and EEMD [33], CEEMDAN effectively mitigates mode mixing and reduces reconstruction errors through adaptive noise strategies. It decomposes complex raw signals into simplified components, providing high-quality inputs for the deep learning network. For spatial modeling, the Transformer architecture is utilized. While the Transformer model has become a research hotspot in time series prediction due to its outstanding performance in Natural Language Processing (NLP) tasks [34], its self-attention mechanism is also inherently suitable for capturing spatial dependencies. Specifically, the weight matrix of the multi-head attention mechanism functions as a dynamic adjacency matrix to extract influence weights for each node. By integrating temporal and spatial features, the GL-STFormer model constructs a comprehensive short-term vessel traffic flow prediction framework. It can accurately simulate dynamic changes in traffic flow and significantly improve prediction accuracy, which may promote the efficiency of decision-making processes.

The main contributions of this paper are:

1. Use the Transformer model to capture both global and local spatial features of inland vessel traffic flow.

2. Propose a GL-STFormer deep learning model to explore the complex dynamic patterns of vessel traffic flow in both temporal and spatial dimensions.

3. Compared with baseline models, the GL-STFormer shows better performance.

The structure of this paper is as follows: Section 2 describes the vessel traffic data collection process. Section 3 explains the model framework in detail. Section 4 discusses experimental data settings and result analysis. Section 5 summarizes the research findings and looks at future research directions.

2. Problem analysis and data processing

2.1 Problem description

Vessel traffic flow prediction is an important part of intelligent transportation systems. In waterway transportation, the inland traffic network can be represented as an undirected graph $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathrm{A})$, where $V = (V_1, V_2, \ldots, V_N)$ is the set of N nodes, E is the set of M edges, and $A \in \mathrm{R}^{N\times N}$ is the adjacency matrix of the graph. The historical vessel traffic data at each node can be represented as $X = (X_{t-M+1}, X_{t-M+2}, \ldots, X_t) \in R^{M\times N}$, where M is the number of historical time steps of vessel traffic data. The predicted future vessel traffic data is represented as $\hat{X} = (X_{t+1}, X_{t+2}, \ldots, X_{t+P}) \in R^{P\times N}$, where P represents the number of future time steps. X_t represents the vessel traffic data at time t for the N nodes. Therefore, the vessel traffic flow prediction problem can be expressed as the following equation:

$$(X_{t+1}, X_{t+2}, \ldots, X_{t+P}) = f(X_{t-M+1}, X_{t-M+2}, \ldots, X_t) \tag{1}$$

where f is the prediction model used in this study. Figure 1 illustrates the overall process from training to final prediction of the model.

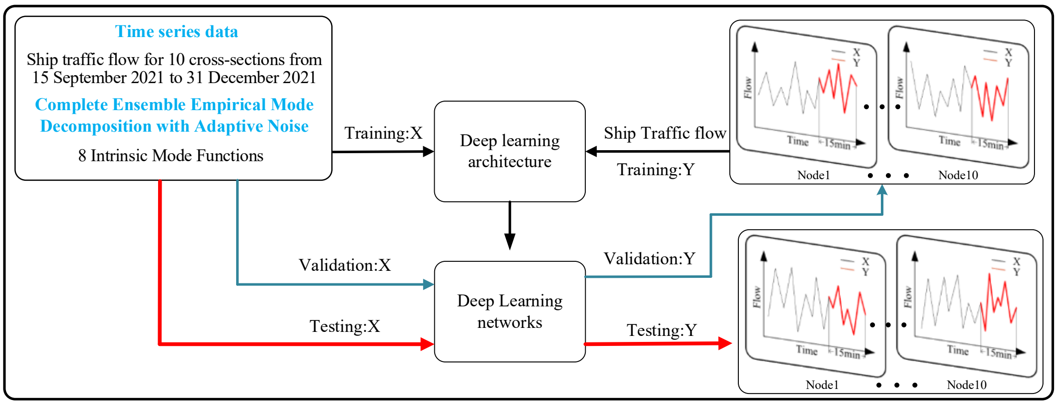

Time series data
Ship traffic flow for 10 cross-sections from 15 September 2021 to 31 December 2021
Complete Ensemble Empirical Mode Decomposition with Adaptive Noise
8 Intrinsic Mode Functions
Training:X
Deep learning architecture
Ship Traffic flow
Training:Y
Validation:X
Testing:X
Deep Learning networks
Validation:Y
Testing:Y
Flow
Time
15min
X
Y
Node1
Node10

Fig. 1 Training, validation, and testing process for vessel traffic flow prediction using deep learning

2.2 Cross-sectional vessel traffic data collection

This section examines all vessels passing through cross-sections to mitigate missing values during periods of low vessel traffic in the channel. The cross-section is established with two endpoints denoted as α and β. The mathematical representation of the cross-section is $Virsection^N = \left(lon_\alpha^N, lon_\beta^N, lat_\alpha^N, lat_\beta^N\right)$, where N denotes the finite set of cross-sections. A vessel's passage through the cross-section is validated by the intersection of its sub-trajectory with the cross-section. The sub-trajectory comprises two consecutive trajectory points that are nearest to the cross-section. The methodology involves initially organizing the data according to vessel MMSI and temporal sequence, followed by an analysis of all trajectory points for each vessel. In the case where a vessel traverses from the left to the right side of a cross-section, its position at the previous time step is denoted as $P_{i,left}^N$ with coordinates $(lon_{i,left}^N, lat_{i,left}^N)$, while its subsequent position is represented as $P_{i,right}^N$ with coordinates $(lon_{i,right}^N, lat_{i,right}^N)$. The variable i denotes the i-th trajectory point. In two-dimensional plane geometry, the sign of the vector cross product indicates the relative directional relationship between two vectors. This mathematical property enables the determination of relative positions between points and line segments. As illustrated in Figure 2, vector cross products are employed to determine whether vessel trajectory points lie on opposite sides of the virtual cross-section and whether cross-section control points are situated on opposite sides of the vessel trajectory. The mathematical formulation is expressed as follows:

$$\vec{S} = \overrightarrow{\alpha\beta} = (lon_\beta^N - lon_\alpha^N, lat_\beta^N - lat_\alpha^N) \tag{2}$$

$$\vec{V_1} = \overrightarrow{\alpha P_{i,left}^N} = (lon_{i,left}^N - lon_\alpha^N, lat_{i,left}^N - lat_\alpha^N) \tag{3}$$

$$\vec{V_2} = \overrightarrow{\alpha P_{\iota,r\iota ght}^N} = (lon_{i,right}^N - lon_\alpha^N, lat_{i,right}^N - lat_\alpha^N) \tag{4}$$

$$\vec{T} = \overrightarrow{P_{\iota,left}^N P_{\iota,r\iota ght}^N} = (lon_{i,right}^N - lon_{i,left}^N, lat_{i,right}^N - lat_{i,left}^N) \tag{5}$$

$$\vec{V_3} = \overrightarrow{P_{\iota,left}^N \alpha} = (lon_\alpha^N - lon_{i,left}^N, lat_\alpha^N - lat_{i,left}^N) \tag{6}$$

$$\vec{V_4} = \overrightarrow{P_{\iota,left}^N \beta} = (lon_\beta^N - lon_{i,left}^N, lat_\beta^N - lat_{i,left}^N) \tag{7}$$

Here, vector $\vec{S}$ represents the direction from α to β. Vector $\vec{V_1}$ and $\vec{V_2}$ represent the directions from α to the left and right adjacent trajectory points of the sub-trajectory, respectively. Vector $\vec{T}$ represents the vessel's movement from its previous position to its subsequent position. Vectors $\vec{V_3}$ and $\vec{V_4}$ represent the directions from the vessel's previous position to points α and β, respectively.

When $cross_1 \cdot cross_2 < 0$ and $cross_3 \cdot cross_4 < 0$, it indicates that the vessel has passed through the cross-section:

$$cross_1 = \vec{V_1} \times \vec{S} = \begin{vmatrix} lon_{i,left}^N - lon_\alpha^N & lat_{i,left}^N - lat_\alpha^N \\ lon_\beta^N - lon_\alpha^N & lat_\beta^N - lat_\alpha^N \end{vmatrix} \tag{8}$$

$$cross_2 = \vec{V_2} \times \vec{S} = \begin{vmatrix} lon_{i,right}^N - lon_\alpha^N & lat_{i,right}^N - lat_\alpha^N \\ lon_\beta^N - lon_\alpha^N & lat_\beta^N - lat_\alpha^N \end{vmatrix} \tag{9}$$

$$cross_3 = \vec{T} \times \vec{V_3} = \begin{vmatrix} lon_{i,right}^N - lon_{i,left}^N & lat_{i,right}^N - lat_{i,\text{left}}^N \\ lon_\alpha^N - lon_{i,left}^N & lat_\alpha^N - lat_{i,left}^N \end{vmatrix} \tag{10}$$

$$cross_4 = \vec{T} \times \vec{V_4} = \begin{vmatrix} lon_{i,right}^N - lon_{i,left}^N & lat_{i,right}^N - lat_{i,left}^N \\ lon_\beta^N - lon_{i,left}^N & lat_\beta^N - lat_{i,left}^N \end{vmatrix} \tag{11}$$

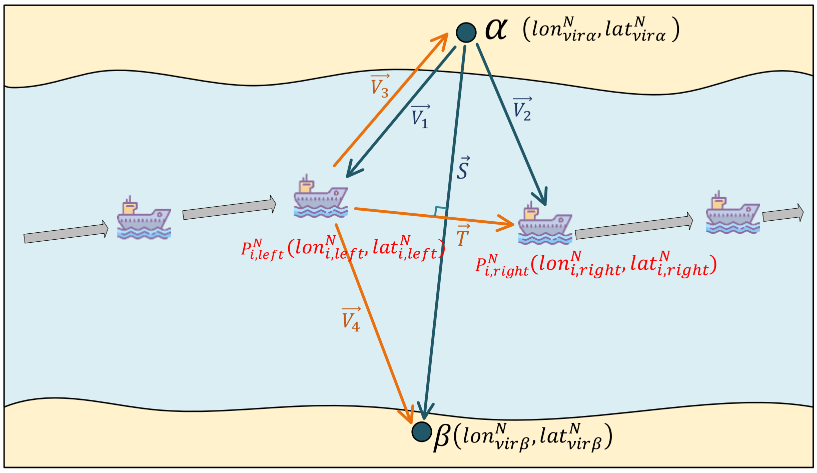
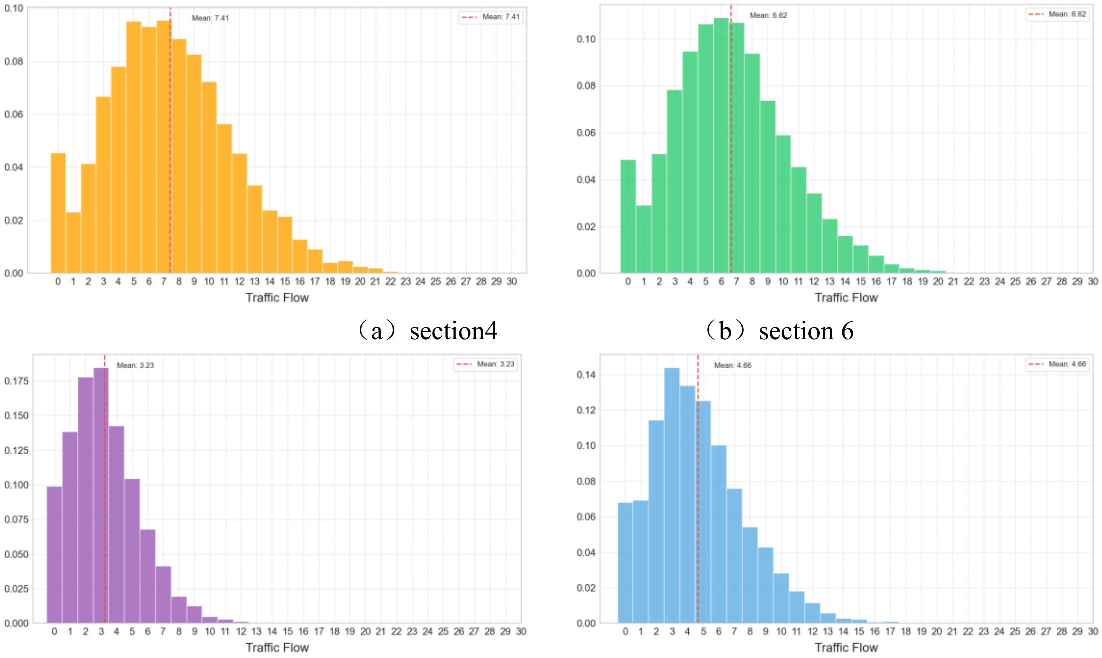

α $(lon_{vir\alpha}^N, lat_{vir\alpha}^N)$
$\vec{V_3}$
$\vec{V_1}$
$\vec{V_2}$
$\vec{S}$
$\vec{T}$
$P_{i,left}^N(lon_{i,left}^N, lat_{i,left}^N)$
$P_{i,right}^N(lon_{i,right}^N, lat_{i,right}^N)$
$\vec{V_4}$
$\beta(lon_{vir\beta}^N, lat_{vir\beta}^N)$

Fig. 2 Vector cross product method for detecting vessel passage through selected cross-sections



（a）section4 （b）section 6

（c）section 8 （d）section 10

Fig. 3 Distribution of vessel traffic flow data

After determining whether vessels pass through virtual cross-sections, we developed a vessel traffic flow dataset by calculating vessel arrival times at cross-sections, referencing previous research methods [35]. The final dataset contains 10,326 vessel traffic flow records. As shown in Figure 3, we conducted probability density analysis on the vessel flow data to visualize its macro characteristics. The analysis reveals that the number of vessels passing through the cross-section within 15-minute intervals in the Nantong section of the Yangtze River typically ranges from 0 to 20 vessels. This pattern aligns with the navigational characteristics of the Yangtze River.

3. Modelling

This section presents a GL-STFormer model framework for multi-cross-section vessel traffic flow prediction in inland waterways. As shown in Figure 4, the framework consists of three steps:

1. Experimental data preparation. The framework references AIS data preprocessing methods proposed in previous studies to enhance vessel trajectory quality. Subsequently, vessel trajectory density is visualized through heat maps, and cross-sections are selected perpendicular to waterways in high-density areas. The number of vessels passing through all cross-sections is calculated using vector cross-productmethod.

2. CEEMDAN decomposition of vessel traffic flow data. To reduce the volatility and complexity of raw data, the CEEMDAN decomposition algorithm is employed to decompose the original data into multiple Intrinsic Mode Functions (IMFs). Min-max normalization is applied to facilitate rapid model convergence.

3. Deep learning model for vessel traffic flow prediction. The GL-STFormer model is developed to accurately predict vessel traffic flow by learning complex spatiotemporal dependencies. In the temporal dimension, the model integrates GRU and self-attention mechanisms to capture evolutionary patterns. In the spatial dimension, the Transformer architecture is employed to dynamically extract and fuse global and local spatial information. Furthermore, the Whale Optimization Algorithm (WOA) is applied to optimize the model's hyperparameters.

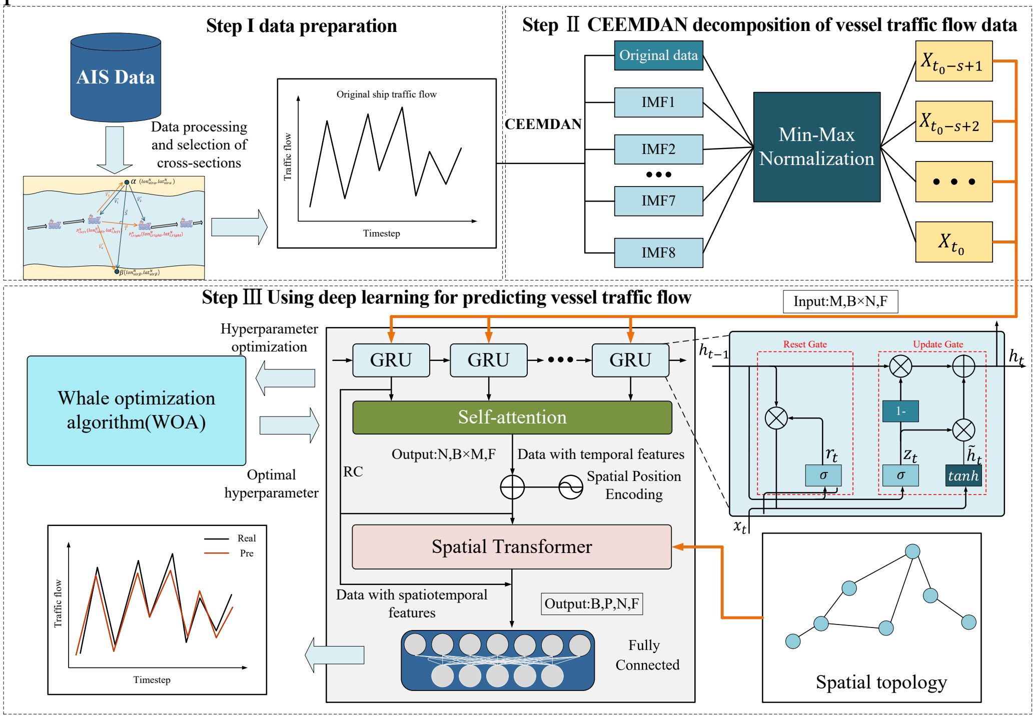


Fig. 4 Research framework

3.1 Inland waterway traffic network

In road traffic, sensors are typically installed at accident-prone locations or congested areas. However, inland waterways lack dedicated traffic monitoring infrastructure. Previous studies placed cross-sections aligned with bridges or perpendicular to channels. Areas with high vessel density present complex navigation environments and increased safety risks, making them priority locations for studying traffic flow patterns and conducting predictions.

In this study's inland waterway network construction, we selected cross-sections perpendicular to channels at locations with dense vessel trajectories. As shown in Figure 5, the vessel trajectory heat map, generated from preprocessed AIS data, clearly indicates vessel activity levels in the waterway. We selected ten virtual cross-sections in areas with relatively dense trajectories. These virtual cross-sections serve as equivalents to road traffic sensors, recording vessel counts at specific locations. Table 1 lists the control point coordinates of these ten virtual cross-sections.

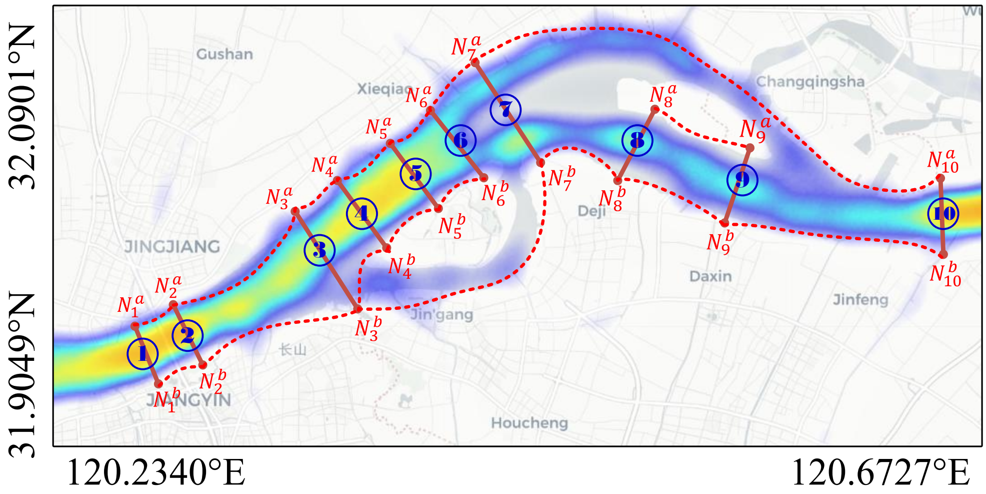


Fig. 5 Construction of inland waterway traffic network with virtual cross-sections at high vessel trajectory density locations

Table 1 Control point coordinates of ten virtual cross-sections

$Cross-sections$	Endpoint	Longitude E(°)	Latitude N(°)
$Node_1$	N_1^a	120.261389	31.947778
	N_1^b	120.270000	31.929722
$Node_2$	N_2^a	120.276111	31.955556
	N_2^b	120.288333	31.937222
$Node_3$	N_3^a	120.339167	31.996944
	N_3^b	120.370556	31.959722
$Node_4$	N_4^a	120.363056	32.011389
	N_4^b	120.388333	31.984167
$Node_5$	N_5^a	120.382222	32.023333
	N_5^b	120.408333	31.997500
$Node_6$	N_6^a	120.398889	32.035278
	N_6^b	120.425833	32.010556
$Node_7$	N_7^a	120.429722	32.055000
	N_7^b	120.459167	32.018889
$Node_8$	N_8^a	120.528056	32.042778
	N_8^b	120.507222	32.008889
$Node_9$	N_9^a	120.570556	32.026111
	N_9^b	120.557222	31.992778
$Node_{10}$	N_{10}^a	120.653611	32.011389
	N_{10}^b	120.659167	31.981667

Considering the closed and interconnected nature of inland waterways, we defined an inland traffic network based on waterway geographical structures and relative positions between cross-sections. Each cross-section represents a node. The adjacency matrix element A_{ij} is set to 1 if nodes N_i and N_j are connected in graph , and 0 otherwise, as shown in equation (12):

$$A_{ij} = \begin{cases} 1, if N_i \ connceted \ to \ N_j, i \neq j \\ 0, otherwise \end{cases} \tag{12}$$

Here, N_i and N_j represent the i-th and j-th nodes in the undirected graph, respectively.

3.2 Complete ensemble empirical mode decomposition with adaptive noise

Original vessel traffic flow signals exhibit nonlinear and non-stationary characteristics. Consequently, the CEEMDAN algorithm is employed to decompose the original data. As illustrated in Figure 6, CEEMDAN transforms the complex original signal into multiple IMFs. These IMFs possess simplified patterns that are effectively learned by deep learning models. By utilizing adaptive noise strategies and iterative optimization, CEEMDAN generates precise and stable components. This mechanism reduces error accumulation caused by noise introduction and improves decomposition resolution.

The steps for implementing CEEMDAN is as follows:

Step 1: The original vessel traffic signal is denoted as $x(t)$.To perform decomposition, we first add noise to the signal. A Gaussian white noise $n^{(i)}(t)$ is generated and added to the original signal to form a new signal $x'^{(i)}(t)$:

$$x'^{(i)}(t) = x(t) + \alpha_1 n^{(i)}(t) \tag{13}$$

where α_1 is the noise intensity factor. $n^{(i)}(t)$ is the i-th generated Gaussian white noise with mean 0 and variance σ_n^2. This operation is repeated m times to obtain m noise signals $x'^{(i)}(t)$.

Step 2: For each noise-added signal $x'^{(i)}(t)$, the Empirical Mode Decomposition (EMD) method is used to obtain the first IMF component $IMF_1^{(i)}(t)$ and residual $r_1^{(i)}(t)$:

$$IMF_1(t) = \frac{1}{m}\sum_{i=1}^{m} IMF_1^{(i)}(t) \tag{14}$$

$$r_1(t) = x(t) - IMF_1(t) \tag{15}$$

where $r_1(t)$ is the residual signal after removing the first IMF.

Step 3: New noise $n^{(i)}(t)$ is added to the residual signal $r_1(t)$ to generate a new noise signal $r_1'^{(i)}(t)$. EMD is performed on each new noise signal $r_1'^{(i)}(t)$ to obtain the second IMF component $IMF_2^{(i)}(t)$ and residual $r_2^{(i)}(t)$:

$$r_1'^{(i)}(t) = r_1(t) + \alpha_2 n^{(i)}(t) \tag{16}$$

$$IMF_2(t) = \frac{1}{m}\sum_{i=1}^{m} IMF_2^{(i)}(t) \tag{17}$$

$$r_2(t) = r_1(t) - IMF_2(t) \tag{18}$$

Step 4: The above steps are repeated to extract more IMF components. For each new residual $r_k(t)$, noise is continually added to generate new noise signals $r_k'^{(i)}(t)$. EMD is then performed again to extract the (k+1)-th IMF component from the new noise signal:

$$r_k'^{(i)}(t) = r_k(t) + \alpha_{k+1} n^{(i)}(t) \tag{19}$$

$$IMF_{k+1}(t) = \frac{1}{m}\sum_{i=1}^{m} IMF_{k+1}^{(i)}(t) \tag{20}$$

$$r_{k+1}(t) = r_k(t) - IMF_{k+1}(t) \tag{21}$$

This process continues until reaching the maximum iteration number *K*:

$$\hat{x}(t) = \sum_{j=1}^{K} IMF_j(t) + r_K(t) \tag{22}$$

$IMF_j(t)$ represents the *j*-th Intrinsic Mode Function (IMF). $r_K(t)$ is the final residual, typically representing the low-frequency trend or residual components in the signal. $\hat{x}(t)$ is the vessel traffic flow signal after CEEMDAN decomposition.

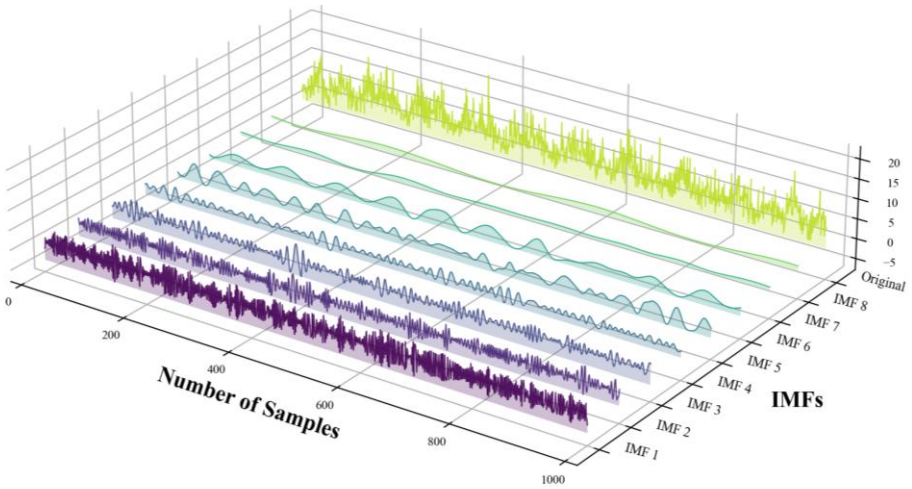


Fig. 6 IMF components after CEEMDAN decomposition of vessel traffic flow data (partial data from cross-section 1)

3.3 Vessel traffic flow prediction based on deep learning method

3.3.1 GRU

Traditional recurrent neural networks can process sequential tasks but often suffer from severe gradient vanishing when handling long-term dependencies. Compared to traditional RNNs, GRU introduces a gating mechanism that effectively mitigates the vanishing gradient problem while simplifying the network structure and reducing computational complexity while maintaining performance.

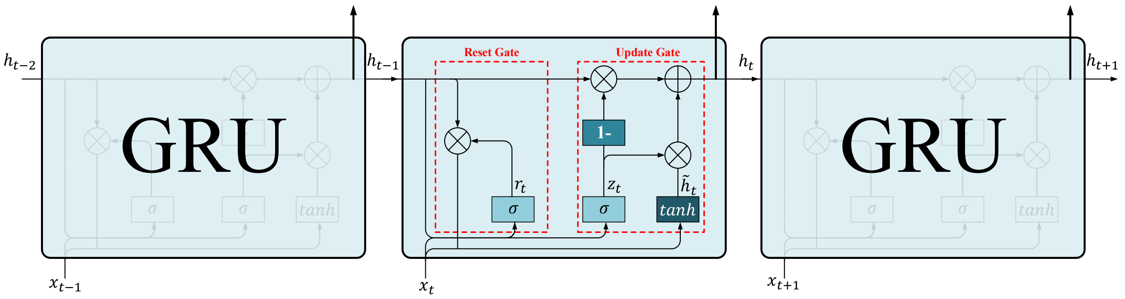


Fig. 7 Gated Recurrent Unit structure

GRU inherits LSTM’s ability to capture both long-term and short-term dependencies while improving computational efficiency by merging gate mechanisms. This makes it widely applicable in time series prediction scenarios, such as traffic flow, stock prices, and temperature forecasting tasks [36-38]. When processing time series tasks, GRU controls information flow through its gating mechanism based on input data and network hidden states. It selectively retains important information while filtering redundant data to generate high-quality temporal features for subsequent tasks.

As shown in Figure 7, GRU's core working mechanism includes two gates: Update Gate and Reset Gate, along with a hidden state representation. These two gates effectively capture dynamic temporal relationships by controlling the fusion of current and historical information. The specific calculation process of GRU is as follows:

Update Gate: The update gate determines the weight ratio between the previous hidden state h_{t-1} and current input x_t in generating the new hidden state:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{23}$$

Here, z_t is the update gate value controlling the influence of historical information on current output; W_z, U_z, b_z are learnable model parameters; σ is the sigmoid activation function.

Reset Gate: The reset gate determines whether to ignore certain parts of the previous hidden state:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{24}$$

Here, r_t is the reset gate value adjusting the dependency strength between current input and historical information; W_r, U_r, b_r are learnable model parameters.

Candidate Hidden State: The candidate hidden state is jointly generated by the current input and the reset gate-adjusted previous hidden state:

$$\tilde{h}_t = tanh\ (W_h x_t + r_t \odot (U_h h_{t-1}) + b_h) \tag{25}$$

The $tanh$ activation function limits the output value range, and $\odot$ represents element-wise multiplication. The final hidden state is regulated by the update gate controlling information interaction between the candidate hidden state and previous hidden state:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{26}$$

In this paper, the new unit state h_t is employed as both the output of the GRU structure and the input of the self-attention layer.

3.3.2 Self-attention

However, the GRU model can only process data sequentially along the time series, making it difficult to capture nonlinear or irregular interactions between time steps. In such scenarios, relying solely on GRU makes it challenging to obtain ideal temporal feature representations.

To address these issues, the Self-attention mechanism was introduced as a complementary approach [39]. The attention mechanism is inspired by how the human brain processes information. It achieves information processing by selectively allocating computational resources. Similar to how humans focus on specific tasks while ignoring surrounding distractions, this mechanism highlights important information while weakening or filtering out secondary information. This enables precise capture and processing of key information [40]. Therefore, we introduce the output of the GRU model $H_t = (h_1, h_2, h_3, \dots, h_t)$ into the self-attention mechanism, which can dynamically adjust the information weights of different time steps and significantly enhance the modeling capability for long-range interaction relationships [41]. Figure 8 illustrates the internal mechanism of self-attention.

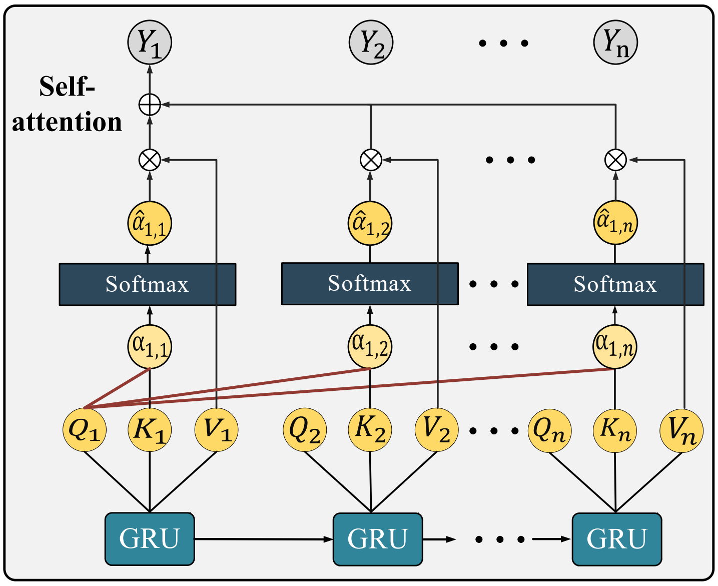


Fig. 8 Self-attention mechanism

The self-attention mechanism generates weight matrices based on input sequences to focus on important time steps while weakening secondary information. The specific process includes the following steps, with formulas:

$$Q = H_t W_q, K = H_t W_k, V = H_t W_v \tag{27}$$

$$\alpha_{ij} = \frac{QK^T}{\sqrt{d_k}} \tag{28}$$

where $Q \in R^{M\times d_k}, K \in R^{M\times d_k}, V \in R^{M\times d_k}$ and QK^T represents the dot product of query vectors and key vectors for different sequence positions, resulting in a similarity matrix. The scaling factor $\sqrt{d_k}$ prevents dot product values from becoming too large in high dimensions, which could cause gradient vanishing or unstable updates:

$$\hat{\alpha}_{ij} = softmax(\alpha_{ij}) = \frac{\exp\,(\alpha_{ij})}{\sum_{j=1}^{M} \exp\,(\alpha_{ij})} \tag{29}$$

The attention weight matrix is obtained by normalizing α_{ij} through the softmax function, ensuring weights sum to 1 across time steps. This weight matrix represents the information dependency relationships between each input sequence position and other positions:

$$T_i = \sum_{j=1}^{M} \hat{\alpha}_{ij} V_j, T \in R^{M\times d_v} \tag{30}$$

Here, T_i represents the output feature at time step i; V_j represents the feature at time step j in the value matrix; weight value α_{ij} reflects the importance of time step j to time step i. Finally, the output matrix T preserves the feature information from the input sequence while enhancing the importance of key time steps, enabling the model to generate more expressive temporal features.

3.3.3 Transformer

In the preceding temporal extraction layer, GRU and self-attention mechanisms were employed to capture time-series variations. However, exclusive reliance on temporal features ignores inter-nodal dependencies, thereby limiting prediction accuracy. Therefore, the comprehensive extraction of spatial features is critical. Existing Graph Neural Networks face specific limitations. For instance, Graph

Convolutional Networks (GCN) [42] are restricted by static topological assumptions, while Graph Attention Networks (GAT) [43] primarily focus on local neighbors and often overlook long-range global dependencies.

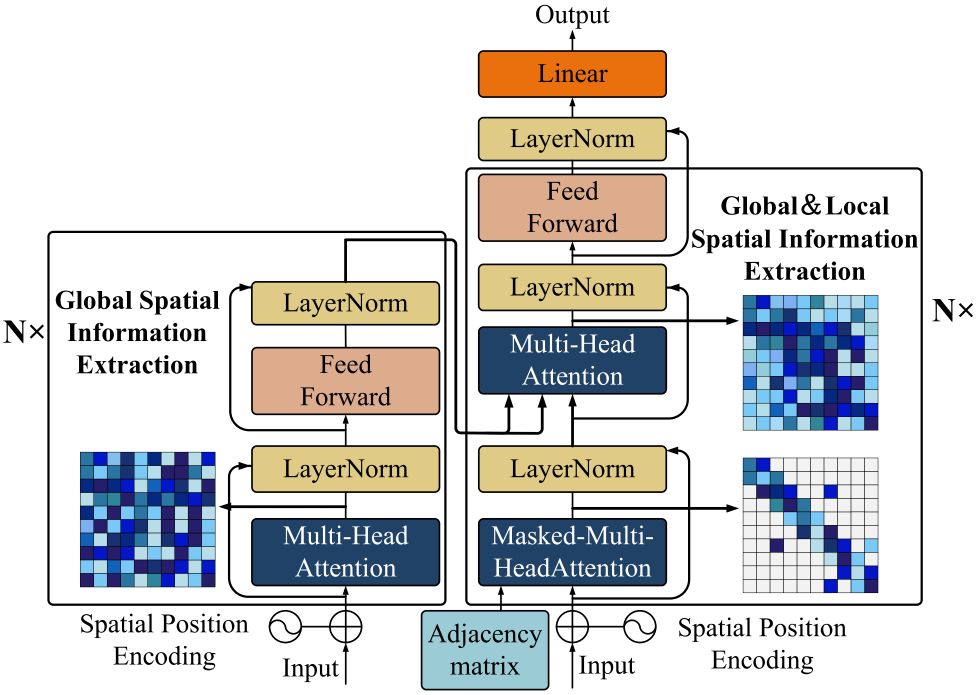

Output
Linear
LayerNorm
Feed Forward
Global&Local Spatial Information Extraction
LayerNorm
Multi-Head Attention
N×
LayerNorm
Masked-Multi-HeadAttention
Global Spatial Information Extraction
N×
LayerNorm
Feed Forward
LayerNorm
Multi-Head Attention
Spatial Position Encoding
Input
Adjacency matrix
Input
Spatial Position Encoding

Fig. 9 Transformer model architecture

To overcome these constraints,, this study adopts the Transformer model as the spatial feature extraction module [44], as shown in Figure 9. Unlike traditional graph-based methods, the encoder-decoder architecture of the Transformer offers distinct advantages. The encoder utilizes multi-head self-attention mechanism to capture global dependencies within the traffic network, whereas the decoder employs masked multi-head attention mechanism to refine local spatial features. Subsequently, the cross-attention mechanism integrates these features to extract global-local spatial correlations. Consequently, this architecture enables the simultaneous extraction of global spatial correlations across the network and local interactions between adjacent nodes.

3.3.4 Position Encoding

The core self-attention mechanism of Transformer is essentially a fully connected structure. Since this mechanism processes data solely based on content relationships between elements without explicitly capturing positional information in the input sequence, Transformer cannot perceive the arrangement order of input sequence elements without additional mechanisms. Since data signals correspond to cross-sections with specific physical locations in geographical space, and this positional information is crucial for establishing spatial dependencies between cross-sections, incorporating position encoding to explicitly add this positional information is vital for model construction.

To effectively mark spatial positional relationships between nodes, Transformer introduces sinusoidal position embeddings to record positional information of time series data, with the formula:

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d}\right) \tag{31}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d}\right) \tag{32}$$

where pos represents the position in the sequence, i represents the sequence index, and d represents the dimension of sequence features [45]. The output T from the temporal feature extraction layer is combined with positional encodings to form $X = T + PE$, where $X \in R^{N\times H\times M}$, $PE \in R^{N\times H\times M}$.

3.3.5 Global spatial feature extraction

Inland waterway traffic networks entail potential long-range dependencies, where geographically distant nodes may maintain strong correlations due to spatiotemporal traffic propagation effects. To capture these global characteristics, the Transformer encoder module employs the multi-head self-attention mechanism. Unlike GCN which rely on static adjacency matrices to aggregate local information, the Encoder dynamically computes attention weights between all node pairs based on real-time input features. This fully connected attention structure allows the model to extract global dependencies across the entire waterway topology, thereby providing a holistic spatial representation that complements local physical connections.

Using the position-encoded feature tensor $X \in \mathrm{R}^{M\times H\times N}$, new global spatial feature vector $\varphi \in R^{M\times H_\mathrm{g}\times N}$ is obtained through the following calculations:

$$FFN(x) = W_2 ReLU(W_1 x + b_1) + b_2 \tag{33}$$

where W_1, W_2, b_1, b_2 are learnable parameters;

$$LN(x) = \frac{x-\mu}{\sigma}\cdot\gamma + \beta \tag{34}$$

where μ is the mean, σ is the standard deviation, γ and β are learnable scaling coefficients and offset values:

$$Q_i^g = XW_i^1, K_i^g = XW_i^2, V_i^g = XW_i^3 \tag{35}$$

$$A_i^g = softmax\left(\frac{Q_i^g (K_i^g)^T}{\sqrt{d_k}}\right) \tag{36}$$

$$head_i = Attention(Q_i^g, K_i^g, V_i^g) = A_i^g V_i^g \tag{37}$$

$$MutilheadAttention(Q,K,V) = Concat(head_1, \cdots, head_h)W^O \tag{38}$$

$$\zeta = LN(X + MutilheadAttention(Q,K,V)) \tag{39}$$

$$\varphi = LN(\zeta + FFN(\zeta)) \tag{40}$$

where $W_i^1, W_i^2, W_i^3 \in R^{N\times d_k}$, $W^O \in R^{(h\times d_k)\times N}$ are linear transformation matrices. $d_k = \frac{H}{h}$, d_k is the feature dimension of each head, h is the number of attention heads, A_i^g represents the weight matrix between the i-th node and other nodes, where these weights indicate the strength of dependency relationships between nodes. The final output φ, processed through multi-head self-attention mechanism, residual connections, normalization layers, and fully connected layers, constitutes the total output of the encoder. This output contains node representations that incorporate fused global spatial features, where contextual information from other nodes is further integrated based on the nodes' inherent features, thereby forming global spatial characteristics.

3.3.6 Global-local spatial feature extraction

The spatial distribution of vessel traffic flow is governed by the topological structure of waterways. To integrate these physical constraints while retaining the global context, the Decoder module is designed to extract local features and fuse them with global representations.

To capture the physical connectivity, a masked multi-head self-attention mechanism is implemented. A binary mask matrix, derived from the physical adjacency of the waterway, is imposed on the attention scores. By assigning negative infinity to the weights of unconnected nodes, the attention mechanism is mathematically restricted to physically adjacent neighbors. This operation enforces the extraction of explicit local spatial features that strictly adhere to the static graph topology. The formula for masked attention score processing is given by:

$$Q_i^l = XW_i^4, K_i^l = XW_i^5, V_i^l = XW_i^6 \tag{41}$$

$$A_i^l = Softmax\left(\frac{Q_i^l(K_i^l)^T}{\sqrt{d_k}} + mask\right) \tag{42}$$

$$\psi_i = Attention(Q_i, K_i, V_i, mask) = Softmax\left(\frac{Q_i^l(K_i^l)^T}{\sqrt{d_k}} + mask\right)V_i^l \tag{43}$$

where mask represents the mask matrix, $mask = \begin{cases} 0, & M_{ij} = 1 \\ -\infty, & M_{ij} = 0' \end{cases}$

$$\psi = MutilheadAttention(\psi_1, \psi_2, \cdots \psi_n) \tag{44}$$

where $W_i^4, W_i^5, W_i^6 \in R^{H \times d_k}$. A_i^l represents the attention weight matrix where the *i*-th node only focuses on neighboring nodes. The final attention weights are used to weight the value vectors of neighboring nodes, updating the local representation of nodes.

Subsequently, the integration of global and local information is achieved through multi-head cross-attention. In this layer, the extracted local features function as Q, while the global spatial features from the Encoder output serve as K and V. This architecture adaptively fuses global dependencies with local physical connectivity. The calculation process is formulated as follows:

$$\rho_i = CrossAttention(\psi, \varphi, \varphi) = softmax\left(\frac{\psi W_i^Q(\varphi W_i^K)^\top}{\sqrt{d_k}}\right)\varphi W_i^V \tag{45}$$

$$\xi = LN(\psi + MutilheadAttention(\rho_1, \rho_2, \cdots \rho_n)) \tag{46}$$

$$X_s = LN(\xi + FFN(\xi)) \tag{47}$$

Through the fusion of multi-head attention mechanism and residual connection processing, the final output tensor X_s captures both the global dependency relationships of nodes and the strong local correlations with neighboring nodes. This ensures that the model possesses the comprehensive advantages of both global and local information. The combination of global and local features represents a crucial approach for improving the accuracy and robustness of spatio-temporal prediction tasks.

3.4 Hyperparameter optimization

3.4.1 Whale optimization algorithm

The WOA is a metaheuristic algorithm inspired by biological behavior [46]. The inspiration comes from the hunting process of Humpback Whales. WOA is commonly used for multi-objective hyperparameter optimization. The algorithm achieves global optimization by simulating three behaviors of whales: encircling prey, spiral bubble-net attack, and random prey search [47]. WOA features a simple structure with few parameters. It utilizes its global optimization capabilities and efficient search characteristics to automatically explore neural network hyperparameter space. Figure 10 shows the flowchart of using WOA to find optimal model hyperparameters.

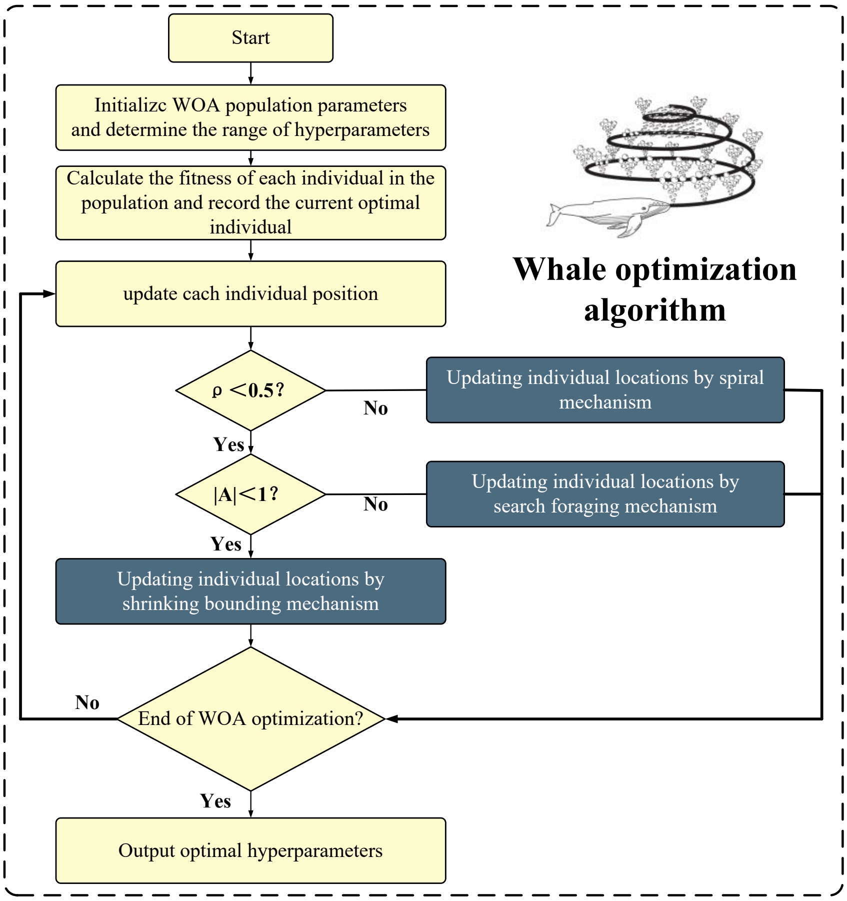

Start
Initializc WOA population parameters and determine the range of hyperparameters
Calculate the fitness of each individual in the population and record the current optimal individual
update cach individual position
ρ <0.5?
No
Updating individual locations by spiral mechanism
Yes
|A|<1?
No
Updating individual locations by search foraging mechanism
Yes
Updating individual locations by shrinking bounding mechanism
No
End of WOA optimization?
Yes
Output optimal hyperparameters
Whale optimization algorithm

Fig. 10 Model hyperparameter optimization algorithm framework

In GL-STFormer model hyperparameter optimization, an initial population containing N_w whales is generated. Each whale's position can be viewed as a vector distributed in d-dimensional parameter space: $x_i = \left(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)}\right)$, where dimension *d* represents the number of hyperparameters to be optimized.

The process of using WOA for hyperparameter optimization is:

Step 1: involves population initialization. In the first round, initial positions $x_i^{(0)}$ of population individuals need to be randomly generated as parameter combinations. The generation rule for whale position $x_i^{(0)}$ follows the formula below:

$$x_i^{(j)} \sim \begin{cases} U(a_j, b_j),\text{If the " j-th " hyperparameter is a consecutive value} \\ R(V_j),\text{if the " j-th" hyperparameter is a discrete value} \end{cases} \tag{48}$$

where a_j, b_j are upper and lower bounds of continuous hyperparameters, and V_j is the candidate set of discrete hyperparameters.

Step 2: updates the convergence factor $\boldsymbol{a(t)}$,

$$a(t) = 2 - \frac{2t}{T} \tag{49}$$

where *t* is the current iteration number and *T* is the maximum iteration number.

Step 3: selects behavior mode.

A random number $\rho \sim U(0,1)$ is generated. When $\rho \geq 0.5$, spiral bubble attack is selected. During hunting, humpback whales form spiral trajectories around prey for fine-grained search of target areas. This process can be simulated through the spiral motion formula:

$$x_i(t+1) = |x^*(t) - x_i(t)| \cdot e^{bl} \cdot \cos(2\pi l) + x^*(t) \tag{50}$$

where $|x^*(t) - x_i(t)|$ represents the distance from whale to target point, b is a shape control coefficient (constant), and l represents random factor $l \sim U(-1,1)$ for generating different trajectories.

When $\rho < 0.5$, encircling or random prey search is selected. Dynamic coefficients $A(t)$ and $C(t)$ are calculated:

$$A(t) = 2a(t)r_1 - a(t) \tag{51}$$

$$C(t) = 2r_2 \tag{52}$$

where $r_1, r_2 \sim U(0,1)$;

When $|A(t)| < 1$, whales perform encircling hunting behavior toward the current best solution x^*, dynamically adjusting current position $x_i(t)$ to approach the target. The formula shows:

$$x_i(t+1) = x^*(t) - A(t) \cdot |C(t) \cdot x^*(t) - x_i(t)| \tag{53}$$

where $x^*(t)$ is the global optimal solution found in iteration t.

To avoid local optima when $|A(t)| \geq 1$, other solutions in the population are randomly selected for exploration. The random search behavior formula shows:

$$x_i(t+1) = x_{rand} - A(t) \cdot |C(t) \cdot x_{rand} - x_i(t)| \tag{54}$$

where x_{rand} represents the position of a randomly selected whale from the population.

3.4.2 Model evaluation metrics

Three commonly used evaluation metrics in regression tasks are employed to scientifically assess the feasibility and accuracy of the model. These metrics are Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Coefficient of Determination (R^2). Their definitions are as follows:

$$MAE = \frac{1}{n}\sum_{k=1}^{n} \left|y_{prediction} - y_{real}\right| \tag{55}$$

$$RMSE = \sqrt{\frac{\sum_{k=1}^{n}\left(y_{prediction} - y_{real}\right)^2}{n}} \tag{56}$$

$$R^2 = 1 - \frac{\sum_{k=1}^{n}\left(y_{prediction} - y_{real}\right)^2}{\sum_{k=1}^{n}\left(y_{prediction} - \bar{y}_{real}\right)^2} \tag{57}$$

where n represents the number of data points, y_{real} represents historical traffic flow data, and $y_{prediction}$ represents predicted traffic volume data. Lower values of $RMSE$ and MAE indicate better prediction performance of the model. A higher R^2 value indicates better model fitting, thus higher prediction accuracy.

4. Experiments

4.1 Data description

This section validates the feasibility and performance of the proposed prediction framework using real AIS data. Data transmission may experience loss or errors, causing historical vessel trajectories to deviate from actual conditions. Direct use of raw AIS data for prediction would result in significant errors. Therefore, we reference previous AIS data preprocessing methods to improve vessel trajectory quality [48]. The processed AIS data issued to collect historical vessel flow data at selected cross-sections of the Nantong section of the Yangtze River. Data is collected from September 30, 2021 to December 31, 2021, with a temporal granularity of 15 minutes. CEEMDAN decomposed the vessel traffic flow into 8 IMF components. The dataset was divided into three parts: 70% for training, 20% for validation, and 10% for testing. A sliding window approach was used to generate time series datasets. A sliding window approach is employed to construct the time-series dataset. To achieve prediction horizons of 15, 30, and 60 minutes, the model adopts a multi-step prediction strategy, directly outputting the vessel traffic flow for the corresponding future timestamps.

4.2 Hyperparameter optimization using WOA

In deep learning-based vessel traffic flow prediction for the Nantong section of the Yangtze River, model performance heavily depends on hyperparameter selection. Traditional manual parameter tuning relies too much on experience and requires extensive trials. It is time-consuming, labor-intensive, subjective, and prone to local optima. Therefore, this paper employs the previously introduced WOA to find optimal model hyperparameters.

The hyperparameters selected for optimization include: learning rate, dropout rate, batch size, input time steps, temporal and spatial feature dimensions, the number of GRU layers, the number of Transformer encoder/decoder layers, and the number of attention heads.

In the optimization process, the population size is set to 40, representing 40 candidate hyperparameter combinations. Each combination was trained for 100 epochs with 10 iterations. The final output results are shown in Table 2.

Table 2 Optimal model hyperparameters obtained by WOA

Hyperparameters	Set range	Optimal result
Learning rate	[0.0001,0.01]	0.00091
Dropout rate	[0,0.5]	0.10409
Temporal feature dimensions	[32, 64, 128, 256]	128
Spatial feature dimensions	[24, 48, 96, 192]	48
Batch size	[16, 32, 64, 128]	32
Input time steps	[8, 14]	12
GRU layers	[1, 2, 3, 4]	2
Transformer encoder/decoder layers	[1, 2, 3, 4]	1
Attention heads	[2, 4, 6, 8]	8

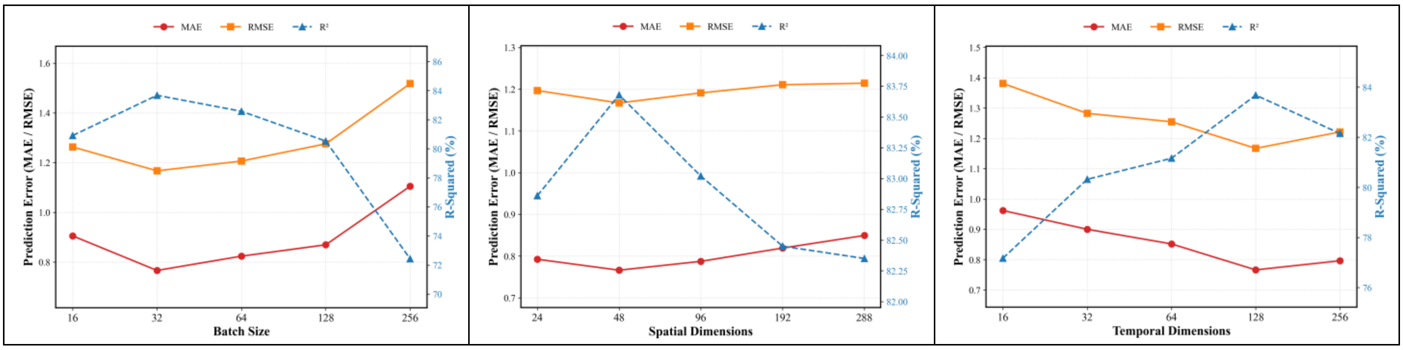


Fig. 11 (a) Effect of different batch sizes on model performance, (b) Effect of different spatial dimensions on model performance, (c) Effect of different temporal dimensions on model performance

To validate the effectiveness of the hyperparameter combination identified by the WOA, a sensitivity analysis was conducted by varying batch sizes, temporal dimensions, and spatial dimensions to evaluate prediction performance. As illustrated in Figure 11, the optimal performance across all metrics is achieved with the parameters determined by WOA. These results confirm the validity of the selected hyperparameter combination.

4.3 Experimental setup

All experiments in this chapter were conducted in the same experimental environment, as shown in Table 3. MAE was used as the model's loss function, with parameters continuously updated through backpropagation for model training. The Adam optimizer was used to assist the loss function in approaching global minimum. This reduced the loss value in both training and validation sets. Through multiple optimization experiments, the optimal model parameters from the validation set were saved and used to output final prediction results on the test set.

Table 3 Experimental facilities

Device Name	Configuration/Version
GPU	Intel Core i5-13490F (2.5 GHz)
Graphics card	NVIDIA GeForce GTX 1660 SUPER
Memory	32 GB RAM
Operating system	Windows11 64
Python version	Python 3.10
Pytorch version	Pytorch 2.1

4.4 Experimental result

The final experiment was conducted using the optimal hyperparameters determined by the WOA. Figure 12 shows the training and validation losses. The model gradually stabilized after 50 epochs without overfitting or underfitting.

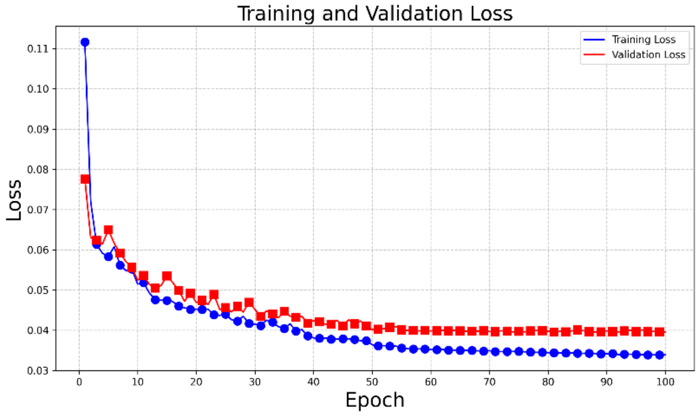
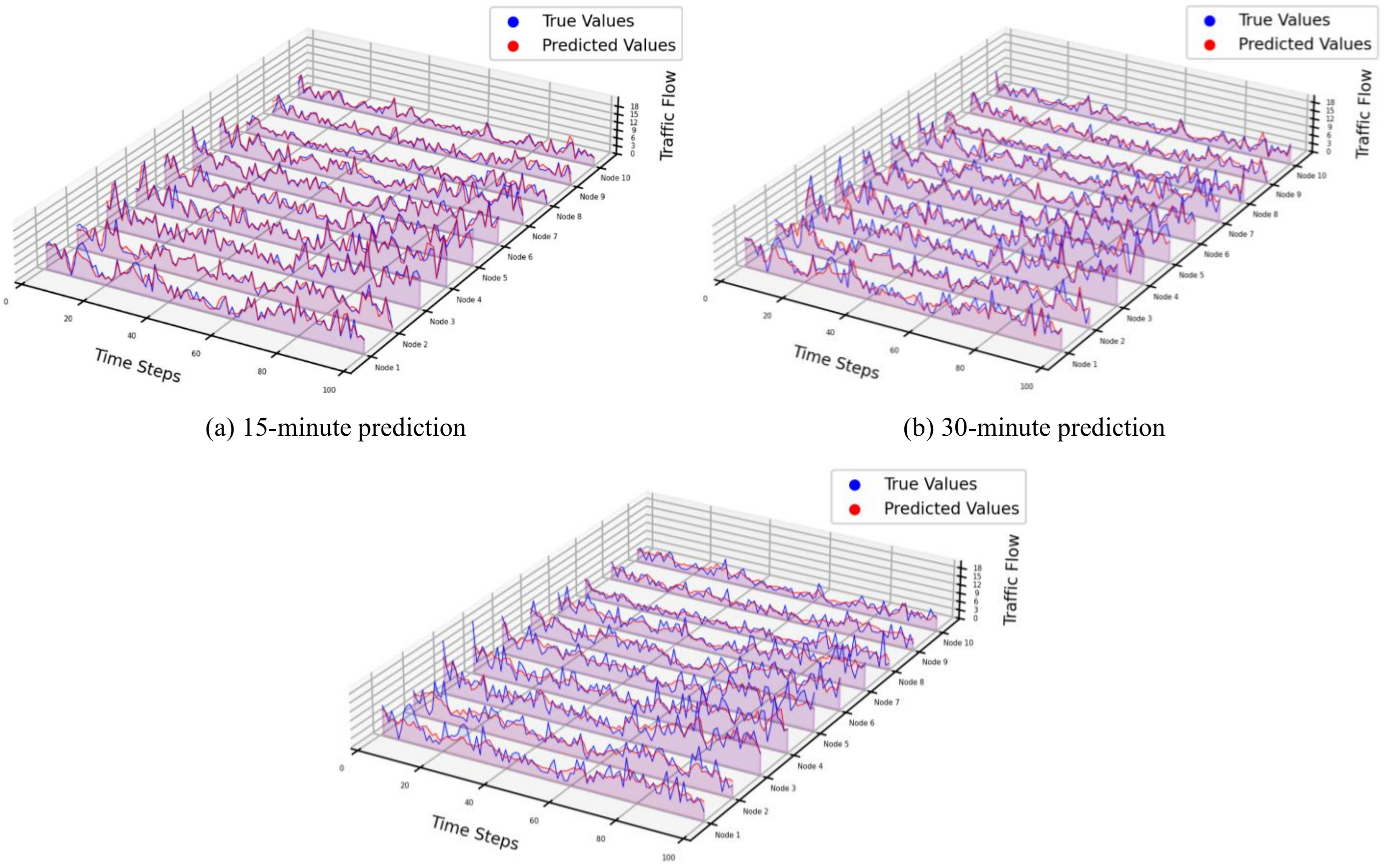


Fig. 12 Model training and validation loss curves

Figure 13 presents the multi-step prediction results of GL-STFormer on the test set. As shown in Figure 13(a), the predicted values fit well with the ground truth for the 15-minute horizon. As illustrated in Figures 13(b) and 13(c), the prediction deviation increases as the horizon extends to 30 and 60 minutes. Despite the performance decline in long-term forecasting, the evolutionary trends of vessel traffic flow are captured. These results validate the effectiveness of GL-STFormer in multi-step scenarios.



(a) 15-minute prediction

(b) 30-minute prediction

(c) 60-minute prediction

Fig. 13 Visualization of multi-step vessel flow prediction results on the test set

To intuitively observe the spatial feature capturing ability of GL-STFormer, we randomly selected a test sample. We visualized its global and local spatial attention weight matrices at a specific time step, as shown in Figure14. The left panel displays the global spatial attention matrix extracted by the encoder. The right panel shows the local spatial attention matrix extracted by the decoder. By assigning different attention weights, these matrices adaptively adjust the importance of the input information. Therefore, the GL-STFormer model can effectively capture complex spatial dependencies among the data.

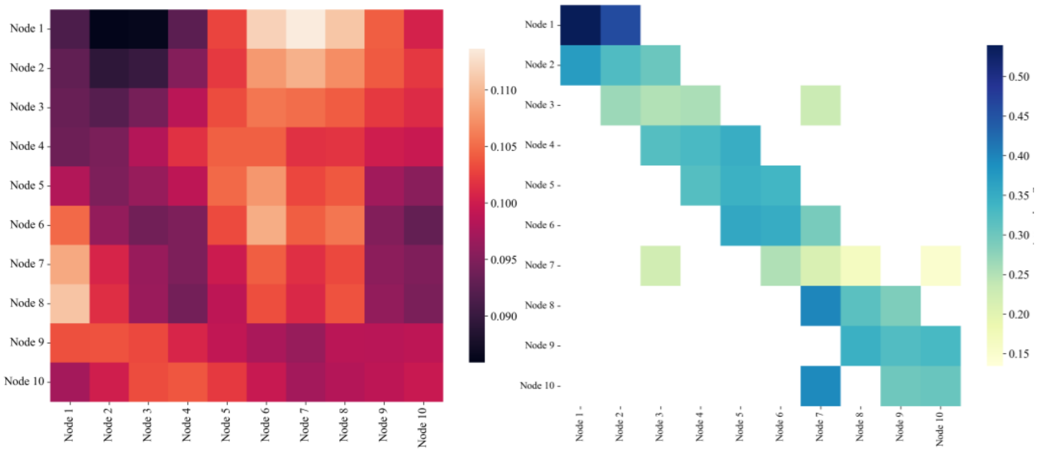


Fig. 14 Visualization of global and local spatial attention weight matrices of the GL-STFormer model

4.5 Comparative experiment

To rigorously evaluate the effectiveness and accuracy of the proposed model, comparative experiments were conducted against a variety of classical machine learning and deep learning models. These baselines include Support Vector Regression (SVR), Autoregressive Integrated Moving Average (ARIMA), GCN, GRU, LSTM, Bi-LSTM, Spatial-Temporal Graph Convolutional Network (STGCN), and LSTM-GAT. It is noteworthy that both STGCN and LSTM-GAT were integrated with the CEEMDAN decomposition algorithm to ensure a consistent experimental basis. To guarantee a fair evaluation, default parameters from the original implementations were adopted for all baseline methods.

As presented in Table 4, the baseline models demonstrate limited capability in capturing the complex nonlinear patterns of vessel traffic flow. In contrast, the proposed GL-STFormer exhibits superior predictive performance, validating the effectiveness of the model. Specifically, GL-STFormer achieves lower prediction errors compared to both STGCN and LSTM-GAT. This performance disparity can be attributed to the differences in their spatial modeling mechanisms. STGCN relies on a pre-defined static adjacency matrix to extract spatial features, which limits its ability to capture time-varying spatial dependencies. Furthermore, although LSTM-GAT incorporates a dynamic attention mechanism, it primarily focuses on aggregating information from connected neighbors, thereby capturing only local spatial correlations. In contrast, the Transformer architecture in GL-STFormer utilizes the encoder-decoder structure to simultaneously capture and fuse both global and local spatial dependencies. This capability allows the model to learn long-range interaction features beyond physical connectivity constraints, thereby achieving superior results in the experiments.

Table 4 Comparison of prediction accuracy of different models (unit: vessels)

Model	15min			30min			60min		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
ARIMA	1.978	2.591	0.196	1.998	2.632	0.170	2.138	2.817	0.049
SVR	2.057	2.626	0.174	2.084	2.661	0.152	2.108	2.692	0.132
GCN	1.912	2.489	0.258	1.920	2.506	0.248	1.981	2.576	0.205
GRU	1.867	2.445	0.283	1.905	2.483	0.262	1.916	2.521	0.237
LSTM	1.871	2.450	0.281	1.882	2.487	0.259	1.909	2.534	0.230
Bi-LSTM	1.868	2.451	0.280	1.907	2.488	0.258	1.914	2.518	0.240
STGCN	1.406	1.840	0.594	1.523	2.001	0.520	1.605	2.117	0.463
LSTM-GAT	0.831	1.232	0.818	1.095	1.528	0.720	1.327	1.781	0.620
GL-STFormer	**0.766**	**1.167**	**0.836**	**1.021**	**1.466**	**0.742**	**1.268**	**1.742**	**0.637**

4.6 Ablation experiment

To verify the contribution of different components to vessel traffic flow prediction, we conducted ablation studies by removing certain modules. Four ablation versions were constructed: w/o CEEMDAN(removes the signal decomposition module), w/o GRU(removes the GRU layer), w/o Self-attention(removes the self-attention mechanism), and w/o Transformer(removes the spatial modeling module). The models were quantitatively evaluated using MAE, RMSE, and R^2. The results are shown in Table 5.

The ablation study results are presented in Table 5. Notably, after removing the CEEMDAN module, the model accuracy dropped sharply. Specifically, for the 15-minute horizon, the MAE increased from 0.766 to 1.816. This indicates that using the original data directly into the model without decomposition leads to inferior prediction performance. This is due to the multi-scale non-stationarity and complex fluctuation characteristics of the raw data. Furthermore, performance degradation is observed in variants without the GRU and Transformer modules. This validates the necessity of temporal and spatial feature extraction. Overall, the proposed model can effectively capture the spatiotemporal characteristics of vessel traffic flow data.

Table 5 Comparison of ablation model prediction accuracy (unit: vessels)

Model	15min			30min			60min		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
w/o CEEMDAN	1.816	2.396	0.313	1.831	2.417	0.301	1.848	2.460	0.276
w/o GRU	1.489	1.945	0.547	1.618	2.122	0.461	1.715	2.252	0.393
w/o Self-attention	0.796	1.178	0.823	1.053	1.487	0.733	1.290	1.762	0.622
w/o Transformer	0.928	1.326	0.789	1.187	1.605	0.691	1.449	1.932	0.553
GL-STFormer	**0.766**	**1.167**	**0.836**	**1.021**	**1.466**	**0.742**	**1.268**	**1.742**	**0.637**

To verify the effectiveness of the CEEMDAN decomposition module, this study compares it with two classical algorithms: EMD and EEMD. The results are illustrated in Figure 15. The CEEMDAN-based model achieves superior performance across all metrics compared to the other decomposition methods. In contrast, EMD yields larger errors due to mode mixing. While EEMD improves the goodness of fit relative to EMD, its overall prediction accuracy remains inferior to the proposed CEEMDAN approach. These results indicate that CEEMDAN possesses significant advantages in extracting features from complex vessel traffic flow.

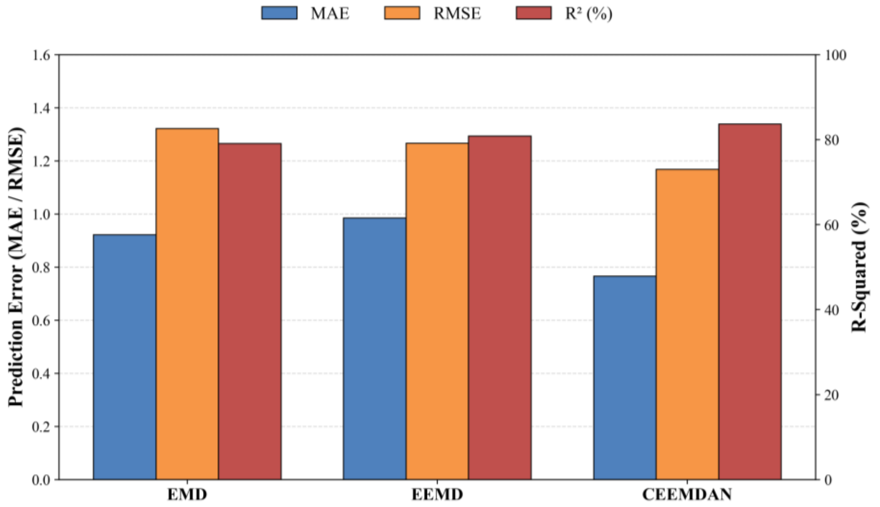


Fig. 15 Performance comparison of different signal decomposition strategies.

5. Conclusions

The study developed a spatiotemporal prediction model based on signal decomposition and hybrid deep learning to address complex inland vessel traffic flow prediction. The model framework integrates CEEMDAN, GRU, self-attention mechanism, Transformer, and WOA. Initially, the CEEMDAN algorithm decomposes the original vessel flow sequence into multiple IMFs, significantly reducing data complexity. During feature extraction, GRU networks capture long-term dependencies in flow sequences, while self-attention mechanism enhances feature weights of key time steps. Additionally, combining inland traffic networks, the Transformer encoder-decoder architecture establishes cross-waterway spatial dependencies, achieving collaborative perception between local channels and global networks through multi-head attention mechanism. To overcome hyperparameter sensitivity, the WOA globally optimizes model hyperparameters to output final predictions using optimal parameters.

Training was conducted using real historical AIS data from the Yangtze River. Experimental results demonstrate that the proposed model outperforms traditional baseline models in MAE, RMSE, and R², validating its effectiveness and superiority in capturing complex spatiotemporal correlations.

Future research could explore model performance over longer prediction horizons. Furthermore, integrating multiple data types including AIS, meteorological data, and vessel motion data could provide more detailed learning information, enhancing the robustness of deep learning methods and improving modeling capabilities for complex waterway traffic characteristics. These extensions could enable the model to play a greater role in improving inland shipping safety and transport efficiency, while opening new pathways for intelligent shipping and maritime traffic management systems development.

ACKNOWLEDGEMENT

This research was funded by the National Natural Science Foundation of China (Grant No. 52501441 and 52301401) and the Fundamental Research Project (JCKY2023206A023).

REFERENCES

[1] Welch, K., Lambert, L. H., Lambert, D. M., Kenkel, P., 2022. Flood-induced disruption of an inland waterway transportation system and regional economic impacts. *Water*, 14(5), 753. https://doi.org/10.3390/w14050753

[2] Ma, Q., Wang, Z., Zhou, T., Liu, Z., 2024. Robust optimization method of emergency resource allocation for risk management in inland waterways. *Brodogradnja*, 75(1), 1-22. https://doi.org/10.21278/brod75103

[3] Chai, T., Weng, J., Li, G., 2020. Estimation of vessel collision frequency in the Yangtze river estuary considering dynamic ship domains. *Journal of Marine Science and Technology*, 25(3), 964-77. https://doi.org/10.1007/s00773-019-00693-6

[4] Ma, Q., Tang, H., Liu, C., Zhang, M., Zhang, D., Liu, Z., Zhang, L., 2024 A big data analytics method for the evaluation of maritime traffic safety using automatic identification system data. *Ocean & Coastal Management*, 251, 107077. https://doi.org/10.1016/j.ocecoaman.2024.107077

[5] Liu, C., Suominen, M., Musharraf, M., 2025. An ensemble machine learning model for predicting the need for icebreaker assistance in ice-covered waters. *Engineering Applications of Artificial Intelligence*, 158, 111489. https://doi.org/10.1016/j.engappai.2025.111489

[6] Ma, Q., Pan, J., Zhou, Y., Zhou, S., Zhang, M., 2025. Optimization of maritime emergency base placement for inland waterway accident response: a case study of the Yangtze River. *Brodogradnja*, 76(1), 1-24. https://doi.org/10.21278/brod76107

[7] Yan, X., Wan, C., Zhang, D., Yang, Z., 2017. Safety management of waterway congestions under dynamic risk conditions-A case study of the Yangtze River. *Applied Soft Computing*, 59, 115-28. https://doi.org/10.1016/j.asoc.2017.05.053

[8] Zhang, M., Zhang, D., Fu, S., Kujala, P., Hirdaris, S., 2022. A predictive analytics method for maritime traffic flow complexity estimation in inland waterways. *Reliability Engineering & System Safety*, 220, 108317. https://doi.org/10.1016/j.ress.2021.108317

[9] Eski, Ö., Tavacioglu, L., 2024. A combined method for the evaluation of contributing factors to maritime dangerous goods transport accidents. *Brodogradnja*, 75(4), 1-20. https://doi.org/10.21278/brod75408

[10] Sui, Z., Wang, S., Wen, Y., Cheng, X., Theotokatos, G., 2024. Multi-state ship traffic flow analysis using data-driven method and visibility graph. *Ocean Engineering*, 298, 117087. https://doi.org/10.1016/j.oceaneng.2024.117087

[11] Cui, Y., Chen, Y., Chen, Y., Cai, X., Yin, C., Cheng, Y., 2025. Spatial-temporal quantification of Yangtze River traffic flow using AIS data. *Ocean Engineering*, 318, 120144. https://doi.org/10.1016/j.oceaneng.2024.120144

[12] Xiao, F., Ligteringen, H., Van Gulijk, C., Ale, B., 2025. Comparison study on AIS data of ship traffic behavior. O*cean Engineering*, 95, 84-93. https://doi.org/10.1016/j.oceaneng.2014.11.020

[13] Svanberg, M., Santén, V., Hörteborn, A., Holm, H., Finnsgård, C., 2019. AIS in maritime research. *Marine Policy*, 106, 103520. https://doi.org/10.1016/j.marpol.2019.103520

[14] Kumar, S. V., Vanajakshi, L., 2015. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *European Transport Research Review*, 7, 1-9. https://doi.org/10.1007/s12544-015-0170-8

[15] He, W., Zhong, C., Sotelo, M. A., Chu, X, Liu, X., Li, Z., 2019. Short-term vessel traffic flow forecasting by using an improved Kalman model. *Cluster Computing*, 22, 7907-7916. https://doi.org/10.1007/s10586-017-1491-2

[16] Wang, M., Cui, Y., Wang, X., Xiao, S., Jiang, J., 2017. Machine learning for networking: Workflow, advances and opportunities. *IEEE Network*, 32(2), 92-99. https://doi.org/10.1109/MNET.2017.1700200

[17] Duo, M., Qi, Y., Lina, G., Xu, E., 2017. A short-term traffic flow prediction model based on EMD and GPSO-SVM. *Proceedings of the 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 26-26 March, Chongqing, China, 2554-2558. https://doi.org/10.1109/IAEAC.2017.8054485

[18] Zhang, L., Alharbe, N. R., Luo, G., Yao, Z., Li, Y., 2018. A hybrid forecasting framework based on support vector regression with a modified genetic algorithm and a random forest for traffic flow prediction. *Tsinghua Science and Technology*, 23(4), 479-492. https://doi.org/10.26599/TST.2018.9010045

[19] Aouedi, O., Le, V. A., Piamrat, K., Ji, Y., 2025. Deep learning on network traffic prediction: Recent advances, analysis, and future directions. *ACM Computing Surveys*, 57(6), 1-37. https://doi.org/10.1145/3703447

[20] Tian, Y., Zhang, K., Li, J., Lin, X., Yang, B., 2018. LSTM-based traffic flow prediction with missing data. *Neurocomputing*, 318, 297-305. https://doi.org/10.1016/j.neucom.2018.08.067

[21] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[22] Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:14123555.

[23] Cao, Y., Zhang, J., Ma, A., Xu, H., Liu, J., 2026. Marine engine cylinder exhaust temperature prediction based on PSO-optimized CNN-LSTM-attention network. *Brodogradnja*, 77(1), 77101. https://doi.org/10.21278/brod77101

[24] Zhang, Z.-G., Yin, J.-C., Wang, N.-N., Hui, Z.-G., 2019. Vessel traffic flow analysis and prediction by an improved PSO-BP mechanism based on AIS data. *Evolving Systems*, 10, 397-407. https://doi.org/10.1007/s12530-018-9243-y

[25] Dong, Z., Zhou, Y., Bao, X., 2024. A short-term vessel traffic flow prediction based on a DBO-LSTM model. *Sustainability*, 16(13), 5499. https://doi.org/10.3390/su16135499

[26] Chang, Y., Ma, J., Sun, L., Ma, Z., Zhou, Y., 2024. Vessel Traffic Flow Prediction in Port Waterways Based on POA-CNN-BiGRU Model. J*ournal of Marine Science and Engineering*, 12(11), 2091. https://doi.org/10.3390/jmse12112091

[27] Man, J., Chen, D., Wu, B., Wan, C., Yan, X., 2024. An effective approach for Yangtze river vessel traffic flow forecasting: A case study of Wuhan area. *Ocean Engineering*, 296, 116899. https://doi.org/10.1016/j.oceaneng.2024.116899

[28] 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 Transactions on Intelligent Transportation Systems*, 23(12), 23694-23707. https://doi.org/10.1109/TITS.2022.3199160

[29] Ma, Q., Du, X., Zhang, M., Wang, H., Lang, X., Mao, W., 2024. A spatial-temporal attention method for the prediction of multi ship time headways using AIS data. *Ocean Engineering*, 311, 118927. https://doi.org/10.1016/j.oceaneng.2024.118927

[30] Li, L., Pan, M., Liu, Z., Sun, H., Zhang, R., 2024. Semi-dynamic spatial–temporal graph neural network for traffic state prediction in waterways. *Ocean Engineering*, 293, 116685. https://doi.org/10.1016/j.oceaneng.2024.116685

[31] Torres, M. E., Colominas, M. A., Schlotthauer, G., Flandrin, P., 2011. A complete ensemble empirical mode decomposition with adaptive noise. *Proceedings of the 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 4144-4147. https://doi.org/10.1109/ICASSP.2011.5947265

[32] Huan, Y., Kang, X., Zhang, Z., Zhang, Q., Wang, Y., Wang, Y., 2022. AIS-based vessel traffic flow prediction using combined EMD-LSTM method. *Proceedings of the Proceedings of the 4th International Conference on Advanced Information Science and System*, 45, 1-8. https://doi.org/10.1145/3573834.3574517

[33] Wang, W., Zheng, Q., Li, R., Zhang, C., Tian, Y., Zhang, Y., 2024. Ship Traffic Flow Prediction Using EEMD-edRVFL-RF. *Proceedings of the 2024 International Conference on Automation in Manufacturing, Transportation and Logistics (ICaMaL)*, 1-7. https://doi.org/10.1109/ICaMaL62577.2024.10919820

[34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

[35] 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. *Engineering Applications of Artificial Intelligence*, 133, 108172. https://doi.org/10.1016/j.engappai.2024.108172

[36] Fu, R., Zhang, Z., Li, L., 2016. Using LSTM and GRU neural network methods for traffic flow prediction. *Proceedings of the 2016 31st Youth academic annual conference of Chinese association of automation (YAC)*, 324-328. IEEE. https://doi.org/10.1109/YAC.2016.7804912

[37] He, Z., Jiang, T., Jiang, Y., Luo, Q., Chen, S., Gong, K., He, L., Feng, H., Yu, Q., Tan, F., 2022. Gated recurrent unit models outperform other machine learning models in prediction of minimum temperature in greenhouse based on local weather data. *Computers and Electronics in Agriculture*, 202, 107416. https://doi.org/10.1016/j.compag.2022.107416

[38] Chen, C., Xue, L., Xing, W., 2023. Research on improved GRU-based stock price prediction method. *Applied Sciences*, 13(15), 8813. https://doi.org/10.3390/app13158813

[39] Lin, Z., Yue, W., Huang, J., Wan, J., 2023. Ship trajectory prediction based on the TTCN-attention-GRU model. *Electronics*, 12(12), 2556. https://doi.org/10.3390/electronics12122556

[40] Ran, X., Shan, Z., Fang, Y., Lin, C., 2019. An LSTM-based method with attention mechanism for travel time prediction. *Sensors*, 19(4), 861. https://doi.org/10.3390/s19040861

[41] Wang, K., Ma, C., Qiao, Y., Lu, X., Hao, W., Dong, S., 2021. A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction. *Physica A: Statistical Mechanics and its Applications*, 583, 126293. https://doi.org/10.1016/j.physa.2021.126293

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

[43] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:171010903.

[44] Yan, H., Ma, X., Pu, Z., 2021. Learning dynamic and hierarchical traffic spatiotemporal features with transformer. *IEEE Transactions on Intelligent Transportation Systems*, 23(11), 22386-99. https://doi.org/10.1109/TITS.2021.3102983

[45] Huo, G., Zhang, Y., Wang, B., Gao, J., Hu, Y., Yin, B., 2023. Hierarchical spatio–temporal graph convolutional networks and transformer network for traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 24(4), 3855-3867. https://doi.org/10.1109/TITS.2023.3234512

[46] Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm. *Advances in engineering software*, 95, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[47] Xie, H., Ding, R., Qiao, G., Dai, C., Bai, W., 2024. Research on Ship Traffic Flow Prediction using CNN-BIGRU and WOA with Multi-Objective Optimization. *IEEE Access*, 12, 138372-138385. https://doi.org/10.1109/ACCESS.2024.3466527

[48] Ma, Q., Lian, Z., Du, X., Jiang, Y., BahooToroody, A., Zhang, M., 2025. A deep learning method to predict ship short-term trajectory for proactive maritime traffic management. *Reliability Engineering & System Safety*, 265, 111542. https://doi.org/10.1016/j.ress.2025.111542