Li, H, Zhang, Y, Li, Y, Lam, JSL, Matthews, C and Yang, Z

Deep multi-view information-powered vessel traffic flow prediction for intelligent transportation management

Transportation Research Part E 197 (2025) 104072

Contents lists available at ScienceDirect

Transportation Research Part E

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

Deep multi-view information-powered vessel traffic flow prediction for intelligent transportation management

Huanhuan Li [a], Yu Zhang [b], Yan Li [b,*], Jasmine Siu Lee Lam [c], Christian Matthews [d], Zaili Yang [a,*]

[a] Liverpool Logistics, Offshore and Marine (LOOM) Research Institute, Liverpool John Moores University, Liverpool, UK
[b] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China
[c] Department of Technology, Management and Economics, Technical University of Denmark, Denmark
[d] Department of Maritime and Mechanical Engineering, Liverpool John Moores University, Liverpool, UK

ARTICLE INFO

Keywords:
Vessel traffic flow prediction
Intelligent transportation management
Automatic Identification System (AIS)
Maritime transportation
Port planning

ABSTRACT

Vessel traffic flow (VTF) prediction, essential for intelligent transportation management, is derived from the statistical analysis of longitude and latitude information from Automatic Identification System (AIS) data. Traditional deep learning approaches have struggled to effectively capture the intricate and dynamic characteristics inherent in VTF data. To address these challenges, this paper proposes a new prediction model called a Multi-view Periodic-Temporal Network with Semantic Representation (i.e., MPTNSR), which leverages three perspectives: periodic, temporal, and semantic. VTF typically conceals the periodic and temporal characteristics during its evolution. A Convolutional Neural Network and Bidirectional Long Short-Term Memory (CNN-BiLSTM) model, constructed from periodic and temporal views, effectively captures this information. However, real-world scenarios frequently involve predicting VTF for multiple target regions simultaneously, where correlations between VTF changes in different areas are significant. The semantic view seeks to extract relationships across different channels based on the similarity of VTF data fluctuations and geographical distribution across regions, utilising a Graph Convolutional Network (GCN). The final prediction result is generated by fusing the information from these three views. Additionally, an optimised loss function is developed in the MPTNSR model that integrates local and global measurement information. In summary, the proposed model combines the strengths of a multi-view learning network and an optimised loss function. Quantitative comparative experiments demonstrate that the MPTNSR model outperforms eighteen state-of-the-art methods in VTF prediction tasks. To enhance the model's scalability, Graphics Processing Unit (GPU)-accelerated computation is introduced, significantly improving its efficiency and reducing its running time. The model enables accurate and robust prediction, effectively assisting in port planning and waterway management, thereby enhancing the safety and sustainability of maritime transportation.

1. Introduction

The ever-expanding volume of global maritime trade, coupled with the increasing emphasis on efficient and safe port operations,

* First corresponding author: Y. Li. Second corresponding author: Z. Yang.
E-mail addresses: liyanWHU@whu.edu.cn (Y. Li), z.yang@ljmu.ac.uk (Z. Yang).

https://doi.org/10.1016/j.tre.2025.104072
Received 28 July 2024; Received in revised form 4 March 2025; Accepted 4 March 2025
Available online 20 March 2025

has brought significant attention to Vessel Traffic Flow (VTF) prediction in intelligent transportation systems in recent years (Li et al., 2024a; Yang et al., 2023). The dynamic and intricate nature of maritime traffic, along with the complexity of waterways and port infrastructures, presents substantial challenges in predicting VTF accurately but underscores its essential role in managing vessel traffic effectively (Xin et al., 2023; Yang et al., 2024). The emergence of VTF prediction models is closely related to advances in Machine Learning (ML), Deep Learning (DL), and Artificial Intelligence (AI). VTF prediction involves forecasting vessel movements within waterways and ports, enabling early hazard detection and ensuring safe vessel navigation (C. Wang et al., 2024; Q. Wang et al., 2024). Precise VTF prediction is indispensable for enhancing the safety, efficiency, and economy of maritime operations, benefiting port operators, shipping companies, and stakeholders alike (Li et al., 2024b; Xiao et al., 2020).

Recent years have seen significant advancements in VTF prediction models, especially with the integration of advanced sensor technologies like Automatic Identification Systems (AIS), Radio Detection and Ranging (Radar), Light Detection and Ranging (Lidar), and optical imaging (Liu et al., 2025; Shu et al., 2025; Xin et al., 2024; Yu et al., 2022). These technologies enable real-time vessel tracking, bolstering the reliability of VTF prediction models and facilitating proactive traffic management (Zissis et al., 2016). The maritime transportation network, spanning both open sea and port waters, as depicted in Fig. 1, highlights the importance of the space-ground network in maritime navigation infrastructure and emphasises the pivotal role of VTF in ensuring local navigational safety. AIS data, including vessel position, speed, and heading information, has been used to improve VTF prediction accuracy, serving as a foundation for future advancements. VTF prediction is also crucial in the context of intelligent port design and the realisation of Maritime Autonomous Surface Ships (MASS) (Li et al., 2023a).

VTF prediction is a constantly evolving field with a strong focus on improving prediction accuracy and reliability. Researchers mainly use two types of methods: modeling-based and learning-based approaches (Han et al., 2021). Modeling-based methods, based on traditional ML techniques, have limitations in capturing complex and irregular patterns (Lv et al., 2023). In contrast, learning-based methods using Neural Networks (NN) and DL are gaining traction for effectively handling these intricate patterns (Kumar et al., 2023; Yan et al., 2023). Convolutional Neural Networks (CNNs) are widely applied in maritime transportation networks (Liu et al., 2021). Recurrent Neural Networks (RNNs) play a vital role in time series prediction. While they offer powerful capabilities, they face challenges like gradient issues during training (Ma and Mei, 2022; Turkoglu et al., 2022). To overcome these, Long Short-Term Memory (LSTM) and Gate Recurrent Unit (GRU) are popular, especially for time series prediction with complex dependencies (Hua et al., 2019; Weerakody et al., 2021). Bidirectional LSTM (Bi-LSTM) and Bidirectional GRU (Bi-GRU) models, capturing dependencies in both directions, often lead to improved VTF prediction (Li et al., 2024b). The model, called CNN-LSTM, combines CNN and LSTM for time series prediction, leveraging the strengths of both algorithms (Livieris et al., 2020; Zha et al., 2022).

Although demonstrating their effectiveness in terms of temporal features, LSTM, GRU, Bi-LSTM, and Bi-GRU are still limited in identifying spatial and spatial–temporal features of VTF. Moreover, the mutual interaction effects between channels and semantic features in the same areas pose another challenge. Furthermore, the periodic information of VTF data is frequently overlooked, including the information between adjacent time periods on the same day, as well as the connection between the same time period on different days. To address the abovementioned limitations, this paper makes new contributions below.

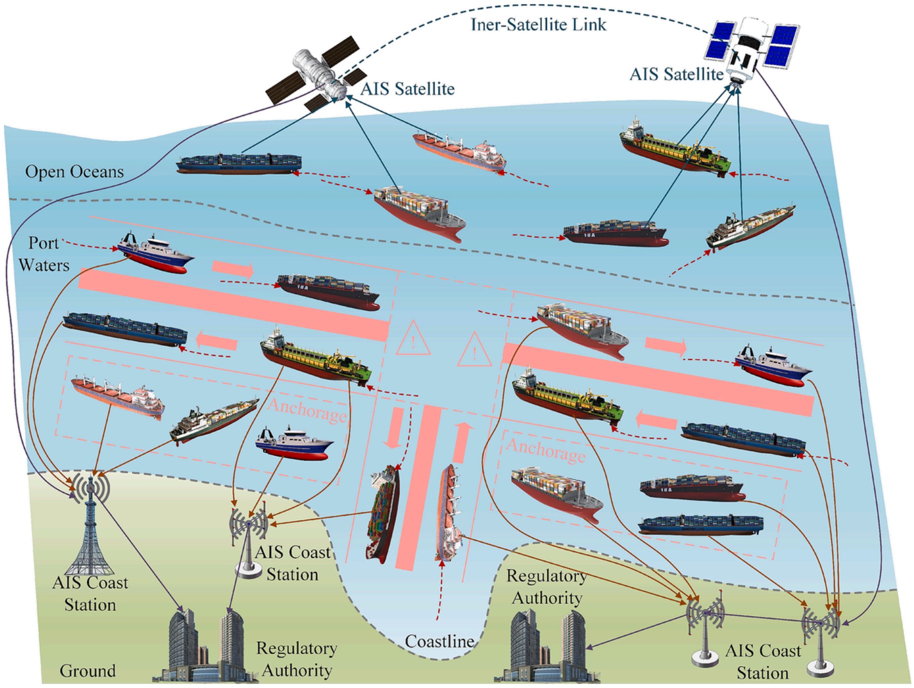


Fig. 1. The comprehensive maritime transportation network.

(1) Develop a novel hierarchical methodology called the Multi-view Periodic-Temporal Network with Semantic Representation (i.e., MPTNSR) for solving the collaborative multi-channel VTF prediction using three key perspectives: periodic, temporal, and semantic.
(2) Enhance the extraction of periodic and temporal features of VTF data by developing a CNN-BiLSTM-based network.
(3) Realise collaborative prediction using a Graph Convolutional Network (GCN) among multiple channels by combining geographical features with the similarity of the traffic flow changes.
(4) Design a novel loss function by integrating Mean Squared Error (MSE) and Pearson Correlation Coefficient (PCC) from both global and local perspectives.

To enhance the scalability and practicality of the model, Graphics Processing Unit (GPU)-accelerated computation has been implemented, bringing substantial improvements in efficiency and significantly reducing running time. GPUs are particularly well-suited for handling the high computational demands of complex models (Li et al., 2024c) like the proposed MPTNSR, which integrates periodic, temporal, and semantic features. By significantly increasing computation speed, GPUs enable real-time processing, making the model practical for dynamic and time-sensitive applications. This capability ensures that the MPTNSR model can deliver accurate predictions promptly, meeting the critical demands of real-time decision-making in maritime traffic systems. The adoption of GPU-based computation underscores the importance of technological integration in optimising advanced models for practical and dynamic environments.

The paper is organised as follows: A systematic literature review on VTF prediction is conducted in Section 2, listing a detailed analysis of various models in VTF research content. Facilitated by the research problems in this paper, the preliminary knowledge about VTF prediction is described in Section 3. Section 4 elaborates on the proposed methodology from spatial, temporal, and spatio-temporal views. The experimental studies are implemented in two real water areas to verify the effectiveness and robustness of the proposed methodology in VTF prediction in Section 5. Section 6 includes the conclusion and future research directions.

2. Literature review

VTF prediction is a hot topic in intelligent maritime traffic management. Especially with the extensive use of AIS data, researchers have been able to determine the VTF for specific areas by using key data points, such as time stamp, latitude and longitude coordinates, course over ground, and speed over ground. This has led to a surge in data-driven prediction methods on AIS data. This section provides an overview of VTF prediction methods using ML and DL prediction methods, respectively, and analyses their advantages and existing problems in detail. Furthermore, the identified research gaps are outlined in Section 2.3.

2.1. *Machine learning methods in vessel traffic flow prediction*

Traditional ML methods, also called statistical methods, are widely used in the initial phases of VTF prediction research. These methods assume a specific relationship between historical and predicted data. Nevertheless, it is crucial to highlight that VTF data typically displays periodic changes over time. As a result, various ML methods have been developed to estimate parameters and fit curves, enabling accurate prediction of VTF data. Various classical ML methods are employed for predicting VTF data, including linear regression, variance analysis models, Kalman Filtering (KF), Grey Model (GM), Markov Model (MM), Kernel Density Estimation (KDE), AutoRegressive Integrated Moving Average (ARIMA), and Support Vector Machine (SVM) (Li and Yang, 2023; Ryu et al., 2018). Different models have their own characteristics, advantages, and disadvantages for VTF prediction. Linear regression (Sousa et al., 2007) and variance analysis (Makowski et al., 2006) models struggle with non-stationary time series data, while KF (Muruganantham et al., 2016) excels in estimating future trends but faces challenges with irregular historical data. GM (Kayacan et al., 2010) proficiently recognises diverse developmental trends but struggles in the face of frequent data mutations. MM model (Zou et al., 2022) predicts event probabilities skillfully but may encounter accuracy issues. KDE (Xiao et al., 2017) captures distribution characteristics but may struggle with non-periodic changes. ARIMA (Sadeghi Gargari et al., 2022) forecasts traffic by combining past and present data linearly. SVM (Zhang and Wu, 2022) handles both linear and non-linear classification tasks but may be inefficient for large VTF datasets. The advantages and disadvantages of classical ML methods for VTF prediction are listed in Table 1.

The abovementioned ML methods are classical and commonly used, suitable for solving time series prediction problems. However, VTF data is constantly affected by external factors (e.g., vessel collision accidents, severe weather, and emergency events like COVID-19), which can cause significant fluctuations, making it difficult for these original ML methods to fit complex change features, thereby affecting prediction accuracy. Therefore, many scholars have optimised these traditional ML methods to improve prediction accuracy.

Upon analysis, it becomes apparent that these ML methods have limitations in VTF prediction research. They frequently rely on manual feature engineering and selection, which is labour-intensive and requires domain expertise. Furthermore, they may struggle to capture the intricate patterns and non-linear relationships present in vessel traffic data. Additionally, the interpretability of these ML models is often limited, posing challenges in comprehending the underlying factors influencing the prediction.

2.2. *Review of deep learning methods for vessel traffic flow prediction*

In contrast, DL methods have emerged as a promising approach to VTF prediction. By automatically extracting features from raw data and adeptly handling intricate patterns, DL models offer the potential to surmount the limitations of traditional ML methods. Moreover, DL models can leverage their hierarchical structures to effectively capture complex temporal dependencies, enhancing the

Table 1
The classical ML methods for VTF prediction.

Methods	Representative references	Characteristics	Advantages	Disadvantages
Linear regression	(Sousa et al., 2007)	(1) Simple and commonly used; (2) Quick computation; (3) Sensitivity to outliers.	(1) Easy to interpret; (2) Provide insight into the relationship between predictor variables and VTF.	(1) Assume a linear relationship; (2) Can not effectively predict non-stationary time series data; (3) Have limitations in capturing complex relationships or non-linear patterns in VTF data.
Variance analysis models	(Makowski et al., 2006)	(1) Analysis of Variance; (2) Comparison of group means; (3) Hypothesis testing.	Useful for analysing variations and identifying factors affecting VTF.	Have limitations in predicting non-stationary time series data effectively.

KF	(He et al., 2019; Muruganantham et al., 2016; Okutani and Stephanedes, 1984)	(1) Dynamic model;
		(2) State estimation;
		(3) Prediction-update cycle.
			(1) Incorporation of uncertainties and dynamic changes in VTF;
			(2) Noise filtering;
			(3) Adaptive to time-varying changes.
				(1) Linear assumption;
				(2) Gaussian distribution assumption;
				(3) Limited handling of nonstationarity VTF patterns;
				(4) Difficulty in capturing complex dependencies.

GM	(Grifoll, 2019; Kayacan et al., 2010)	Regenerate VTF data with strong regularity based on overall change rules.	(1) Handle small data sets; (2) Incorporation of expert knowledge; (3) Parameter optimisation.	(1) Overreliance on historical data; (2) Sensitivity to initial conditions; (3) Difficulty in handling nonlinearities; (4) Limited adaptability to irregular patterns.

MM	(Zou et al., 2022)	(1) Memoryless property;
		(2) Transition probabilities;
		(3) State space definition.
			(1) Efficient computation;
			(2) Adaptability to dynamic environments.
				(1) Memoryless assumption;
				(2) Lack of contextual information;
				(3) Sensitivity to initial conditions.
				(4) Limited prediction horizon (mainly short-term prediction).

KDE	(Li et al., 2018; Xiao et al., 2017)	(1) Infer overall data distribution;
		(2) Predict future VTF based on the probability density function;
		(3) Nonparametric density estimation.
			(1) Localised prediction;
			(2) Probabilistic output;
			(3) Robustness to outliers.
				(1) Bandwidth selection is challenging;
				(2) High computational complexity;
				(3) Sensitivity to sample size.

ARIMA	(Sadeghi Gargari et al., 2022; Williams et al., 1998)	(1) Time series modelling; (2) Combine the autoregressive and moving average components to model the relationship and dependency.	(1) Simple and interpretable; (2) Suitable for predicting future traffic based on historical patterns.	(1) Stationarity assumption; (2) Limited capture of nonlinear relationships for VTF data; (3) Sensitivity to model parameters.

SVM	(Mokhtarimousavi et al., 2019; Zhang and Wu, 2022)	(1) Kernel trick;
		(2) Margin maximisation;
		(3) Support vectors.
			(1) Effective in high-dimensional spaces;
			(2) Robust to outliers;
			(3) Global optimum solution.
				(1) High computational complexity in large VTF datasets;
				(2) Sensitivity to parameter tuning;
				(3) Require a large amount of memory.

VTF prediction accuracy. Therefore, the adaptation of DL methods is deemed essential to augment the precision, robustness, and interpretability of VTF prediction models.

The DL method addresses the challenge of VTF prediction by constructing a training network and mining the changing features from the data through continuous iteration. Various network architectures come with distinct characteristics and benefits. In this section, DL methods are categorised into three groups according to their network training strategies: discriminative, generative, and hybrid-based methods.

2.2.1. Discrimination-based methods

Discrimination-based DL methods can also be called conditional discriminant models, which learn relevant feature changes based on historical observation data (Li et al., 2023c). Subsequently, when forecasting future data, they gauge the likelihood of alignment with the target value. Standard primary networks mainly in this context predominantly encompass CNN (Li et al., 2022c), RNN (Yu et al., 2019), and Autoencoder (AE) (Wang et al., 2016).

CNN, characterised by its deep structure and convolutional computations, is a feedforward network capable of learning and providing a translation-invariant classification of input information, relying on its hierarchical structure. It has frequent applications in addressing challenges within image or video processing domains, encompassing tasks like deblurring, visual enhancement, and object recognition, among others. Over the past few years, an increasing number of researchers have applied it for the study of time series data (e.g., VTF) prediction. While VTF sequence data naturally evolves over time, it also exhibits specific periodic patterns. For instance, VTF data at a given time node not only correlates closely with adjacent time node data but also displays a strong connection with data from the same time period before or after. Consequently, some studies transform one-dimensional (1D) VTF data into two-dimensional (2D) matrices and employ CNN to uncover the periodic variation characteristics within VTF data (Wang et al., 2024). For example, Li et al. (2023d) proposed a multi-view network construction approach aimed at addressing the VTF prediction challenge. In this method, the spatial view leverages CNN to adeptly capture the periodic fluctuations within VTF data. Zhou et al. (2020) divided the research area into a matrix and employed CNN to learn the patterns of all grid-based VTF inflows and outflows, thereby facilitating prediction tasks. Wang et al. (2019) developed a spatiotemporal method to describe and forecast VTF within a specific region. They proposed a prediction technique utilising multi-hexagonal Convolutional Neural Networks (mh-CNN) to offer substantial flexibility and predict VTF data within a variable time window. Liang et al. (2022) introduced a spatiotemporal multi-graph convolutional network designed for VTF prediction to capture both spatial and temporal patterns. Extensive experiments were conducted using actual AIS datasets, and the fine-grained prediction outcomes underscored the outstanding performance of the optimised network, showcasing superior accuracy and robustness.

RNN, is a network specifically engineered for handling sequential data (Salehinejad et al., 2018). It proves highly suitable for time series data research. However, the gradient vanishing or exploding issues during network training can significantly impact prediction accuracy (Hao et al., 2022). To address these issues, LSTM (Graves, 2012) and GRU (Dey and Salem, 2017) have emerged as valuable solutions. Notably, GRU stands as a simplified counterpart of LSTM. While LSTM comprises three gates (i.e., input, forget, and output gates), GRU streamlines the structure to include only two: the reset gate and the update gate (Do et al., 2019). In the study of VTF prediction research, Lee et al. (2023) developed an algorithm using LSTM to forecast future maritime traffic information from past data, with the aim of improving autonomous vessel performance. They assessed the network's predictive capabilities by considering many factors (e.g., sea traffic volume, fluctuations in sea traffic congestion, and volatility). Their conclusion highlighted the model's ability to discern features crucial for predicting maritime traffic conditions. Xiao et al. (2023) introduced a new prediction method that takes into account weather conditions and employs both GRU and Sequence to Sequence (Seq2Seq) models. Their approach sought to improve prediction accuracy by accounting for environmental influences. Xu and Zhang (2022) considered spatiotemporal correlation features and integrated them into a GRU model for VTF data prediction to enhance prediction accuracy. Zhou et al. (2021) proposed a new prediction model termed Particle Swarm Optimisation LSTM (PSOLSTM), addressing the challenge of selecting optimal parameters in LSTM models. The validation, conducted with real inland waterway data, showcased enhanced prediction accuracy.

Recognising the potential benefits of capturing temporal variations in both forward and backward directions, Bi-LSTM (Zhang et al., 2020) and Bi-GRU (Yang et al., 2021) models were developed. Specifically, Bharti and Kumar (2023) introduced a method termed Particle Swarm Optimisation (PSO) and Bi-LSTM (PSO-Bi-LSTM). This method leverages PSO to globally search for the optimal model parameters, employing nonlinear variable inertia weights instead of linear weights, ultimately enhancing prediction accuracy. Gao et al. (2018) developed an online real-time vessel behaviour prediction model by both Bi-LSTM and RNN. This prediction model enhances the correlation between past and future data, thereby boosting accuracy. Wang et al. (2022b) used the Bi-GRU method for traffic flow prediction research, and their experimental results indicated superior performance compared to LSTM, GRU, and Bi-LSTM. While RNN and its variations are proficient in capturing the temporal changing characteristics within sequential data, they encounter a limitation − the necessity for consistent input data size during network training.

To surmount this constraint, Seq2Seq has been applied to the time series data prediction (Xing et al., 2023). Seq2Seq adopts an encoder-decoder structure (Sutskever et al., 2014), with the encoder converting the input sequence into a fixed-length vector and the decoder generating an output sequence based on this vector. Both the encoder and decoder can be composed of RNN and its variations. In practical research, Cao et al. (2022) introduced a SpatioTemporal Seq2Seq Network (STSSN) by considering both the spatial correlations, which are heterogeneous and time-varying, and the temporal correlations, including sequences and periodic patterns. Comparative experiments have validated the superiority of STSSN on two real-world datasets. Forti et al. (2020) explored sequence prediction methods by utilising an LSTM encoder-decoder architecture to effectively extract the long-term temporal correlations within AIS data sequences and enhance the prediction capabilities.

AE is a neural network employed in semi-supervised and unsupervised learning, with its primary function being to represent and learn input information by using it as the learning target (Wang et al., 2016). An autoencoder comprises two essential components, an encoder and a decoder, serving as versatile learning algorithms commonly employed for tasks like dimensionality reduction and outlier detection (Sun et al., 2018). A growing number of researchers have employed this method for time series prediction. For example, Jin et al. (2022) put forward a novel prediction method called Planar Flow-based Variational AE (PFVAE), which integrates LSTM as the autoencoder component and devised a variational autoencoder for prediction to effectively mitigate the impact of noise on prediction. Comparative experiments demonstrated that this method outperformed others in terms of prediction accuracy, highlighting its significance for addressing time series prediction challenges. Tiwari et al. (2022) introduced a new DL method based on AE, which integrates LSTM to perform prediction tasks on target data. It utilises grid search to automatically select the best parameters by exploring various hyperparameter combinations, significantly improving prediction accuracy. Nguyen and Quanz (2021) proposed a new time latent autoencoder method to perform the nonlinear decomposition of multivariate time series. This method enables end-to-end learning of latent spatial prediction models over time using DL techniques. A growing number of researchers have employed this method for time series prediction (Cao et al., 2024; Fu et al., 2022; Hu et al., 2024).

Discrimination-based DL methods focus primarily on identifying feature changes that are directly linked to the target variable in the input data. However, this focus may cause them to overlook complex, nonlinear relationships and deeper feature interactions, especially in intricate scenarios. These limitations can be addressed by improving input feature design and incorporating generative models or other advanced DL techniques to enhance performance and adaptability.

2.2.2. Generation-based methods

Generation-based DL methods are employed in prediction through the utilisation of the combined probability distribution of both sample and target data. They can be employed to generate a set of random instances comprising samples and their corresponding targets (Han et al., 2021). Typical primary networks encompass Restricted Boltzmann Machines (RBM) (Zhang et al., 2018), Deep Belief Networks (DBN) (Mohamed et al., 2012), Generative Adversarial Networks (GAN) (Goodfellow et al., 2020).

RBM is a generative random network with binary variables, including visible and hidden units, each taking on values of either zero or one. The network can be viewed as a bipartite graph, with connections only between visible units and hidden units, and no connections between units of the same type (Wu et al., 2020). RBM has been utilised in dimensionality reduction, classification, and feature learning, and its training can be adapted to specific tasks using either supervised or unsupervised learning methods. Zhang and Xin (2021) proposed a short-term traffic flow prediction method using Deep Learning Support Vector Regression (DL-SVR), which includes an RBM visible input layer and a radial SVR output layer. Meanwhile, they applied the T-mutation particle swarm optimisation algorithm to select essential parameters for the new method, thereby improving the prediction accuracy. Li et al. (2020) put forward a deep fusion technique capable of concurrently handling categorical and continuous variables. In this approach, RBM processes categorical variables, while stacked Gaussian Bernoulli RBM handles continuous variables. The joint layer in this method is responsible for fusing and extracting data change features. Notably, this approach excels in exploring nonlinear and intricate patterns within traffic accident and traffic flow data. Kong et al. (2019) used RBM as a prediction method for sequence data and constructed a long-term model of chaotic time series polymorphism employing phase space reconstruction to identify the data patterns. Comparative experimental results underscore the superior prediction accuracy of this approach compared to shallow neural network prediction methods.

DBN is a generative model that enables the neural network to achieve training data with maximum probability by adjusting the weights between its neurons, featuring multiple neuron layers categorised into dominant and recessive neurons, with explicit elements for input and implicit elements for feature extraction (Yang et al., 2018). This method has been widely employed in handwritten character recognition, speech recognition, and image processing. In recent years, researchers have extended its use to time series prediction tasks. For instance, Zhu et al. (2023) put forward a combined prediction method using DBN and Non-dominated Sorting Genetic Algorithm II (NSGA-II) to optimise network parameters, thereby enhancing the prediction accuracy and stability of the data. Qin et al. (2017) harnessed DBN to express nonlinear relationships effectively and combined it with ARIMA for a new prediction method that temporal correlation, spatial heterogeneity, and complex nonlinear relationships between environmental factors and predicted data, incorporating PSO to improve the overall training efficiency. Huang et al. (2014) proposed a deep architecture comprising two parts: a DBN-based bottom framework for unsupervised feature learning and a multi-task regression layer for data prediction.

GAN consists of two models: a generative model and a discriminative model. The generation model typically employs a deconvolution neural network or fully connected neural network to generate a 2D matrix from input data, while the discriminative model functions as a binary classifier using CNN (Zhan et al., 2018). GAN has found extensive applications in time series data prediction. For instance, Jia and Ma (2023) introduced a Conditional Time-Generated Adversarial Network (CTGAN) that employs adversarial training strategies, trajectory generators, and intention classifiers to generate intention-constrained trajectories by forming a closed loop with feedback information signal, ensuring consistent trajectories adhering to reasonable vessel dynamics. Zheng et al. (2022) proposed a new traffic flow prediction method called Graph Convolution and Generative Adversarial Neural Networks (GCN-GAN), which utilises GCN to extract historical traffic flow information within a graphical structure and employs GAN for generating reliable traffic flow prediction results through adversarial training. Zhang et al. (2021) put forward a self-attention generative adversarial network for sequence data prediction, wherein the GAN module incorporates a self-attention layer to mine patterns of data changes and adjusts the parameters of the entire network framework in the reinforcement learning module, demonstrating the method's reliability in performing prediction tasks through comparative experiments.

While generation-based DL methods are effective at producing realistic data and modelling complex distributions, they face challenges such as high computational demands, training instability, reliance on data quality, and limited scalability. Addressing these issues requires careful model design, optimised hyperparameter tuning, and advanced training techniques to improve their robustness and practical applicability.

2.2.3. Hybrid-based methods

The hybrid-based method essentially involves effectively combining two or more DL methods to predict time sequence data, aiming to integrate the advantages of multiple DL techniques to improve prediction accuracy and stability. For instance, the fusion of CNN and LSTM proves effective in capturing both the periodic change characteristics and temporal development attributes in VTF data, thereby improving the accuracy of training networks in fitting the overall change patterns of data. Li et al. (2023d) put forward a multi-view network framework that combines CNN and LSTM to capture the spatiotemporal attributes of VTF, with comparative experiments confirming its effectiveness compared to a single network. Zhou et al. (2020) introduced three DL-based methods for predicting vessel inflow and outflow within a specified area, incorporating CNN, LSTM, and integrated networks. In particular, the integrated network combines CNN and Bi-LSTM, with comparative experiments demonstrating its superior performance. GCN extends convolution operations from traditional data types (e.g., images or grids) to graph data, enabling accurate mining of spatial relationships within data (Chen and Chen, 2022; Hou et al., 2023). Therefore, some researchers combine GCN with RNNs for prediction tasks on time series data. Zhao et al. (2022) proposed a fusion of K-hop Graph Convolutional Network (k-GCN) and LSTM for vessel speed prediction, leveraging a three-step process that involves creating a vessel network model, using GCN to capture spatial node correlations and employing LSTM to harness spatiotemporal node correlations. Li et al. (2022a) developed a new network prediction framework, Spatial-Temporal GCN-GRU (ST-GGRU), in which GCN captures complex spatial correlations while GRU captures temporal correlations within sequence data. Huang et al. (2020) put forward a GCN-LSTM hybrid method featuring encoder and decoder structures, embedding GCN into the structure of LSTM to capture traffic flow's spatial and temporal characteristics.

As research advances, scholars increasingly explore combining three DL methods to enhance predictive performance. Liu et al. (2022) developed a new hybrid network framework based on DL, effectively combining GANs and Bi-LSTM-based AE. This approach employs GAN to obtain reconstruction residuals and learn discriminative representations while utilising AE to extract essential temporal features, ultimately constructing a supervised learning model for feature integration and diagnostic result prediction. Hou et al. (2021) proposed a multi-step traffic flow prediction method based on an attention-based spatiotemporal graph neural network and a short-term memory neural network. This method can effectively capture the complex spatial dependencies within transportation networks, thereby improving prediction performance. Bao et al. (2023) proposed a multi-step traffic flow prediction method based on an attention-based spatiotemporal graph neural network and a short-term memory neural network. This method improves prediction accuracy by capturing intricate spatial dependencies in transportation networks.

Hybrid methods combine the strengths of multiple DL methods, balancing predictive and generative capabilities, and are ideal for scenarios requiring data generation alongside accurate prediction. Nonetheless, existing prediction techniques in hybrid-based methods overlook the interplay of spatial, temporal, and semantic correlations in VTF data in dynamic water areas. This paper proposed a new MPTNSR model to address this issue. The proposed MPTNSR model offers significant advantages over discrimination-based methods by incorporating GCN to enhance feature representation and by effectively modelling complex spatial–temporal-semantic interactions. Compared to generation-based methods, it focuses directly on prediction tasks, maintaining computational efficiency while delivering robust and accurate results. Additionally, the MPTNSR model demonstrates strong adaptability to noisy data, overcoming the limitations of generation-based methods that rely heavily on accurate data distribution modelling. These strengths make the MPTNSR model a superior choice for VTF prediction tasks in diverse and complex scenarios.

2.3. Research gaps

Based on a comprehensive review of the existing literature, the following research gaps are identified, reflecting the motivation for the proposed MPTNSR model and showcasing its advantages over conventional deep learning models:

(1) Limited modelling of complex and multi-dimensional interactions.

Current VTF prediction methods often rely on DL architectures that are not well-equipped to handle the complex, non-linear interactions between key variables such as vessel traffic density and temporal patterns. These limitations hinder the accurate modelling of the periodic and semantic relationships inherent in VTF data. A new methodology like MPTNSR, which can incorporate periodic, temporal, and semantic perspectives, is essential to effectively capture these multi-dimensional interactions.

(2) Shortcomings in unified long- and short-term prediction.

Existing approaches in the field predominantly emphasise short-term prediction, with limited capacity to address both short- and long-term dynamics of VTF. This drawback restricts their utility for strategic decision-making and long-term planning. By leveraging a CNN-BiLSTM-based architecture, the proposed model can enhance the extraction of periodic and temporal features, bridging the gap between short- and long-term prediction and offering improved robustness in diverse temporal scenarios.

(3) Fragmented analysis across multi-channel AIS data.

Most prediction models fail to coordinate the relationships between multiple AIS data channels effectively, treating them as isolated variables. This fragmented approach overlooks critical dependencies, such as the geographical and temporal correlations among channels. The MPTNSR model overcomes this limitation by employing a GCN, which enhances collaborative prediction across channels by embedding traffic flow similarity and geographical features, leading to a holistic and accurate VTF prediction framework.

(4) Inadequate loss function designs for multi-dimensional data.

Conventional loss functions, such as MSE alone, are insufficient for capturing the complex global and local patterns in multi-channel VTF data. To address this, the proposed MPTNSR integrates MSE and PCC into a novel loss function, optimising the prediction performance by accounting for both pointwise accuracy and correlation-based consistency.

The proposed MPTNSR methodology offers significant advancements over conventional DL models (i.e., discrimination-based methods, generation-based methods, and hybrid-based methods) by simultaneously capturing periodic, temporal, and semantic patterns through holistic feature extraction, which is often overlooked in traditional approaches. From a methodological perspective, the MPTNSR model embodies the strengths of a hybrid approach, combining CNN for spatial feature extraction, LSTM for temporal modelling, and GCN for semantic learning. This integration ensures robust performance, addressing the diverse challenges of VTF prediction. From a descriptive perspective, the model aligns with the practical demands of maritime applications, where efficiency, accuracy, and adaptability to complex relationships are critical. The MPTNSR model ensures it can handle real-world scenarios effectively, making it superior to traditional discrimination-based or overly complex hybrid models.

3. Preliminary

This section aims to provide clear definitions and establish research problems that can aid in comprehending and achieving precise VTF prediction. VTF data in the target area is calculated based on timestamp, latitude, and longitude information from AIS data.

3.1. Definitions

Definition 1. ***Vessel trajectory.*** *A vessel trajectory VT with a length l consists of a series of points. The mathematical expression is shown below.*

$$VT = \{P_1, \cdots, P_o, \cdots, P_l\},\ P_o = \{t_o,\ lat_o, lon_o\},\ o = 1, \cdots, l \tag{1}$$

where t_o, lat_o, and lon_o indicate the time, latitude, and longitude of the *o-th* point P_o, respectively. These vessel trajectories are utilised to calculate VTF data using AIS data in Section 3.3.

Definition 2. ***VTF data.*** *It refers to the number of vessels passing through a designated section of a waterway within a unit of time, measured in units such as vessels per hour, per two hours, or vessels per day. In this study, the VTF data indicates the volume over a two-hour interval, implying that there are* 12 *time periods in a day.*

$$Dataset = \left\{t, d, VTF_{t,d}\right\},\ t = 1, \cdots, 12;\ d = 1, \cdots, D \tag{2}$$

$$M^{12\times D} = \begin{bmatrix} VTF_{1,1} & \cdots & VTF_{1,d} & \cdots & VTF_{1,D} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ VTF_{t,1} & \cdots & VTF_{t,d} & \cdots & VTF_{t,D} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ VTF_{12,1} & \cdots & VTF_{12,d} & \cdots & VTF_{12,D} \end{bmatrix} \tag{3}$$

where $VTF_{t,d}$ express the traffic volume in the *t-th* time interval of the *d-th* day. VTF dataset consists of time intervals, days, and traffic volume.

Definition 3. **VTF time series.** It consists of sequences organised chronologically for each day.

$$\begin{aligned} T_d &= \left\{VTF_{1,d}, \cdots, VTF_{12,d}\right\},\quad d = 1, \cdots, D \\ T^t &= \left\{VTF_{t,1}, \cdots, VTF_{t,D}\right\},\quad t = 1, \cdots, 12 \end{aligned} \tag{4}$$

where T_d express the VTF time series on the *d-th* day, and T^t indicates the VTF time series on the *t-th* time interval.

3.2. Problem statements

Problem generation: The accuracy of VTF prediction is influenced by several crucial factors, including the dynamic, periodic, and time-varying characteristics of VTF data, as well as the outcomes of interactions among different channels. Therefore, the four research problems are listed below.

Problem 1: How to effectively extract the periodic features in VTF data?

To effectively extract the periodic features in VTF data, CNN is applied to different neighbourhood matrices (i.e., different moments within the same day and consecutive moments within a day) in $M^{12\times D}$. The model expression is shown below.

$$f_{CNN}(M^{n\times n}) = F_{1\times W} \tag{5}$$

$$M^{n\times n} = \begin{bmatrix} VTF_{t-l,\ d-l} & \cdots & VTF_{t-l,\ d} & \cdots & VTF_{t-l,\ d+l} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ VTF_{t,\ d-l} & \cdots & VTF_{t,\ d} & \cdots & VTF_{t,\ d+l} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ VTF_{t+l,\ d-l} & \cdots & VTF_{t+l,\ d} & \cdots & VTF_{t+l,\ d+l} \end{bmatrix} \tag{6}$$

where $l = \lfloor n/2 \rfloor$, $n = 3, 5, 7$, $f_{CNN}(\cdot)$ indicates the convolution function operation. $\lfloor \cdot \rfloor$ denotes floor function, which refers to the process of rounding down a real number to the nearest integer that is less than or equal to it. $M^{n\times n}$ represents the local matrix of the whole VTF data matrix, and $F_{1\times W}$ expresses the eigenvector.

Problem 2: How to accurately capture the VTF information on the temporal evolution?

The temporal evolution features can be captured by the Bi-LSTM network, including the information on the forward and backward directions. The eigenvector $F_{1\times W}$ is used as the input to predict the subsequent VTF data.

$$f_{Bi-LSTM}(F^{t-i}_{1\times W}, \cdots, F^{t}_{1\times W}) = VTF_{t+1,d} \tag{7}$$

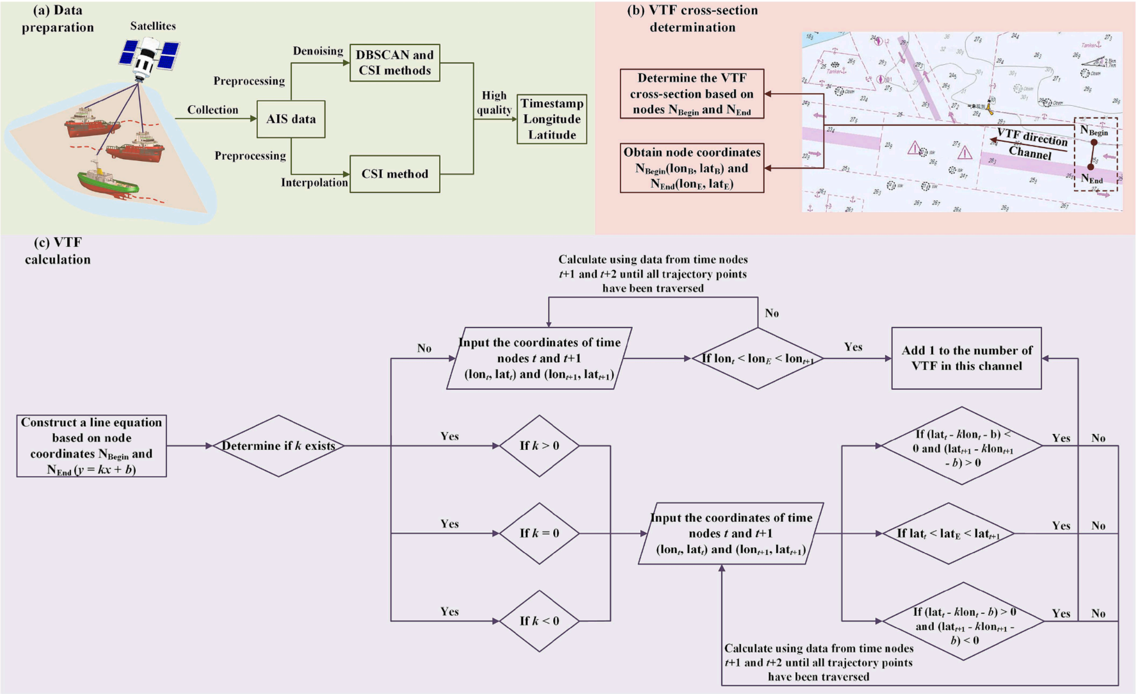
where $f_{Bi-LSTM}(\cdot)$ represents the Bi-LSTM network operation, i is established based on the number of time points provided as input to the Bi-LSTM network, and $VTF_{t+1,d}$ is the VTF data in the next time interval.

Problem 3: How to achieve collaborative prediction among multiple channels?

To realise collaborative prediction, it is necessary to consider the relationships among multiple channels and different time intervals. First, analyse the location information of different channels. Then, measure the similarity between time intervals of different channels. Ensure the sum of the weights for location and similarity information equals 1. Next, calculate semantic information to generate a relation matrix. Finally, use the GCN model to extract the feature vector.

$$\begin{gathered} f(\alpha M_{location} + \beta M_{similarity}) = M_{relation} \\ \alpha + \beta = 1 \\ f_{GCN}(M_{relation}, Dataset) = F_{1\times Z} \end{gathered} \tag{8}$$

where $f(\cdot)$ and $f_{GCN}(\cdot)$ are the semantic information extraction and GCN network operation, respectively. $M_{location}$, $M_{similarity}$, and $M_{relation}$ indicate the location, similarity, and relationship matrix, respectively. α and β are the weights of the location and similarity matrix.



(a) Data preparation
Satellites
Collection
AIS data
Preprocessing
Denoising
DBSCAN and CSI methods
Preprocessing
Interpolation
CSI method
High quality
Timestamp Longitude Latitude
(b) VTF cross-section determination
Determine the VTF cross-section based on nodes N$_{Begin}$ and N$_{End}$
Obtain node coordinates N$_{Begin}$(lon$_B$, lat$_B$) and N$_{End}$(lon$_E$, lat$_E$)
VTF direction
Channel
N$_{Begin}$
N$_{End}$
(c) VTF calculation
Construct a line equation based on node coordinates N$_{Begin}$ and N$_{End}$ ($y = kx + b$)
Determine if k exists
No
Yes
Yes
Yes
If $k > 0$
If $k = 0$
If $k < 0$
Input the coordinates of time nodes t and t+1 (lon$_t$, lat$_t$) and (lon$_{t+1}$, lat$_{t+1}$)
Calculate using data from time nodes t+1 and t+2 until all trajectory points have been traversed
If lon$_t$ < lon$_E$ < lon$_{t+1}$
No
Yes
Add 1 to the number of VTF in this channel
Input the coordinates of time nodes t and t+1 (lon$_t$, lat$_t$) and (lon$_{t+1}$, lat$_{t+1}$)
If (lat$_t$ - $k$lon$_t$ - b) < 0 and (lat$_{t+1}$ - $k$lon$_{t+1}$ - b) > 0
If lat$_t$ < lat$_E$ < lat$_{t+1}$
If (lat$_t$ - $k$lon$_t$ - b) > 0 and (lat$_{t+1}$ - $k$lon$_{t+1}$ - b) < 0
Yes
No
Yes
No
Yes
No
Calculate using data from time nodes t+1 and t+2 until all trajectory points have been traversed

Fig. 2. Visual illustration of the VTF calculation process.

$F_{1\times Z}$ is the feature vector.

Problem 4: How can loss functions enhance the accuracy of measuring prediction errors?

A novel loss function is introduced to ensure precise VTF prediction, which combines the MSE and PCC functions. It takes into account not only the discrepancy between the predicted and actual values by MSE but also the variations in predicted values across the entire series by PCC. Consequently, the new loss function incorporates both local and global changes, as depicted below.

$$f_{loss} = MSE + PCC \tag{9}$$

where $f_{loss}(\cdot)$ indicates the new loss function.

3.3. VTF calculation process based on AIS data

VTF data in the target area is calculated based on timestamp, latitude, and longitude information from AIS data. Each line segment, formed by two nodes, represents a cross-section of the VTF data. When a vessel's trajectory intersects this cross-section, it indicates an increment of 1 in the VTF data for the target area. The calculation of VTF primarily involves determining whether two consecutive points on the vessel's trajectory intersect with the cross-section. The specific calculation method is displayed in detail in Fig. 2.

According to the structure shown in Fig. 2, the VTF calculation process mainly includes two aspects. One is the preparation work, which involves AIS data (i.e., timestamps, latitude, and longitude coordinates) preparation and the determination of VTF cross-section. The original AIS data often contains noise and incomplete information, which may lead to errors if directly used for statistical VTF calculation. Therefore, this paper uses the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) method (Mieczyńska and Czarnowski, 2021) to identify and remove noise points in vessel trajectories and reconstruct the missing points based on the Cubic Spline Interpolation (CSI) method (Wolberg, 1988). To further improve the quality of vessel trajectory data, the CSI method is employed to address the issue of data fragmentation, with interpolation operations every 10s. Furthermore, the VTF cross-section can be determined through two nodes (i.e., N_{Begin} and N_{End} in Fig. 2(b)), with coordinates (lon_B, lat_B) and (lon_E, lat_E), respectively. The calculation task for VTF data is conducted based on the completed preparation work, as illustrated in Fig. 2(c). The values of k and b shown in Fig. 2(c) can be obtained using the coordinates of nodes N_{Begin} and N_{End}, with the following expression:

$$k = \frac{lat_E - lat_B}{lon_E - lon_B} \tag{10}$$

$$b = lat_E - \frac{lat_E - lat_B}{lon_E - lon_B} \times lon_E \tag{11}$$

To ensure that the denominator in Eq. (10) is not zero, it is essential to determine the existence of the parameter k. When it is preliminarily determined that two consecutive trajectory points are sequentially distributed on both sides of the cross-section, further analysis is needed to determine whether the intersection of a line formed by these two trajectory points and the cross-section actually lies on the cross-section. N_{is} represents the above intersection point, whose coordinate is (lon_{is}, lat_{is}). The two conditions that satisfy the above scenario are $lon_B \leq lon_{is} \leq lon_E$ and $lat_B \leq lat_{is} \leq lat_E$. In summary, the current centre of gravity calculates the intersection coordinates based on two cross-section coordinates and two continuous trajectory points, respectively. According to Eqs (10) and (11), as well as the coordinates of two continuous trajectory points at time t and $t+1$, the slope k_{tra} and intercept b_{tra} of the straight line formed by the two continuous trajectory points can be calculated. Therefore, the calculation formula for intersection N_{is} is as follows,

$$lon_{is} = \frac{b_{tra} - b}{k - k_{tra}} \tag{12}$$

$$lat_{is} = \frac{b_{tra} \times k - b \times k_{tra}}{k - k_{tra}} \tag{13}$$

where lon_{is} and lat_{is} are the longitude and latitude of the intersection N_{is}, respectively. k and b indicate the slope and intercept of the straight line where the cross-section is located, respectively. k_{tra} and b_{tra} denote the slope and intercept of the line formed by two continuous trajectory points, respectively. In principle, the straight cannot coincide with the straight line where the cross-section is located. Therefore, the denominator in Eqs. (12) and (13) are not equal to 0. However, there is a particular situation to consider, which occurs when k_{tra} does not exist. In such a case, the abscissa of the intersection point N_{is} is equal to the abscissa of two arbitrary trajectory points, and the abscissa of the two trajectory points is equal. Additionally, the y-coordinate of the intersection point N_{is} is obtained by substituting the abscissa of any trajectory point into the equation of the line where the cross-section is located. Their calculation formula is as follows,

$$lon_{is} = lon_t \tag{14}$$

$$lat_{is} = k \times lon_t + b \tag{15}$$

where lon_t indicates the abscissa of the trajectory point at time t. If the current two consecutive trajectory points do not intersect with the cross-section, the analysis continues by considering the trajectory point at time $t+1$ as the first point, the trajectory point at time $t+2$ as the second point, and so on. This process continues until it is determined that the trajectory intersects with the cross-section or all trajectory points have been examined.

Once it is established that the trajectory intersects with the cross-section, the next step is to determine the time period in which this intersection occurs. Each vessel's AIS data trajectory point includes time information, such as hours, minutes, and seconds. To facilitate the calculation of the time period when the trajectory passes through the cross-section, this paper converts all time values into seconds using a uniform conversion formula:

$$TimeData = 3600 \times DataH + 60 \times DataM + DataS \tag{16}$$

where *DataH*, *DataM*, and *DataS* represent the hours, minutes, and seconds in the timestamp, respectively.

For each pair of consecutive trajectory points, the timestamps are averaged. According to Table 2, the resulting average timestamp is classified into the corresponding time interval, determining the time period when the trajectory passes through the cross-section. It is essential to mention that during the data preparation stage of this study, interpolation operations were performed, resulting in a trajectory point recorded every 10s. As a result, any small errors arising from calculating the time average of two consecutive trajectory points to determine the time period of crossing the cross-section can be considered negligible.

4. Methodology

A detailed introduction is provided for a new VTF prediction framework and methodology. This approach thoroughly takes into account periodicity, temporal progression, and the effects of changes in adjacent region data. The technical structure encompasses three dimensions: periodic, temporal, and semantic, with a novel loss function. Within this framework, each component of the proposed MPTNSR model is clearly explained.

4.1. *The overarching framework*

This overarching framework includes three distinct perspectives and an additional loss function. The periodic view centres on the current time node's data, forming a local matrix that includes surrounding data. This local matrix is then fed into a CNN to capture the periodic characteristics of VTF data, with the surrounding data encompassing adjacent time nodes and past or future data from the same period. Multiple consecutive time nodes' local VTF matrices are processed by the CNN to obtain feature vectors corresponding to different time nodes. These feature vectors are subsequently input into the temporal view to capture temporal changes, both forward and backward, over time.

In practical scenarios involving multiple target regions performing VTF prediction tasks simultaneously, a specific correlation often exists between these regions based on geographical location and VTF variation patterns. Therefore, the semantic view aims to leverage geographical location factors and the similarity of VTF data changes to establish correlation relationships between the channel executing the prediction task and other target channels. GCN is employed to capture these unique relationships.

The output vectors from the temporal and semantic views are merged into a new vector, which is subsequently processed by a fully connected network to generate the prediction results for the next time node. To provide a more comprehensive evaluation of the difference between predicted and actual values, the new loss function effectively integrates PCC as a global assessment, serving as an additional correction term alongside the original MSE. During network training, the initial 1D sequence data is converted into a 2D matrix. The matrix's horizontal and vertical dimensions represent the number of days and data points within each day. In summary, the visual representation of the proposed MPTNSR framework is depicted in Fig. 3.

4.2. *The proposed MPTNSR model*

4.2.1. *Periodic view*

As previously stated, the periodic view employs CNN to capture the regular change attributes of the current time node. According to the input layer, a period of continuous data is selected to form a series of matrices around them as the training set $TrainSet = \{M_t^{n\times n}, M_{t+1}^{n\times n}, \cdots, M_{t+i}^{n\times n}\}$. In particular, the calculation of the VTF local matrix $M_t^{n\times n}$, as determined by Eq. (6), involves the parameter n which represents the size of the local VTF matrix. Each of these local matrices is then fed into the CNN to learn the periodic variations within the data. This process ultimately yields a feature vector as part of the periodic view.

One example of the feature vector extracted from a local VTF matrix by the CNN is employed to elaborate on the training process. During actual training, if the data is situated at the edges of the VTF matrix, it becomes necessary to pad the local matrix with zeros. Since the network data demands input in the form of a three-dimensional (3D) tensor, it is necessary to transform the local matrix from

Table 2
Time interval information corresponds to different time periods.

Time period	Time interval (second)	Time period	Time interval (second)	Time period	Time interval (second)
1st	1–7200	5th	28801–36000	9th	57601–64800
2nd	7201–14400	6th	36001–43200	10th	64801–72000
3rd	14401–21600	7th	43201–50400	11th	72001–79200
4th	21601–28800	8th	50401–57600	12th	79201–86400

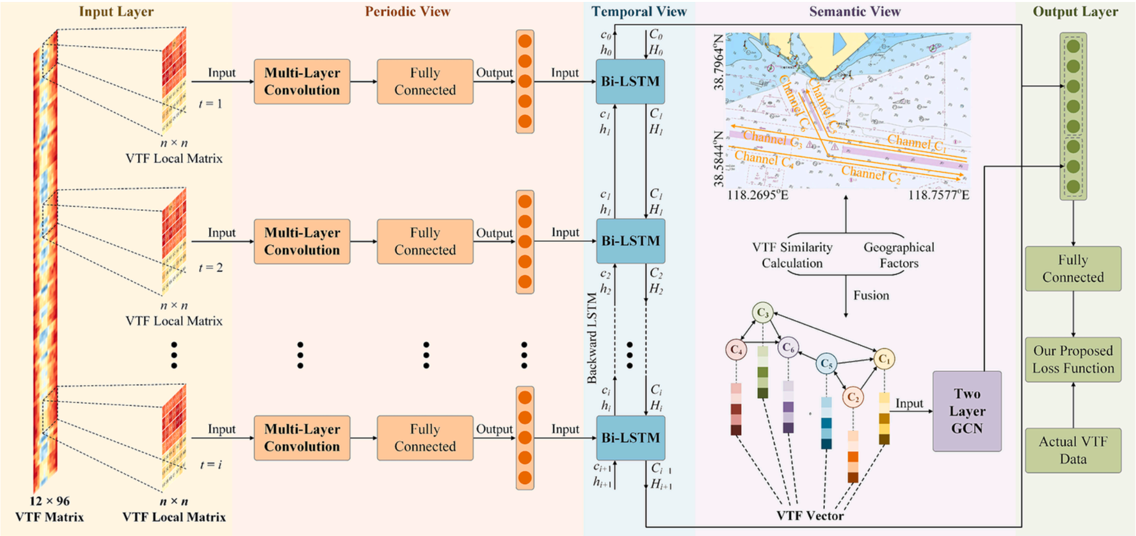


Fig. 3. The proposed framework for VTF prediction. **Periodic view**: CNN has applied to mine the periodic changes between the VTF data of the current time node and other adjacent time nodes. **Temporal view**: Bi-LSTM captures the patterns of forward and backward data changes over time. **Semantic view**: The geographical location and VTF change similarity (i.e., a network diagram of the waterways) between different target areas are combined to find the internal connection between multiple channels, as well as effectively learn the associated features based on GCN. **Loss function**: Both local (using MSE) and global (employing PCC) information are integrated into the newly proposed loss function.

a 2D to a 3D representation. CNN captures the periodic variation features within the local VTF matrix using a multi-layer convolution calculation, which can be described by the following formula:

$$F_t = f_{CNN}\left(M_t^{n\times n} \otimes W_t^{CNN}\right) + b_t^{CNN} \tag{17}$$

where W_t^{CNN} and b_t^{CNN} denote the weight and bias for convolution operations, respectively. $f_{CNN}(\cdot)$ indicates the activation function, which is the Rectified Linear Unit (ReLu) in this paper (Hara et al., 2015). $\otimes$ denotes the convolutional operation.

The MPTNSR prediction method uses multi-layer convolution calculation in actual training, resulting in the following general function expression:

$$F_t^p = f_{CNN}\left(F_t^{p-1} \otimes W_t^p\right) + b_t^p \tag{18}$$

where F_t^{p-1} represents the input data while F_t^p corresponds to the output data of the convolution layer. p signifies the specific layer of convolution being referred to. In particular, when p equals one, it indicates that F_t^{p-1} is the local VTF matrix $M_t^{n\times n}$.

The MPTNSR method utilises four convolution layers to construct a training network. Therefore, the range of values for p spans from one to four. Additionally, the kernel size used in each convolutional layer is set to 3 × 3.

Following the convolution calculation, the tensor data (i.e., $F_t^{Out} \in \mathbb{R}^{n\times n\times 2m}$) undergoes a transformation into a 2D matrix (i.e., $M_t^{Out} \in \mathbb{R}^{n\times n\times 2m}$). Subsequently, this matrix is fed into a fully connected network to get the feature vector (i.e., $FV_t^{Out} \in \mathbb{R}^{1\times 2m}$). The functional expression is described as follows:

$$FV_t^{Out} = fc\left(W_t \cdot M_t^{Out} + b_t\right) \tag{19}$$

where W_t and b_t denote the weight and bias value of a fully connected network, respectively. $fc(\cdot)$ is a linear function within the fully connected network.

A visual representation of the detailed calculation process for extracting periodic variation characteristics from the VTF data using the periodic view is presented in Fig. 3. Additionally, the optimal values of the hyperparameters n and m involved in this view will be analysed in Section 5.3.

4.2.2. Temporal view

After processing with the periodic view, a continuous local VTF matrix can obtain a series of feature vectors $FVs = \left\{FV_t^{out}, FV_{t+1}^{out}, \cdots, FV_{t+i}^{out}\right\}$ that change over time. In the temporal view, the sequence FV_S is converted into a 3D tensor and fed into a Bi-LSTM network to capture the temporal dynamic of VTF data.

The network structure of Bi-LSTM is an extension of the LSTM architecture, combining both forward LSTM and backward LSTM. Unlike the original RNN, LSTM introduces a memory state (C) in addition to the hidden state (h). Within LSTM, there are three gating mechanisms that collectively control the updates to h and C at each time step. The forgetting gate determines how much information from the previous time node $C_{t\text{-}1}$ should be transmitted to the next time node C_t. The input gate decides the information that should be stored in C_t based on the input data at the current time step. The output gate regulates the flow of information from C_t to the output hidden state (h_t). In Bi-LSTM, there are two sets of C and h, one for the forward pass and one for the backward pass, allowing information to flow in both directions. It is important to note that there is no direct connection between the forward and backward hidden layers in Bi-LSTM, ensuring that the network remains acyclic. Fig. 4 visually illustrates the structures and data flow in both LSTM and Bi-LSTM, highlighting this distinction.

To enhance the visualisation of the Bi-LSTM training process, this section illustrates the procedure using the feature vector FV_t^{out} of the t-th time step as an example, providing a comprehensive explanation. Initially, a linear transformation is performed by the linear layer on the forward hidden unit $H_{t\text{-}1}$ and the backward hidden unit $h_{t\text{-}1}$ from the previous time step t-1, and the input data FV_t^{out} from the current time node t. Additionally, an activation function is applied to map the results of the linear transformations to the range between zero and one. The representation of this process is as follows:

$$f_t^F = \sigma^F\left(W_f^F\left[H_{t-1}, FV_t^{out}\right] + b_f^F\right) \tag{20}$$

$$f_t^B = \sigma^B\left(W_f^B\left[h_{t-1}, FV_t^{out}\right] + b_f^B\right) \tag{21}$$

where W_f^F and b_f^F represent the forward network's weight matrix and bias values, respectively. W_f^B and b_f^B denote the backward network's weight matrix and bias values, respectively. σ^F and σ^B are the activation functions of the forward and backward networks, respectively, which are typically the Sigmoid functions. In particular, if the output results of f_t^F and f_t^B are both one, it indicates that all the information from both the forward and backward networks at the previous time node has been successfully transferred to the current time node.

Secondly, the input data FV_t^{out} from the current time step, along with the two hidden units (i.e., $H_{t\text{-}1}$ and $h_{t\text{-}1}$) from the previous time step, perform a linear transformation. The ultimate results are computed using distinct activation functions. The precise mathematical expressions are as follows:

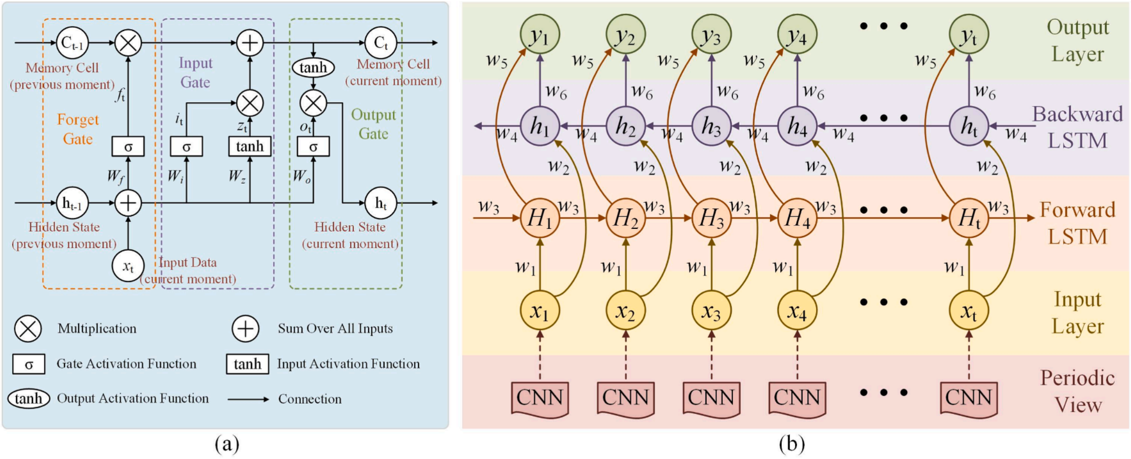
$$in_t^F = \sigma^F\left(W_{in}^F\left[H_{t-1}, FV_t^{out}\right] + b_{in}^F\right) \tag{22}$$

$$z_t^F = \tanh^F\left(W_z^F\left[H_{t-1}, FV_t^{Out}\right] + b_z^F\right) \tag{23}$$

$$in_t^B = \sigma^B\left(W_{in}^B\left[h_{t-1}, FV_t^{out}\right] + b_{in}^B\right) \tag{24}$$

$$z_t^B = \tanh^F\left(W_z^B\left[h_{t-1}, FV_t^{Out}\right] + b_z^B\right) \tag{25}$$

where both σ (using Sigmoid) and tanh represent the activation functions of forward and backward networks. Eqs. (22) and (23) correspond to the output of the forward network in the input gate, while Eqs. (24) and (25) pertain to the output of the backward network in the input gate. $W_{in}^F, W_z^F, W_{in}^B$, and W_z^B, along with $b_{in}^F, b_z^F, b_{in}^B$, and b_z^B, denote the weight matrix and bias values in the forward and backward networks.



C_{t-1}
Memory Cell (previous moment)
f_t
Forget Gate
σ
W_f
h_{t-1}
Hidden State (previous moment)
x_t
Input Gate
i_t
z_t
σ
tanh
W_i
W_z
Input Data (current moment)
C_t
tanh
Memory Cell (current moment)
o_t
Output Gate
σ
W_o
h_t
Hidden State (current moment)
Multiplication
Sum Over All Inputs
σ Gate Activation Function
tanh Input Activation Function
tanh Output Activation Function
Connection
(a)
y_1 y_2 y_3 y_4 • • • y_t
Output Layer
w_5
w_6
h_1 h_2 h_3 h_4 • • • h_t
w_4
Backward LSTM
w_2
H_1 H_2 H_3 H_4 • • • H_t
w_3
Forward LSTM
w_1
x_1 x_2 x_3 x_4 • • • x_t
Input Layer
CNN CNN CNN CNN • • • CNN
Periodic View
(b)

Fig. 4. The working mechanism, (a) LSTM Network Structure and (b) Bi-LSTM Network Structure. Specifically, ω_1 and ω_2 denote the weight matrices responsible for the connections from the input layer to the forward and backward hidden layers, respectively. Likewise, ω_3 and ω_4 correspond to the weight matrices facilitating connections between the two distinct hidden layers, while ω_5 and ω_6 represent the weight matrices that link the forward and backward hidden layers to the output layer.

Thirdly, based on the above results, C_t^F and C_t^B in the forward and backward networks at time t can be calculated, whose mathematical expressions are as follows:

$$C_t^F = f_t^F \times C_{t-1}^F + in_t^F \times z_t^F \tag{26}$$

$$C_t^B = f_t^B \times C_{t-1}^B + in_t^B \times z_t^B \tag{27}$$

Fourthly, the output result of the hidden unit is closely related to the amount of information from the previous moment, the input data of the current time node, and the storage unit. The calculation of hidden units (i.e., $H_{t\text{-}1}$ and $h_{t\text{-}1}$) in both the forward and backward networks is outlined as follows:

$$o_t^F = \sigma^F\left(W_o^F\left[H_{t-1}, FV_t^{out}\right] + b_o^F\right) \tag{28}$$

$$H_t = o_t^F \times \tanh\left(C_t^F\right) \tag{29}$$

$$o_t^B = \sigma^B\left(W_o^B\left[h_{t-1}, FV_t^{out}\right] + b_o^B\right) \tag{30}$$

$$h_t = o_t^B \times \tanh\left(C_t^B\right) \tag{31}$$

where W_o^F, b_o^F and W_o^B, b_o^B denote the weight matrix and bias values in the forward and backward networks, respectively.

Finally, the fusion of H_t and h_t from the forward and backward networks is subsequently fed into a linear fully connected network to yield the final result, expressed as follows:

$$F_{PT} = fc\left(W_{fc} \cdot [H_t, h_t] + b_{fc}\right) \tag{32}$$

where W_{fc} and b_{fc} represent a fully connected network's weight and bias value, respectively. Referring to Fig. 4, the output of this view (denoted as F_{PT}) will be integrated into a new vector within the semantic view. The ultimate prediction result will be derived by passing this combined vector through a fully connected network.

4.2.3. Semantic view

In practical scenarios, multiple channels within a target area usually execute VTF prediction tasks simultaneously. Therefore, to achieve better prediction performance, it is important to consider both the geographical location and the similarity between VTF sequence data in the semantic view. As depicted in Fig. 5, the two research water areas, Caofeidian Port (CFDP) and Chengshan Jiao (CJ), comprise six and eight distinct channels, respectively. The geographical locations of these channels, along with vessel headings, determine the inter-channel correlation in VTF changes. The CFDP water area, located in the western Bohai Sea, is notable for its intense traffic and strategic location for a major deep-water port development. The CJ water area is a crucial maritime route to the Bohai Sea and northern Yellow Sea ports, characterised by heavy ship traffic, complex weather, and high collision risks, with over 800,000 ships annually.

As depicted in Fig. 5 (a), the CFDP water area comprises six distinct channels, and the geographical locations of these channels, as well as vessel headings, determine the inter-channel correlation in VTF changes. To elucidate:

(1) Channel C$_1$ serves as the primary entry point for vessel traffic into the study area, and as such, VTF in C$_1$ is primarily self-contained and unrelated to the other five channels. The weight between C$_1$ and C$_1$ is set to 1 because of their proximity.

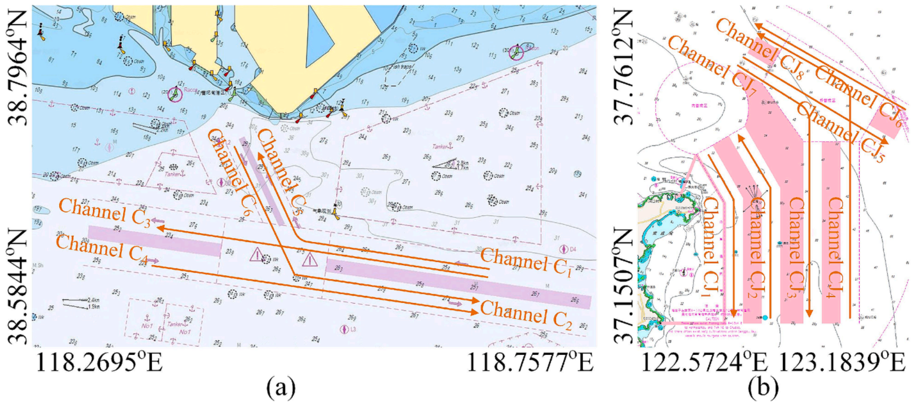


Fig. 5. The schematic diagram of two water areas: (a) the CFDP area and (b) the CJ water area. Significantly, the arrow indicates the flow direction of VTF.

(2) Channel C_2 witnesses vessel traffic from both C_6 and C_4, as vessels exiting these channels enter C_2 before departing the study area. Therefore, the correlation weight between C_2 and C_4 and between C_2 and C_6 is set to 1. Since C_1 is the main entry point for vessel traffic, the weight between C_2 and C_1 is also set to 1. Moreover, the weight between C_2 and C_2 is set to 1 due to their proximity.

(3) For Channel C_3, vessel traffic originates from C_1, with part of C_1′s traffic flowing into C_5 and the remainder flowing into C_3. Consequently, a weight of 1 exists between C_3 and C_1, as well as between C_3 and C_5. Similarly, the weight between C_3 and C_3 is set to 1.

(4) In the case of Channel C_4, vessels enter ports such as Tianjin Port through Channel C_3 and exit through Channel C_4 after completing their transportation tasks. Therefore, the weight between C_4 and C_3 is set to 1. Given that C_1 is the primary entrance for vessel traffic and there is a directional flow from C_3 to C_1 (weight is 1), the weight between C_4 and C_1 is also set to 1. Likewise, the weight between C_4 and C_4 is set to 1.

(5) For Channel C_5, vessel traffic originates from C_1, with part of C_1′s traffic flowing into C_3 and the other part flowing into C_5. Consequently, weights of 1 are assigned between C_5 and C_3, as well as between C_5 and C_1. Similarly, the weight between C_5 and C_5 is set to 1.

(6) Channel C_6 experiences vessel traffic from the CFDP via Channel C_5, with vessels leaving the study area through C_6 upon completing their transportation tasks. Therefore, the weight between C_6 and C_5 is set to 1. Given that C_1 serves as the primary entrance for ship traffic and there is a directional flow from C_5 to C_1 (weight is 1), the weight between C_6 and C_1 is also set to 1. Additionally, the weight between C_6 and C_6 is set to 1.

In summary, Table 3 illustrates the geographical position weight matrix for the CFDP water area, while the matrix for the CJ water area, calculated in the same manner, is presented in Table 4.

The geographical location of a waterway plays a pivotal role in shaping the interactions among multiple target areas. Additionally, there might be specific correlations among the VTF variation patterns observed in different channels. In the context of network training, this study establishes each period as a baseline and utilises data from consecutive days within that period as input for the training set, aiming to predict VTF data for the subsequent day. Consequently, the semantic view employs the Dynamic Time Warping (DTW) method to evaluate the similarity between VTF sequence data from two channels during different periods (Li et al., 2022b). The exponential function is selected to map DTW measures and weight values one-to-one. When the VTF variations of two channels are similar, the weight value is 1, indicating strong similarity. Conversely, if the VTF variation characteristics differ significantly, the weight value approaches 0, indicating weaker similarity. The mathematical expression for this calculation is provided below:

$$\omega_{sim} = e^{-DTW(p,q)} \tag{33}$$

where p and q represent VTF sequence data from two different waterways, respectively. $DTW(p,q)$ denotes the similarity measure between two sequences.

To comprehensively consider the impact of the geographical location of the waterway and the variation pattern of VTF, the semantic view integrates the above two influencing factors, and its calculation formula is as follows,

$$\omega = \alpha \times \omega_{gl} + \beta \times \omega_{sim} \tag{34}$$

$$\alpha + \beta = 1 \tag{35}$$

where α and β are the weights that determine the relative importance of two factors (i.e., geographic location and VTF change similarity) during network training.

In Section 5.4, an extensive set of comparative experiments will be conducted to provide a deeper understanding of how the values of these two hyperparameters affect prediction accuracy and stability. The value of ω_{gl} is derived from Table 1.

The value of ω serves to assess the correlation between two waterways by taking into account their geographical location information and the similarity in VTF changes. The weight matrix M_w can be constructed to express connections between navigation channels using ω, and each navigation channel's VTF sequence data is represented as the feature vector V_f. These individual channel feature vectors are consolidated into a 2D feature matrix M_f, which is then combined with the weight matrix and input into the GCN network to obtain hidden relationship vectors, as visually illustrated in Fig. 6. The functional expression is described in Eq. (36),

Table 3
Geographical position weight matrix between different channels in the CFDP water area (1 indicates the existence of traffic flow inflow relationship between the two channels, and 0 indicates no relationship).

	C_1	C_2	C_3	C_4	C_5	C_6
C_1	1	0	0	0	0	0
C_2	1	1	0	1	0	1
C_3	1	0	1	0	1	0
C_4	1	0	1	1	0	0
C_5	1	0	1	0	1	0
C_6	1	0	0	0	1	1

Table 4
Geographical position weight matrix between different channels in the CJ water area.

	CJ_1	CJ_2	CJ_3	CJ_4	CJ_5	CJ_6	CJ_7	CJ_8
CJ_1	1	1	0	0	0	0	0	0
CJ_2	1	1	0	0	0	0	0	0
CJ_3	0	0	1	1	1	0	1	0
CJ_4	0	0	1	1	0	0	0	0
CJ_5	0	0	1	0	1	1	1	0
CJ_6	0	0	0	0	1	1	1	0
CJ_7	0	0	0	1	0	1	1	1
CJ_8	0	0	0	1	0	1	1	1

$$F_G = GCN(M_w, M_f) \tag{36}$$

4.2.4. Loss function

The loss function serves as a measure of the disparity between the predicted values generated by a network model and the actual values. During network training, the objective is to minimise this loss function through successive iterations, utilising predefined network parameters and optimisation algorithms. In general, the choice of the loss function hinges on the specific application context and typically falls into two broad categories: regression and classification tasks.

In the context of this study, the loss function is applied to the VTF prediction task, which falls under regression. Consequently, two widely employed and effective regression loss functions are extensively used in network training. The first is the MSE (Allen, 1971), which calculates the average squared distance between the predicted and actual values of a sample. The second is the Mean Absolute Error (MAE) (Chai and Draxler, 2014), which measures the average absolute difference between the predicted values generated by the network model and the actual values of a sample. If the data is noise-free, MSE is preferred, while MAE is more suitable for noisy data. Since the data in this study has been preprocessed and is noise-free, MSE is chosen as the appropriate metric. It will be a crucial component of the proposed loss function. The functional expression of MSE is represented by Eq. (37).

$$MSE = \frac{1}{num}\sum_{i=1}^{num}(pre_i - act_i)^2 \tag{37}$$

where *num* represents the total number of VTF data (which can also be the number of time nodes). pre_i and act_i are the predicted and actual values at the i-th time node, respectively.

In this paper, the training network is devised to forecast data for the upcoming time node by utilising sample values from a sequence of continuous time nodes. Consequently, the value of *num* is configured to 1 in Eq. (37). While MSE falls within the category of regression loss functions, they each possess distinct advantages and disadvantages when it comes to assessing the effectiveness of a network model's training.

MSE offers greater complexity in gradient computation, facilitating dynamic changes in gradients that lead to rapid and precise convergence. Furthermore, the VTF data used in this paper essentially contains no outliers. Therefore, in summary, MSE will be a crucial component of the proposed prediction method's loss function.

To effectively evaluate the difference between predicted and actual values in a specific sequence, this paper combines the network's input sequence $X_{in} = \{inData_1, inData_2, \ldots, inData_n\}$ with the expected ($Data_{pre}$) and actual ($Data_{act}$) values to form two new sequences $X_{pre} = \{inData_1, inData_2, \ldots, inData_n, Data_{pre}\}$ and $X_{act} = \{inData_1, inData_2, \ldots, inData_n, Data_{act}\}$, respectively. The correlation between the original sequence X_{act} and the sequence X_{pre} (i.e., the sequence with the predicted values) is evaluated using the absolute value of the Pearson Correlation Coefficient (PCC) (Benesty et al., 2008). The function is defined as follows,

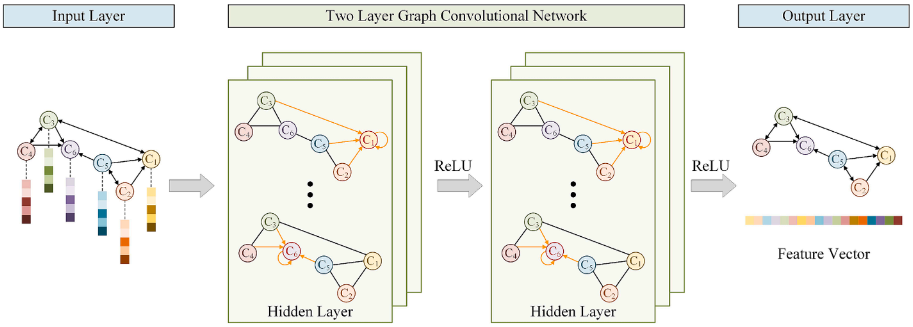


Fig. 6. The schematic mechanism of the GCN.

$$\mathrm{PCC} = \left| \frac{E\left(X_{pre}X_{act}\right) - E\left(X_{pre}\right)E\left(X_{act}\right)}{\sqrt{E\left(X_{pre}^{2}\right) - \left(E\left(X_{pre}\right)\right)^{2}}\sqrt{E\left(X_{act}^{2}\right) - \left(E\left(X_{act}\right)\right)^{2}}} \right| \tag{38}$$

This paper goes beyond considering only MSE and computes the PCC between X_{act} and the sequence X_{pre}. The proposed loss function incorporates global information into the optimised loss function, enabling a comprehensive evaluation. At its core, the proposed loss function combines local knowledge, primarily represented by MSE, with global communication, primarily indicated by PCC. The formula for the new loss function is defined as follows:

$$LF_{MSEPCC} = MSE + (1 - PCC) \tag{39}$$

The PCC falls within a value range of 0 to 1, where a higher PCC value indicates a more robust correlation between the predicted results and the original data. In practical applications, the network's objective is to minimise the loss function through iterative processes during training.

4.3. Description of the prediction component

This paper concentrates on forecasting VTF data for future time intervals based on a collection of continuous historical data used as training samples. The proposed MPTNSR method adeptly captures the periodic and temporal evolution patterns of VTF via periodic and temporal views, consequently yielding the feature vector F_{PT}. Furthermore, it employs the semantic view to extract the interrelationships among various waterways, resulting in the feature vector F_G. The amalgamation of F_{PT} and F_G vectors produces a novel vector, which is subsequently fed into a fully connected network to generate the ultimate prediction outcome. The mathematical representation for this process is defined as follows:

$$F_{New} = F_{PT} \oplus F_G \tag{40}$$

$$pre = f(W_{FC}F_{New} + b_{FC}) \tag{41}$$

where $\oplus$ is the operational symbol for merging two vectors into one vector. $f(\cdot)$ denotes the linear function of the fully connected network. W_{FC} and b_{FC} represent the weights and bias values of the fully connected network, respectively. *pre* indicates the ultimate predicted result. The comprehensive procedure for executing multi-objective regional (or multi-channel) VTF collaborative prediction using the MPTNSR method is outlined in Algorithm 1.

Algorithm 1: The proposed MPTNSR

```
Input: VTF matrix for all channels: $VTFSeq_{1,\cdots,n}^{N} = \{VTFM_1^{12\times92}, VTFM_2^{12\times92}, \cdots, VTFM_n^{12\times92}\}$;
Directed graph of VTF relationship among six channels during different periods: $GSeq_{1,\cdots,12}^{T} = \{G_1, G_2, \cdots, G_{12}\}$;
Sequence length: $SL$;
Size of local VTF matrix: $SLVM$;
Output: The trained MPTNSR model
Expand the VTF matrix of each channel outward based on SLVM
$VTFNSeq_{1,\cdots,n}^{N} = \left\{ VTFNM_1^{\left(12+\lfloor\frac{SLVM}{2}\rfloor\times2\right)\times\left(92+\lfloor\frac{SLVM}{2}\rfloor\times2\right)}, VTFNM_2^{\left(12+\lfloor\frac{SLVM}{2}\rfloor\times2\right)\times\left(92+\lfloor\frac{SLVM}{2}\rfloor\times2\right)}, \cdots, VTFNM_n^{\left(12+\lfloor\frac{SLVM}{2}\rfloor\times2\right)\times\left(92+\lfloor\frac{SLVM}{2}\rfloor\times2\right)} \right\}$
Initialisation;
for $\forall i \in N$ do
    for $\forall j \in T$ do
        for $\forall k \in SL$ do // the periodic view learns the periodic features of VTF.
            $F_{1\times W}^{k} = f_{CNN}(M_{SLVM\times SLVM})$;
            $M_{SLVM\times SLVM}$ as the VTF local matrix is determined by Eq. (6) calculated;
        end
        $F_{PT} = f_{Bi-LSTM}(F_{1\times W}^{1}, F_{1\times W}^{2}, \cdots, F_{1\times W}^{k})$; // the temporal view captures the temporal variation characteristics of VTF.
        $F_G = f_{GCN}\left(G_j, AllVTF_j^{Feature}\right)$; // the semantic view extracts the characteristics of the relationship between different channels.
        $AllVTF_j^{Feature} = \begin{bmatrix} VTFM_1^{12\times92}[j:,] \\ VTFM_2^{12\times92}[j:,] \\ \vdots \\ VTFM_n^{12\times92}[j:,] \end{bmatrix}$;
        $pre_{i,j} = f_{FC}([F_{PT}, F_G])$; // Fuse the feature vectors $F_{PT}$ and $F_G$ and input them into a fully connected network to obtain the final prediction result.
    end
end
Initialise all network parameters $\theta$ in MPTNSR;
repeat
    Optimise $\theta$ by minimising the optimised loss function Eq. (9).
until the stopping criteria are met. // The number of network training sessions has reached the iteration value.
```

5. Experimental results and discussion

To thoroughly evaluate the predictive capabilities of the proposed MPTNSR method in addressing VTF prediction challenges, this study conducts extensive comparative experiments using VTF data from all channels in the CFDP and CJ water areas. These experiments provide a quantitative analysis of the effects of various network parameters, the optimised loss function, and different information weights on the prediction accuracy and stability of the MPTNSR method. Additionally, the performance of 19 prediction models is assessed using three key indicators, offering valuable insights into their effectiveness. Furthermore, ablation experiments and time complexity analysis are conducted to validate the robustness and efficiency of the proposed MPTNSR model. Finally, the prediction performance on low-quality data is also explored to demonstrate the effectiveness of the proposed MPTNSR model.

All method comparisons were conducted under the same software and hardware conditions to ensure experimental fairness. Specifically, the environment consisted of Python 3.9.13 with PyTorch version 1.13.1, running on an Intel i7-12700KF Dodeca Core processor with 32 GB of host memory.

5.1. Experimental datasets

To verify the effectiveness of the proposed MPTNSR model in VTF prediction tasks, this paper focuses on six waterways within the CFDP water area and eight channels in the CJ water area. The extracted VTF data visualisation and schematic diagram of the two water areas are presented in Fig. 7. Fig. 7 (a) and (b), providing an overview of vessel trajectories and density distribution within the CFDP and CJ water area, respectively. Fig. 7 (c) and (d) depict the distribution of the calculated VTF cross-sections for each channel in the two water areas, which serve as the basis for calculating the VTF data. If trajectories cross the corresponding channel interface, the traffic flow statistics increase by 1. Specifically, there are six and eight channels in the CFDP and CJ water areas, respectively, resulting in six and eight datasets from C_1 to C_6 and CJ_1 to CJ_8. The coordinates listed in Table 5 are derived based on the World Geodetic System-1984 (WGS-84) coordinate system. The longitude and latitude coordinates of nodes in VTF cross-sections for six channels are shown in Fig. 7 (c) and Fig. 7 (d).

The collection period for VTF data in each channel in this experiment spans from 1 July 2020 to 30 September 2020 (i.e., 92 days). Each day is segmented into 12 time intervals, each spanning two hours. As a result, each channel has a dataset containing 1104 time nodes of VTF data, calculated using the method described in Section 3.3. The proposed MPTNSR prediction method requires inputting data into the network in the form of a matrix, rather than a one-dimensional time series. Therefore, in this paper, 1D VTF data is

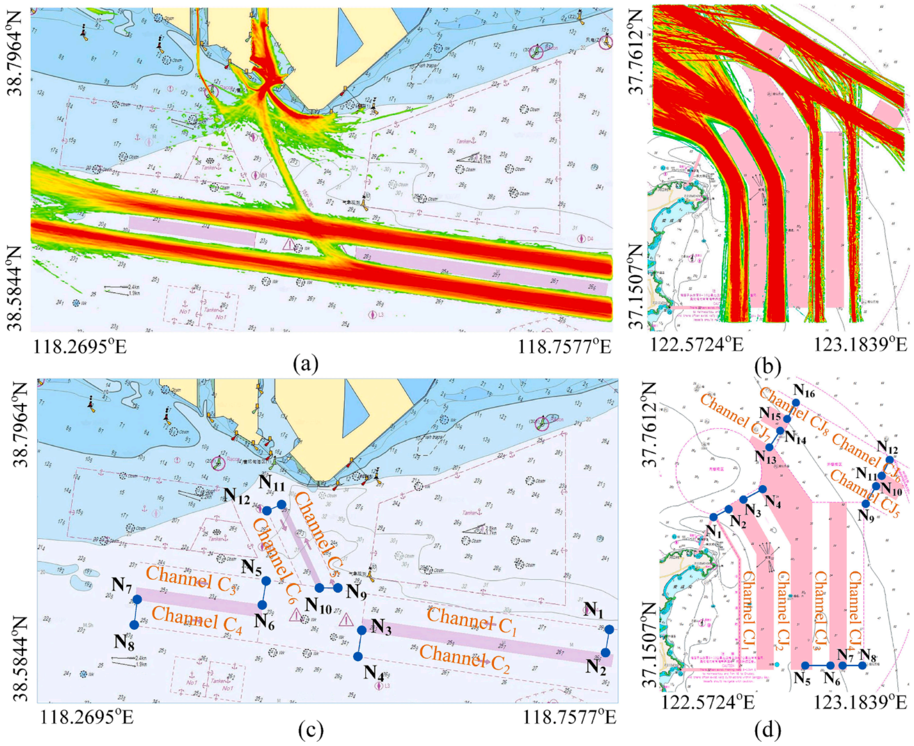


Fig. 7. Visualisation of data and channels in two water areas. (a) visualisation of vessel trajectory density in the CFDP water area, (b) visualisation of vessel trajectory density in the CJ water area, (c) schematic diagram of VTF statistical cross-section in the CFDP water area, and (d) schematic diagram of VTF statistical cross-section in the CJ water area.

converted into a 12 × 92 2D matrix, representing the hours of the day (vertical axis) and the days (horizontal axis). The VTF matrix visualisation results of two water areas are depicted in Figs. 8 and 9, respectively.

5.2. Performance indexes on prediction

This paper thoroughly assesses predictive performance across diverse scenarios by selecting two types of quantitative indicators, representing both fine-grained and coarse-grained analyses. The fine-grained indicator, the mean relative error (REMean), measures the accuracy and stability of various methods in predicting VTF data at distinct time points. The mathematical expressions are as follows:

$$REMean = \frac{1}{num}\sum_{i=1}^{num}\frac{|pre_i - act|}{act} \tag{42}$$

where *num* signifies the count of times a prediction method has been executed. When using learning-based methods for VTF prediction tasks, results can differ with each execution. To ensure reliability and consistency of prediction results, multiple experiments are conducted for each scenario, and the final prediction result is determined by averaging these values. The value of *num* is set as 10 in the experiments. *act* represents the actual VTF values at a specific time node, and pre_i denotes the predicted values for the *i*-th run. In all comparative experiments, the objective is to achieve high accuracy and stable prediction results by minimising the value of REMean.

From a coarse-grained perspective, this experiment chooses the Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and coefficient of determination (R^2) as quantitative evaluation indicators. The expressions are listed below:

$$RMSE = \sqrt{\frac{1}{tn}\sum_{j=1}^{tn}\left(avepre_j - act_j\right)^2} \tag{43}$$

$$MAPE = \frac{1}{tn}\sum_{j=1}^{tn}\left|\frac{avepre_j - act_j}{act_j}\right| \tag{44}$$

$$R^2 = 1 - \frac{\sum_{j=1}^{tn}\left(avepre_j - act_j\right)^2}{\sum_{j=1}^{tn}\left(act_j - aveact\right)^2} \tag{45}$$

where *tn* denotes the number of data nodes in a day, whose value is 12. $avepre_j$ is the average predicted value at the *j*-th time node, while act_j indicates the actual VTF data at the *j*-th time node. *aveact* represents the average of the actual VTF data at 12 time points in a day. The value of R^2 typically ranges from 0 to 1.

In theory, more accurate prediction results tend to have a higher R^2 value. Conversely, RMSE and MAPE have an opposite evaluation mode to R^2. The optimal predictive performance is achieved when the RMSE and MAPE values are minimised.

5.3. Comparison with other state-of-the-art prediction methods

To assess the effectiveness of the proposed MRTSRNet model in VTF prediction, this study carries out comparative experiments against fourteen advanced prediction methods. These methods are grouped into four categories:

(1) ML methods (i.e., ARIMA (Kumar et al., 2022) and SVM (Zhang and Wu, 2022)).
(2) NN methods (i.e., BPNN (Zhang and Wu, 2022) and WNN (Doucoure et al., 2016)).

Table 5
The longitude and latitude coordinates of nodes in VTF cross-sections for different channels in the CFDP and CJ water areas.

Water area	Channel	Node	Longitude(°)	Latitude(°)	Channel	Node	Longitude(°)	Latitude(°)
CFDP	C_1	N_1	118.7574	38.8205	C_2	N_3	118.5418	38.8194
		N_2	118.7536	38.8041		N_4	118.5387	38.8025
	C_3	N_5	118.4622	38.8532	C_4	N_7	118.3516	38.8401
		N_6	118.4587	38.8368		N_8	118.3485	38.8234
	C_5	N_9	118.5188	38.8519	C_6	N_{11}	118.4728	38.9018
		N_{10}	118.5085	38.8492		N_{12}	118.4618	38.8988
CJP	CJ_1	N_1	122.7042	37.4955	CJ_2	N_3	122.7746	37.5277
		N_2	122.7393	37.5111		N_4	122.8107	37.5444
	CJ_3	N_5	122.9429	37.1942	CJ_4	N_7	123.0266	37.1948
		N_6	122.9851	37.1945		N_8	123.0689	37.1942
	CJ_5	N_9	123.0692	37.5176	CJ_6	N_{11}	123.1111	37.5758
		N_{10}	123.0905	37.5474		N_{12}	123.1320	37.6053
	CJ_7	N_{13}	122.8498	37.6212	CJ_8	N_{15}	122.8910	37.6788
		N_{14}	122.8708	37.6501		N_{16}	122.9123	37.7088

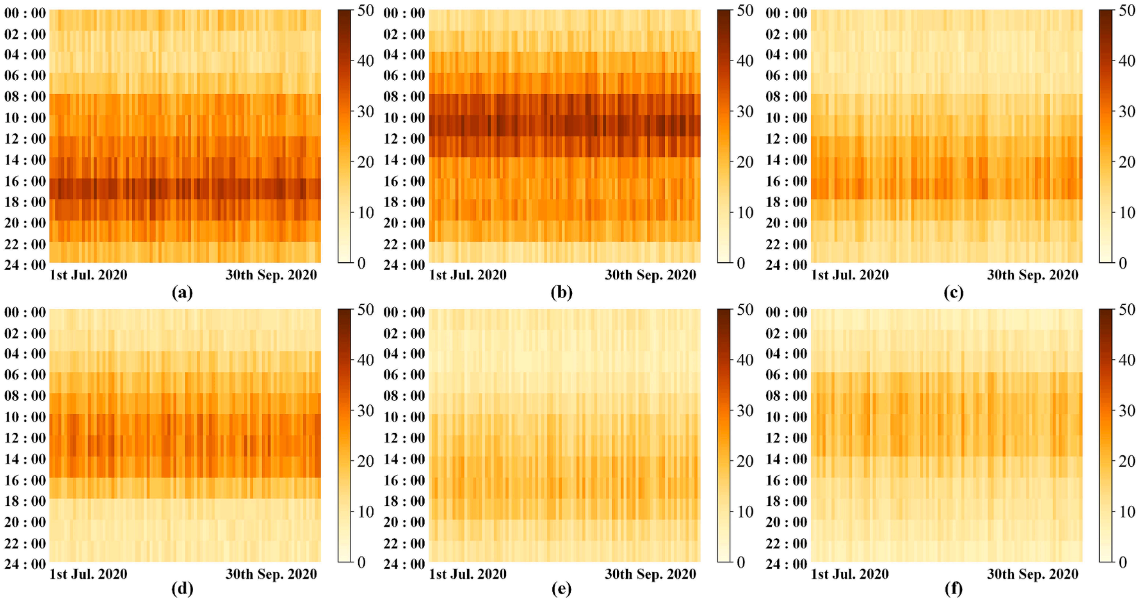


Fig. 8. Visualisation of the VTF matrix for six channels in the CFDP water area.

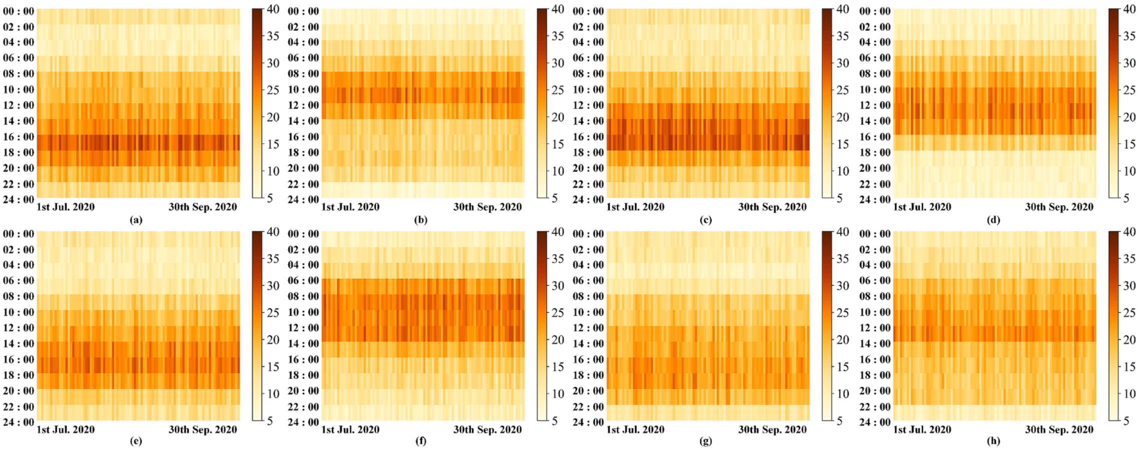


Fig. 9. Visualisation of the VTF matrix for eight channels in the CJ water area.

(3) DL methods (i.e., RNN (Liu et al., 2020), LSTM (Wei et al., 2019), GRU (Li et al., 2019), Bi-LSTM (Zhang et al., 2020), and Bi-GRU (Yang et al., 2021)).

(4) Hybrid methods (i.e., CNN-LSTM (Cheng et al., 2022), CNN-BiLSTM (Shan et al., 2021), CNN-GRU (Ma et al., 2023), CNN-BiGRU (Niu et al., 2022), LSTM-GCN (Ye et al., 2024), BiLSTM-GCN (Li et al., 2023b), GRU-GCN (Wang et al., 2022a), BiGRU-GCN (Xu et al., 2024), and the Improved CNN-LSTM with Similarity Grouping (ICLSGNet) (Li et al., 2023d).

The comparative experiments presented in this paper analyse both traditional and cutting-edge prediction models. The predictive outcomes can effectively validate the efficacy of the model introduced in this paper. The results of these 19 prediction methods are compared in subsequent sections.

5.4. Network parameter settings and optimisation

The proposed MPTNSR model integrates three views: periodic, temporal, and semantic, which are constructed based on CNN, Bi-LSTM, and GCN, respectively. These views are created to capture the cyclic and time-dependent fluctuations in VTF data and enable collaborative prediction by considering the interactions between multiple channels. Therefore, the parameters in network training mainly come from three views (which are essentially CNN, Bi-LSTM, and GCN), as illustrated in Table 6.

Table 6 provides not only the network parameter setting for the three views but also includes some general parameters that are applicable throughout the entire network, such as optimisation algorithm, learning rate, and iteration. These parameters are pivotal in determining the network's capacity to effectively grasp the evolving characteristics of historical data, thereby influencing its accuracy. The bold parameters in Table 6 represent hyperparameters, which are challenging to determine accurately and quickly through conventional methods. Hence, this paper employs an exhaustive process to assign different values to these parameters and analyse the precision and robustness of the prediction results to obtain the optimal parameter values. The multiple values of these parameters are also shown in Table 6, including the size of the local VTF matrix, input size, sequence length, learning rate, and iteration. In particular, the value of 'Input Size' in the temporal view is determined by 'Size of Local VTF Matrix' in the periodic view. Varying the size of the local VTF matrix results in different depths of each convolutional layer, leading to diverse dimensions of the output feature vectors in CNN. The output vector from the periodic view acts as the input vector for the temporal view, establishing the interconnection between these two parameters. Additionally, the parameter 'Sequence Length' in the temporal view determines the number of time nodes simultaneously fed into the network, thereby affecting the incorporation of CNN structures within the periodic view. The effect of these parameters, including the local VTF matrix size, sequence length, learning rate, and iteration count, on the accuracy and stability of VTF prediction under various values in the two water areas will be comprehensively examined. The dimensions of the local VTF matrix in CNN are closely connected to the input size in Bi-LSTM. This connection arises from the fact that alterations in the input matrix size of CNN result in varying convolutional layer depths, which in turn affect the length of the ultimate output vector within the fully connected network of CNN.

CNN captures the periodic characteristics of the local VTF matrix. The output vector obtained from CNN serves as input data for Bi-LSTM. The relationship between local VTF matrices of different sizes and the corresponding input size of Bi-LSTM in a four-layer convolutional CNN model is presented in Table 7.

The proposed MPTNSR method achieves higher prediction accuracy and optimal fitting with a 3 × 3 local VTF matrix size, as demonstrated in Tables I and II in the Appendix. The highlighted results show that this matrix size consistently delivers the highest and lowest RMSE and MAPE values across both water areas, confirming its effectiveness. The 3 × 3 local VTF matrix size outperforms larger sizes like 5 × 5 and 7 × 7 primarily due to the balance it strikes between capturing sufficient local information and maintaining computational efficiency. Therefore, a smaller VTF matrix size 3 × 3 enhances the MPTNSR model's ability to capture localised spatial features and short-range dependencies, improving prediction accuracy in dynamic maritime environments. Additionally, it reduces computational complexity, ensuring efficient processing without sacrificing performance.

The learning rate is a critical parameter that directly influences the convergence speed and stability of the MPTNSR model. Comparative experiments with learning rates of 0.0001, 0.0005, 0.001, 0.005, and 0.01, as detailed in Tables III and IV in the Appendix, reveal the model's sensitivity to changes in this parameter. These results highlight how excessively high learning rates can cause gradients to overshoot the optimal point, leading to instability, while excessively low rates may slow convergence or prevent the loss function from reaching its minimum.

The analysis demonstrates that a balanced learning rate of 0.001 strikes an ideal trade-off, ensuring both faster convergence and training stability. This moderate rate enables the model to avoid overshooting while maintaining a steady progression toward the optimal solution, resulting in enhanced prediction accuracy and robust model performance. Understanding the sensitivity of the model to different learning rates provides valuable insights for optimising training dynamics and ensuring reliable convergence in various datasets.

The number of iterations significantly impacts the MPTNSR model's ability to balance prediction accuracy and training stability. According to the results in Figs. 10 and 11, 500 iterations consistently achieve the highest R^2 values and the lowest RMSE and MAPE values across both water areas. This indicates that 500 iterations allow the model to effectively converge without overfitting or underfitting, resulting in accurate and stable predictions.

The comparative analysis highlights how varying iteration counts (e.g., 100, 300, 500, 700, and 900) influence the training process. Fewer iterations may lead to underfitting, where the model fails to capture sufficient patterns in the data, while excessive iterations can introduce noise or overfitting, reducing generalisation. The optimal choice of 500 iterations balances these extremes, ensuring the model captures both long-term trends and immediate patterns without compromising performance or stability. This analysis

Table 6
Network parameter information for the proposed MRTSRNet method.

	Parameters	Values	Parameters	Values
Periodic View	**Size of local VTF matrix**	**3 × 3, 5 × 5, and 7 × 7**	Number of convolutional layers	4
	Size of convolutional kernel	3 × 3	Stride	1
	Padding	1	–	–
Temporal View	**Input size**	**32, 64, and 128**	Hidden size	2 × Input Size
	Sequence length	**3, 6, 9, 12, and 15**	Number of hidden layer	1
Semantic View	Number of convolutional layers	2	The output dimension of the First Layer Convolution	23
	The output dimension of the second layer convolution	6	–	–
General Parameters	Optimisation algorithm	Adamax	**Learning rate**	**0.0001, 0.0005, 0.001, 0.005, and 0.01**
	Iteration	**100, 300, 500, 700, and 900**	–	–

Table 7
The ConvNet settings and temporal view's input size for different sizes of local VTF matrix.

Size of Local VTF Matrix	First layer convolution	Second layer convolution	Third layer convolution	Fourth layer convolution	Input Size (Temporal View)
3 × 3	conv3-16	conv3-16	conv3-32	conv3-32	32
5 × 5	conv3-32	conv3-32	conv3-64	conv3-64	64
7 × 7	conv3-64	conv3-64	conv3-128	conv3-128	128

underscores the importance of tuning iteration counts to optimise predictive accuracy and robustness in diverse maritime datasets.

Finally, the impact of sequence length (i.e., 3, 6, 9, 12, and 15) is explored on the accuracy of VTF data prediction. Sequence length, derived from the temporal view, is closely linked to the periodic view as it determines the number of time nodes of VTF data input into the training network, which corresponds to the number of CNNs in the periodic view. For this analysis, the other three network parameters—local VTF matrix size, learning rate, and iterations—are fixed at 3 × 3, 0.001, and 500, respectively. The quantitative results, including RMSE, MAPE, and R^2, for five sequence lengths are presented in Tables V and VI in the Appendix. The findings highlight that the MPTNSR model achieves optimal prediction performance with a sequence length of 3 across both water areas. This indicates that the model effectively captures complex changing features from historical data in various application scenarios. Thus, selecting a shorter sequence length of 3 allows the MPTNSR method to optimally predict by efficiently capturing essential features from past data while avoiding overloading the training network.

The quantitative evaluation results, including RMSE, MAPE, and R^2, indicate that the ideal values for the size of the local VTF matrix, learning rate, iteration, and sequence length are 3 × 3, 0.001, 500, and 3, respectively. These conclusions derived from the experiments can serve as the basis for setting the network parameters in subsequent comparative experiments. By utilising these optimal parameter values, the subsequent experiments can ensure better prediction accuracy and an improved fitting effect of the training network.

A key aspect of the MPTNSR method is the optimisation of network parameters during the training process, as these parameters significantly influence the final prediction results. Through quantitative experiments, this section identifies the optimal values for key parameters, including the local VTF matrix size, learning rate, number of iterations, and sequence length. Ensuring consistency in parameter settings is essential for fairness and reliability in comparative experiments, particularly when the proposed MPTNSR model shares overlapping parameters with other methods. The detailed parameter configurations for the comparison methods are summarised in Table 8.

CNN and BiLSTM construct periodic and temporal views of the proposed MPTNSR model. Hence, the network parameters of RNNs (i.e., RNN, LSTM, GRU, Bi-LSTM, and Bi-GRU), CNN-LSTM, CNN-BiLSTM, CNN-GRU, CNN-BiGRU, LSTM-GCN, BiBiLSTM-GCN, GRU-GCN, and BiGRU-GCN are consistent with the proposed MPTNSR model. In particular, there is a different meaning in the parameter 'Input Size' between RNNs and the proposed MPTNSR model. In the MPTNSR model, each network unit is designed to accept a singular VTF data input. Contrastingly, for RNNs, the 'Input Size' signifies the count of VTF data inputs each network unit processes. The 'Input Size' parameters in BPNN and WNN correspond with the 'Sequence Length' parameter seen in both the MPTNSR model and RNNs. This

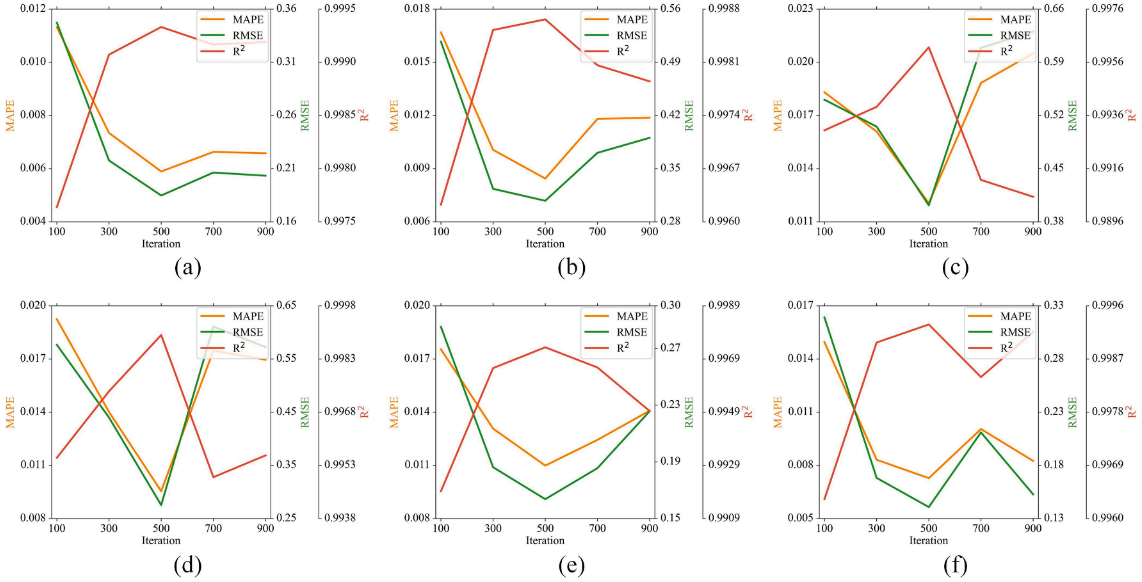
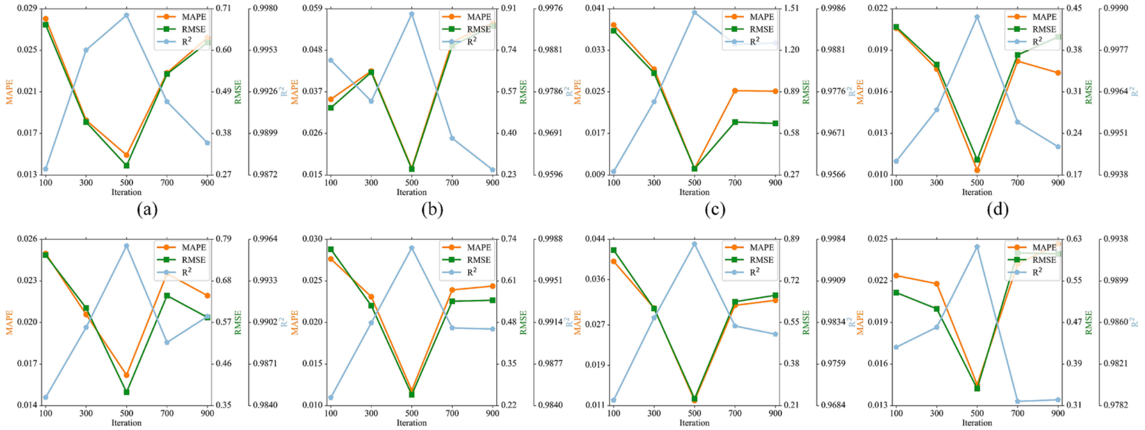


Fig. 10. The prediction results of the proposed MPTNSR model for three indices (i.e., RMSE, MAPE, and R^2) across five different iterations (i.e., 100, 300, 500, 700, and 900) in the CFDP water area. (a) − (f) indicate datasets $C_1 - C_6$, respectively.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 11. The prediction results of the proposed MPTNSR model for three indices (i.e., RMSE, MAPE, and R^2) across five different iterations (i.e., 100, 300, 500, 700, and 900) in the CFDP water area. (a) – (f) indicate datasets CJ$_1$ – CJ$_8$, respectively.

indicates the consecutive time nodes of VTF data are fed into the training network.

5.5. *Comparative analysis of loss functions and information weights*

The strength of the proposed MPTNSR method lies not only in constructing three views but also in optimising the loss function. The optimisation involves incorporating a correction term utilising PCC on top of MSE to effectively evaluate the discrepancy between predicted and actual values within a particular sequence. The resulting loss function is referred to as MSEPCC, while the original remains MSE. This section evaluates the advantages of the optimised loss function through quantitative comparison experiments.

Additionally, this section investigates the inclusion of a weight parameter in the semantic view. This parameter controls how much information regarding VTF data similarity between channels and geographical location relationships is preserved in the VTF correlation matrix.

Tables 9 and 10 compare the prediction performance of the two loss functions, MSEPCC and MSE, across all data nodes in two water areas for the next day. The results consistently show that MSEPCC achieves the highest R^2 and the lowest RMSE and MAPE across all VTF datasets. These findings highlight that the optimised loss function significantly improves the training network's fitting capability and enhances prediction accuracy. The advantage is particularly pronounced due to the inclusion of PCC as a correction term, which

Table 8
Network parameter settings for eighteen comparative prediction methods.

Methods		Parameters	Values	Parameters	Values
BPNN and WNN		Optimisation algorithm	Adamax	Learning rate	0.001
		Iteration	500	Input size	3
		Hidden size	2 × Input Size	Output Size	1
RNN, LSTM, GRU, Bi-LSTM, and Bi-GRU		Optimisation algorithm	Adamax	Learning rate	0.001
		Iteration	500	Sequence length	3
		Input size	1	Hidden size	6
		Output Size	1	Number of hidden layer	1
CNN-LSTM, CNN-BiLSTM. CNN-GRU, and CNN-BiGRU	CNN	Size of local VTF matrix	3 × 3	Number of convolutional layers	4
		Size of convolutional kernel	3 × 3	Stride	1
		Padding	1	–	–
	LSTM, GRU, Bi-LSTM or Bi-GRU	Sequence length	3	Input size	32
		Hidden size	2 × Input size	Number of hidden layer	1
	General Parameters	Optimisation algorithm	Adamax	Learning rate	0.001
		Iteration	500	Output Size	1
LSTM-GCN, BiLSTM-GCN, GRU-GCN, and BiGRU-GCN	LSTM, GRU, Bi-LSTM or Bi-GRU	Sequence length	3	Input size	32
		Hidden size	2 × Input size	Number of hidden layer	1
	GCN	Number of convolutional layers	2	The output dimension of the First Layer Convolution	23
		The output dimension of the second layer convolution	6	–	–
	General Parameters	Optimisation algorithm	Adamax	Learning rate	0.001
		Iteration	500	Output Size	1

effectively strengthens the model's performance.

This experiment examines the impact of three information weight combinations (0.5/0.5, 0.8/0.2, and 1.0/0.0) on prediction accuracy and stability using the same loss function, MSEPCC. Tables 11 and 12 summarise the prediction performance for all data nodes in the CFDP and CJ water areas for the next day. The results show that the 0.8/0.2 weight combination consistently achieves the lowest RMSE and MAPE values and the highest R^2 values across all scenarios, indicating it enables the best model fitting.

Figs. 12 and 13 visualise the directed graphs of relationships between channels in both water areas based on a VTF incidence matrix weighted at 0.8/0.2. These graphs, used as input data for the semantic view, play a critical role in capturing the relational features of VTF changes between channels. This enhances the accuracy and consistency of the MPTNSR method for VTF collaborative prediction tasks across various channels. The comparisons confirm that the 0.8/0.2 weighting effectively balances information, resulting in superior performance.

5.6. Comparative analysis of experimental results

5.6.1. Experimental analysis of nineteen prediction methods

This section delves into a performance analysis of the proposed MPTNSR model relative to eighteen other techniques for VTF prediction challenges. Specifically, it highlights the proficiency of the MPTNSR model in collaboratively predicting VTF across different target regions.

Fig. 14 visualises the REMean results for 19 different methods applied to datasets C_1 to C_6 in the CFDP water area, with panels (a) to (f) corresponding to each dataset. This experiment evaluates the accuracy and consistency of prediction at each time node, comparing the REMean values for each method.

NN methods such as BPNN and WNN, and DL methods, including RNN, LSTM, GRU, Bi-LSTM, Bi-GRU, CNN-BiLSTM, and CNN-BiGRU, consistently outperform traditional machine learning approaches such as ARIMA and SVM in terms of prediction accuracy. DL methods exhibit superior prediction efficiency and robustness compared to NN methods, particularly in handling complex datasets like C_1 to C_6.

Among the evaluated DL methods evaluated, CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU show noticeable improvement over traditional NN techniques but are still outperformed by graph-based models like LSTM-GCN, GRU-GCN, BiLSTM-GCN, and BiGRU-GCN. Hybrid models like CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU improve spatial and temporal feature extraction but fail to capture complex interdependencies between different vessel traffic channels, leading to suboptimal predictions in congested or high-density maritime networks. In contrast, graph-based models (LSTM-GCN, GRU-GCN, BiLSTM-GCN, and BiGRU-GCN) improve relational learning; however, they lack explicit spatial modelling, limiting their ability to differentiate between localised vessel interactions, navigational constraints, and broader maritime patterns.

Despite the advancements offered by these models, the proposed MPTNSR method demonstrates the best overall performance, achieving the lowest REMean values in most cases.

The superior performance of the MPTNSR method underscores the critical importance of the semantic view in improving prediction accuracy. By effectively capturing the relationships and dependencies between VTF changes across channels, the semantic view provides a significant edge over competing models. The MPTNSR method exhibits remarkable consistency and stability in prediction across datasets C_1, C_2, C_3, and C_4. This stability is particularly evident in the lower variability of REMean values compared to other methods, highlighting its reliability in collaborative prediction tasks.

The quantitative results from Fig. 14 confirm the effectiveness of the proposed MPTNSR model in achieving high prediction

Table 9
The prediction results of the proposed MPTNSR model based on different indexes with/without optimised loss function (MSEPCC) in the CFDP water area.

Dataset	Performance Indexes	Loss Function	
		MSEPCC	MSE
C_1	MAPE↓	**0.0056**	0.0197
	RMSE↓	**0.1604**	0.8113
	R^2↑	**0.9994**	0.9871
C_2	MAPE↓	**0.0062**	0.0115
	RMSE↓	**0.1712**	0.3625
	R^2↑	**0.9995**	0.9981
C_3	MAPE↓	**0.0089**	0.0176
	RMSE↓	**0.3188**	0.6573
	R^2↑	**0.9975**	0.9897
C_4	MAPE↓	**0.0082**	0.0246
	RMSE↓	**0.2639**	0.5645
	R^2↑	**0.9990**	0.9957
C_5	MAPE↓	**0.0108**	0.0100
	RMSE↓	**0.1568**	0.1294
	R^2↑	**0.9975**	0.9983
C_6	MAPE↓	**0.0041**	0.0053
	RMSE↓	**0.0805**	0.1198
	R^2↑	**0.9997**	0.9994

Table 10
The prediction results of the proposed MPTNSR model based on different indexes with/without optimised loss function (MSEPCC) in the CJ water area.

Dataset	Performance Indexes	Loss Function	
		MSEPCC	MSE
CJ$_1$	MAPE↓	**0.0101**	0.0113
	RMSE↓	**0.1841**	0.2187
	R^2↑	**0.9990**	0.9986
CJ$_2$	MAPE↓	**0.0107**	0.0124
	RMSE↓	**0.1563**	0.1911
	R^2↑	**0.9986**	0.9979
CJ$_3$	MAPE↓	**0.0098**	0.0085
	RMSE↓	**0.3119**	0.2123
	R^2↑	**0.9977**	0.9989
CJ$_4$	MAPE↓	**0.0084**	0.0098
	RMSE↓	**0.1770**	0.2208
	R^2↑	**0.9989**	0.9984
CJ$_5$	MAPE↓	**0.0146**	0.0188
	RMSE↓	**0.3895**	0.5446
	R^2↑	**0.9958**	0.9918
CJ$_6$	MAPE↓	**0.0103**	0.0114
	RMSE↓	**0.2207**	0.2549
	R^2↑	**0.9985**	0.9980
CJ$_7$	MAPE↓	**0.0092**	0.0123
	RMSE↓	**0.1629**	0.2312
	R^2↑	**0.9988**	0.9977
CJ$_8$	MAPE↓	**0.0110**	0.0111
	RMSE↓	**0.2402**	0.2938
	R^2↑	**0.9966**	0.9949

Table 11
The prediction results of the proposed MPTNSR model based on three different information weights in the CFDP water area.

Dataset	Performance Indexes	Information Weights		
		0.5, 0.5	0.8, 0.2	1.0, 0.0
C$_1$	RMSE↓	0.0050	**0.0056**	0.0058
	MAPE↓	0.1301	**0.1604**	0.1735
	R^2↑	0.9996	**0.9994**	0.9994
C$_2$	RMSE↓	0.0067	**0.0062**	0.0087
	MAPE↓	0.2287	**0.1712**	0.2880
	R^2↑	0.9992	**0.9995**	0.9988
C$_3$	RMSE↓	0.0097	**0.0089**	0.0103
	MAPE↓	0.3249	**0.3188**	0.3698
	R^2↑	0.9974	**0.9975**	0.9967
C$_4$	RMSE↓	0.0235	**0.0082**	0.0262
	MAPE↓	0.4760	**0.2639**	0.5480
	R^2↑	0.9969	**0.9990**	0.9959
C$_5$	RMSE↓	0.0132	**0.0108**	0.0652
	MAPE↓	0.1628	**0.1568**	0.8238
	R^2↑	0.9973	**0.9975**	0.9326
C$_6$	RMSE↓	0.0054	**0.0041**	0.0066
	MAPE↓	0.1183	**0.0805**	0.1593
	R^2↑	0.9994	**0.9997**	0.9990

accuracy and stability. It outperforms traditional machine learning methods, NN approaches, and even other advanced DL models, particularly through the integration of the semantic view. This highlights the robustness of the MPTNSR method, making it a reliable choice for complex VTF prediction tasks in the CFDP water area.

Fig. 15 presents the REMean results for 19 different methods applied to datasets CJ$_1$ to CJ$_8$ in the CJ water area, with subplots (a) to (h) corresponding to each dataset. The performance of these methods is compared to assess prediction accuracy and stability, particularly focusing on the proposed MPTNSR method.

Methods such as ARIMA and SVM exhibit the highest REMean values across all datasets, indicating poor prediction accuracy and stability. These methods struggle to capture complex, dynamic features in VTF data, making them unsuitable for such tasks. Basic NN models like BPNN and WNN show better performance than ARIMA and SVM but still lag behind DL methods. Their higher REMean values indicate limitations in handling temporal dependencies and complex spatial relationships. DL-based approaches, including RNN, LSTM, GRU, Bi-LSTM, and Bi-GRU, demonstrate significantly lower REMean values compared to NN methods. This reflects their ability to better model temporal dependencies in VTF data. Advanced hybrid models such as CNN-LSTM, CNN-BiLSTM, CNN-GRU, and

Table 12
The prediction results of the proposed MPTNSR model based on three different information weights in the CJ water area.

Dataset	Performance Indexes	Information Weights		
		0.5, 0.5	0.8, 0.2	1.0, 0.0
CJ_1	MAPE↓	0.0190	**0.0101**	0.0210
	RMSE↓	0.3765	**0.1841**	0.4326
	R^2↑	0.9960	**0.9990**	0.9947
CJ_2	MAPE↓	0.0309	**0.0107**	0.0214
	RMSE↓	0.4204	**0.1563**	0.3248
	R^2↑	0.9901	**0.9986**	0.9941
CJ_3	MAPE↓	0.0227	**0.0098**	0.0102
	RMSE↓	0.6909	**0.3119**	0.3165
	R^2↑	0.9887	**0.9977**	0.9976
CJ_4	MAPE↓	0.0153	**0.0084**	0.0103
	RMSE↓	0.3165	**0.1770**	0.1956
	R^2↑	0.9967	**0.9989**	0.9987
CJ_5	MAPE↓	0.0193	**0.0146**	0.0162
	RMSE↓	0.4686	**0.3895**	0.3856
	R^2↑	0.9939	**0.9958**	0.9959
CJ_6	MAPE↓	0.0185	**0.0103**	0.0118
	RMSE↓	0.4111	**0.2207**	0.2540
	R^2↑	0.9948	**0.9985**	0.9980
CJ_7	MAPE↓	0.0184	**0.0092**	0.0153
	RMSE↓	0.3248	**0.1629**	0.2684
	R^2↑	0.9954	**0.9988**	0.9969
CJ_8	MAPE↓	0.0218	**0.0110**	0.0140
	RMSE↓	0.5218	**0.2402**	0.3389
	R^2↑	0.9840	**0.9966**	0.9932

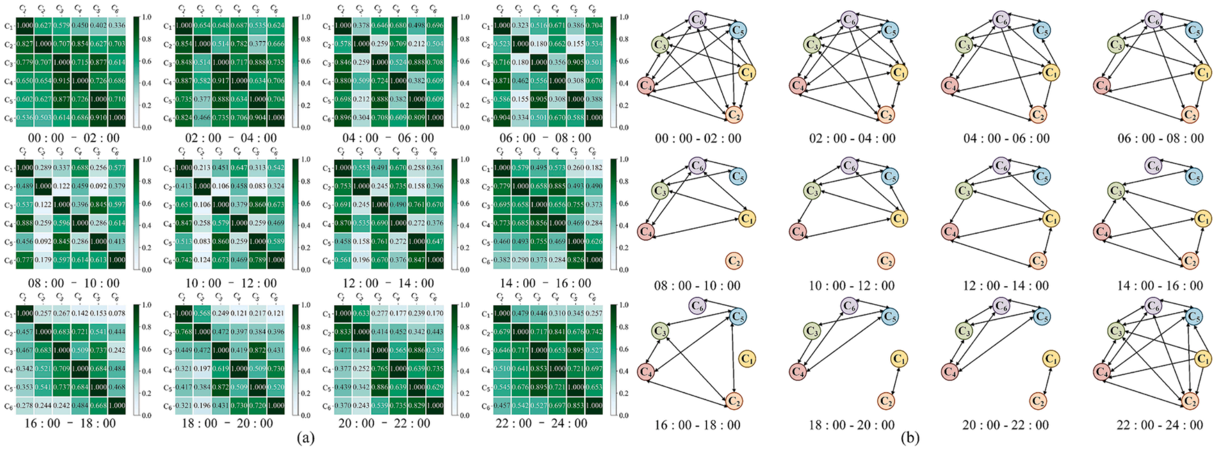


Fig. 12. Visualisation of the VTF similarity matrix and the relationships between six channels in the CFDP water area with information weight values of 0.8 and 0.2, (a) similarity matrix image and (b) the corresponding directed relationship graph.

CNN-BiGRU further improve performance by integrating convolutional layers to extract spatial features alongside temporal modelling. Models like LSTM-GCN, GRU-GCN, BiLSTM-GCN, and BiGRU-GCN show further reductions in REMean values. By leveraging graph structures, these models effectively capture the relational features of VTF data between different channels, improving both accuracy and stability.

The proposed MPTNSR method consistently achieves the lowest REMean values across all datasets (CJ_1 to CJ_8). This demonstrates its superior ability to handle complex spatiotemporal dependencies and relational dynamics between channels. Its performance advantage is particularly evident in datasets CJ_3, CJ_4, CJ_6, and CJ_7, where it maintains both accuracy and stability better than all other methods.

The MPTNSR method outperforms all comparison methods in terms of REMean, achieving the highest prediction accuracy across all datasets while exhibiting minimal variability across time points, which highlights its stability in prediction. This consistent performance underscores the critical role of the semantic view in effectively capturing channel relationships and enhancing prediction outcomes.

Based on the results in Tables 13 and 14, the performance of 19 different prediction models for VTF data is compared across the CFDP and CJ water areas using three key metrics: RMSE, MAPE, and R^2. These metrics evaluate each method's capability to holistically

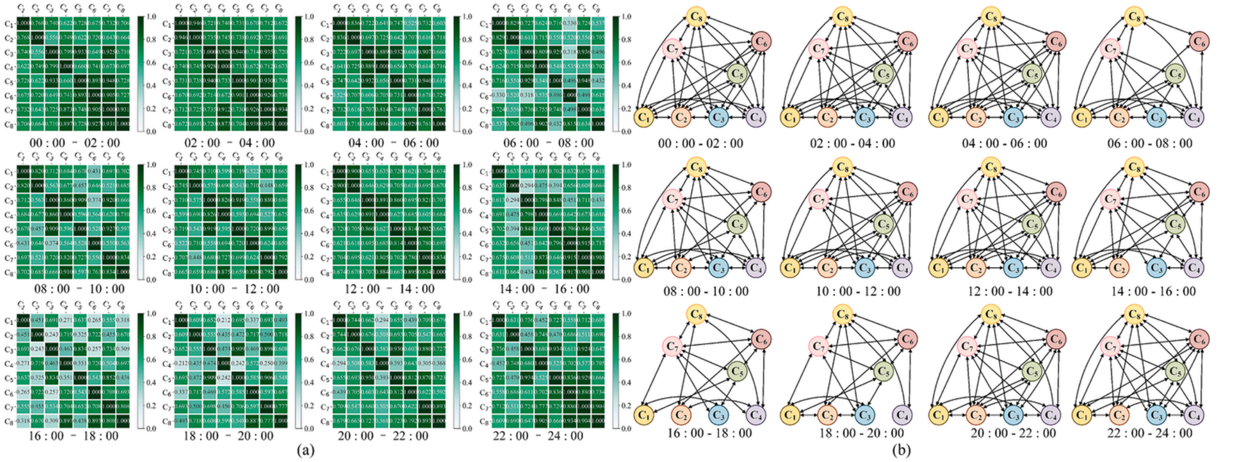


Fig. 13. Visualisation of the VTF similarity matrix and the relationships between eight channels in the CJ water area with information weight values of 0.8 and 0.2, (a) similarity matrix image and (b) the corresponding directed relationship graph.

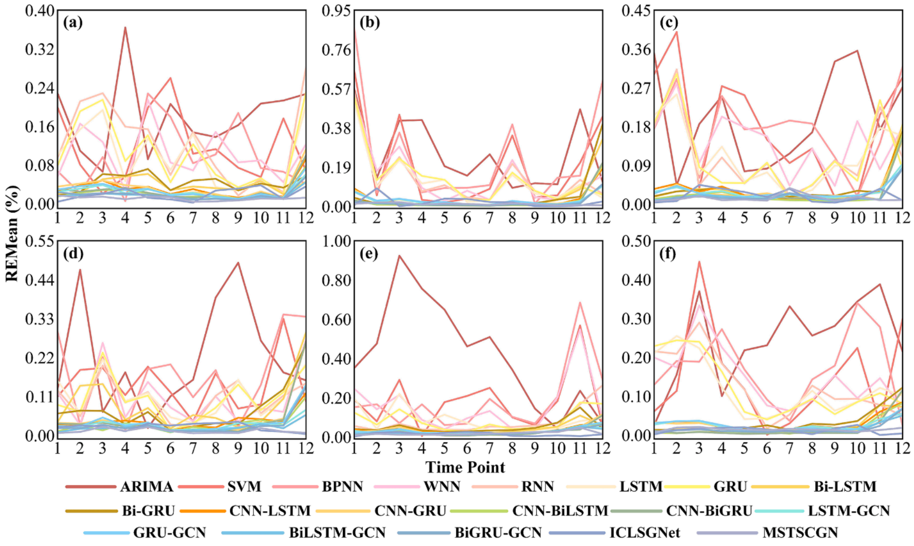


Fig. 14. The visualisation results of REMean using 19 different comparison methods in the CFDP water area. (a) – (f) indicate datasets C_1 – C_6, respectively.

predict the next day's VTF data, encompassing 12 time nodes.

In the CFDP water area, traditional methods like ARIMA and SVM exhibit high error rates and inconsistent performance. While SVM generally outperforms ARIMA in terms of MAPE and RMSE, its R^2 scores indicate limited predictability across several datasets. NN methods, including BPNN and WNN, demonstrate improved performance over traditional models, particularly in terms of RMSE values, though their consistency across datasets remains limited.

Advanced DL models such as RNN, LSTM, GRU, and their bidirectional variants (Bi-LSTM and Bi-GRU) achieve significantly better performance, particularly in datasets C_1 to C_6, where R^2 values suggest strong predictive capabilities. Among these, Bi-GRU often outperforms its counterparts due to its ability to capture bidirectional temporal dynamics.

Hybrid models like CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU integrate convolutional and recurrent layers, allowing them to leverage both spatial and temporal features. These models deliver competitive results, demonstrating relatively low RMSE and high R^2 scores. Graph-based models such as LSTM-GCN, GRU-GCN, BiLSTM-GCN, and BiGRU-GCN further enhance performance, particularly in reducing RMSE, by capturing relational dependencies between channels.

The proposed MPTSRNet model consistently achieves the lowest MAPE and RMSE values and the highest R^2 scores across all datasets. Notably, in datasets C_1, C_3, and C_6, the MPTSRNet model's MAPE values are significantly lower—often an order of magnitude—compared to other models, and its R^2 scores are nearly 1, indicating exceptional predictability. This superior performance is attributed to MPTSRNet's integration of multiple network architectures and its advanced feature extraction capabilities, enabling it to

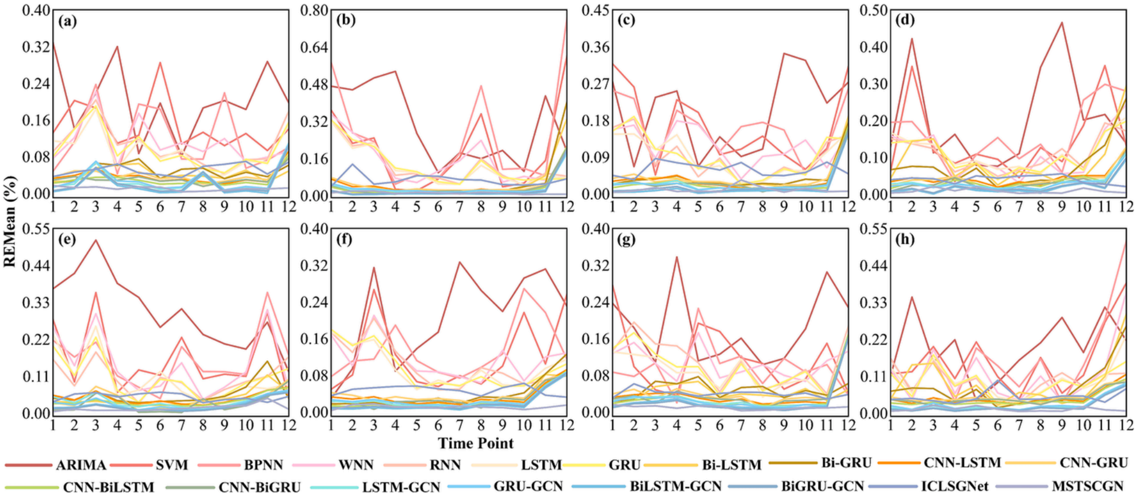


Fig. 15. The visualisation results of REMean using 19 different comparison methods in the CJ water area. (a) – (h) indicate datasets CJ_1 – CJ_8, respectively.

effectively capture complex nonlinear relationships in VTF data.

In the CJ water area, traditional methods like ARIMA and SVM struggle with high error rates and lower R^2 scores, reflecting limited accuracy and predictability. Neural network methods such as RNN, LSTM, GRU, and their bidirectional variants (Bi-LSTM and Bi-GRU) perform significantly better, delivering lower errors and higher R^2 values. Among these, Bi-GRU and Bi-LSTM show notable improvements due to their ability to model bidirectional temporal dependencies.

Hybrid models, including CNN-BiLSTM and CNN-BiGRU, and graph-based models like LSTM-GCN and BiLSTM-GCN, further enhance performance by capturing both spatial and temporal relationships. These models consistently achieve lower RMSE values and higher R^2 scores compared to traditional and standard neural network methods.

The MPTSRNet model once again demonstrates the best performance, achieving the lowest MAPE and RMSE values and the highest R^2 scores across all datasets in the CJ water area. Its robustness and precision highlight its ability to effectively integrate spatio-temporal and semantic features, making it highly reliable for complex predictive tasks. For instance, in datasets CJ_3, CJ_4, CJ_6, MPTSRNet significantly outperforms all other models, underscoring its advanced architecture and capability to adapt to diverse datasets.

The MPTSRNet model's superior performance in both the CFDP and CJ water areas demonstrates its ability to handle complex and dynamic VTF prediction tasks with exceptional accuracy and consistency. This makes it a valuable tool for both theoretical research and practical applications, setting a new benchmark for predictive models in intelligent maritime transportation systems.

Furthermore, the MPTSRNet model provides several key advantages:

(1) Stronger semantic learning: Compared to CNN-LSTM, CNN-BiLSTM, CNN-GRU, and CNN-BiGRU, MPTNSR models vessel interactions, traffic dependencies, and navigational constraints, essential for port congestion and regulated waterways.
(2) Holistic feature fusion: By combining spatial, temporal, and semantic modelling, MPTNSR provides context-aware vessel trajectory prediction, showing superiority over LSTM-GCN, GRU-GCN, BiLSTM-GCN, and BiGRU-GCN in terms of a dedicated spatial feature extractor.
(3) Improved stability and generalisation: The MPTNSR model achieves lower REMean values across all datasets, particularly excelling in high-traffic areas, where other methods struggle with fluctuating traffic conditions.

Empirical results confirm MPTNSR's superior accuracy and stability compared to all other models. It maintains low prediction errors, minimal variability, and robust learning of vessel traffic dependencies, making it highly reliable for maritime traffic forecasting and navigation planning.

MPTNSR's integration of spatial, temporal, and semantic learning sets it apart from existing hybrid models. By capturing vessel interactions, adapting to dynamic maritime environments, and addressing localised navigation constraints, it offers a more reliable and accurate approach to maritime traffic prediction.

5.6.2. Model performance validation

The evaluation of the MPTNSR model's predictive performance is conducted by analysing the downward trends of loss function curves across two distinct water regions and comparing the loss values for each. Figs. 16 and 17 visually depict this reduction, showing a consistent decline in both areas. The *x*-axis of these graphs represents the number of iterations, while the *y*-axis details the corresponding loss values.

Figs. 16 and 17 illustrate the loss function curves for the proposed MPTSRNet model during training on datasets from the CFDP and

Table 13
Prediction results of VTF data based on 19 different models under 3 indicators in the CFDP Water area.

Dataset	Performance Indexes	Methods										
		ARIMA	SVM	BPNN	WNN	RNN	LSTM	GRU	Bi-LSTM	Bi-GRU	CNN–BiLSTM	CNN–BiGRU
C_1	MAPE↓	0.1799	0.1168	0.0950	0.0996	0.1223	0.0962	0.1047	0.0341	0.0404	0.0217	0.0251
	RMSE↓	4.7187	3.3929	3.4068	2.9611	3.1376	2.4865	2.6710	0.9761	1.1362	0.6254	0.7690
	R^2↑	0.5638	0.7745	0.7726	0.8282	0.8071	0.8788	0.8602	0.9813	0.9747	0.9923	0.9884
C_2	MAPE↓	0.2526	0.2012	0.2443	0.1387	0.1303	0.1223	0.1302	0.0435	0.0476	0.0241	0.0245
	RMSE↓	6.5360	4.5252	5.1693	3.1249	3.0032	2.8199	3.0002	1.2967	1.4494	0.7403	0.7523
	R^2↑	0.3975	0.7112	0.6231	0.8622	0.8727	0.8878	0.8730	0.9762	0.9703	0.9922	0.9920
C_3	MAPE↓	0.2003	0.1811	0.1747	0.1402	0.1101	0.1063	0.1060	0.0344	0.0363	0.0257	0.0254
	RMSE↓	4.6699	3.3670	3.5524	2.7530	2.0342	1.9773	2.0271	0.9391	0.9024	0.7840	0.7607
	R^2↑	0.4810	0.7302	0.6996	0.8196	0.9015	0.9069	0.9022	0.9790	0.9806	0.9853	0.9862
C_4	MAPE↓	0.2174	0.1430	0.1724	0.0971	0.0966	0.1064	0.1004	0.0835	0.0666	0.0291	0.0387
	RMSE↓	5.7014	3.2710	3.8969	2.5262	2.3128	2.3035	2.2624	1.6382	1.2077	0.5833	0.8172
	R^2↑	0.5618	0.8557	0.7953	0.9139	0.9278	0.9284	0.9310	0.9638	0.9803	0.9954	0.9909
C_5	MAPE↓	0.4128	0.1802	0.1856	0.1470	0.0659	0.0781	0.0711	0.0306	0.0470	0.0224	0.0272
	RMSE↓	4.4677	2.2681	2.3933	1.9163	0.9015	0.8391	0.7972	0.4365	0.6051	0.2768	0.3643
	R^2↑	0.3109	0.4894	0.4315	0.6355	0.9193	0.9301	0.9369	0.9810	0.9636	0.9923	0.9868
C_6	MAPE↓	0.2427	0.1454	0.1680	0.1435	0.1266	0.1194	0.1215	0.0227	0.0328	0.0148	0.0160
	RMSE↓	4.3053	2.5643	3.0714	2.5362	2.1642	1.8365	1.9509	0.3857	0.5339	0.2990	0.3277
	R^2↑	0.3318	0.7629	0.6599	0.7681	0.8311	0.8784	0.8628	0.9946	0.9897	0.9967	0.9961

Dataset	Performance Indexes	Methods							
		CNN-LSTM	CNN-GRU	LSTM-GCN	GRU-GCN	BiLSTM-GCN	BiGRU-GCN	ICLSGNet	MPTSRNet
C_1	MAPE↓	0.0317	0.0347	0.0248	0.0274	0.0128	0.0150	0.0194	**0.0056**
	RMSE↓	0.8378	0.9467	0.6215	0.6989	0.3537	0.4273	0.6487	**0.1604**
	R^2↑	0.9862	0.9824	0.9924	0.9904	0.9975	0.9964	0.9917	**0.9994**
C_2	MAPE↓	0.0380	0.0383	0.0294	0.0311	0.0144	0.0151	0.0206	**0.0062**
	RMSE↓	0.8231	0.8360	0.5944	0.6234	0.3790	0.3892	0.7768	**0.1712**
	R^2↑	0.9904	0.9901	0.9950	0.9945	0.9979	0.9978	0.9914	**0.9995**
C_3	MAPE↓	0.0368	0.0339	0.0293	0.0285	0.0151	0.0148	0.0197	**0.0089**
	RMSE↓	0.8536	0.7948	0.5933	0.5753	0.4322	0.4344	0.4375	**0.3188**
	R^2↑	0.9826	0.9849	0.9916	0.9921	0.9955	0.9955	0.9954	**0.9975**
C_4	MAPE↓	0.0409	0.0490	0.0307	0.0370	0.0171	0.0244	0.0235	**0.0082**
	RMSE↓	0.7718	0.9017	0.6023	0.6885	0.3521	0.4736	0.6735	**0.2639**
	R^2↑	0.9919	0.9890	0.9951	0.9936	0.9983	0.9969	0.9938	**0.9990**
C_5	MAPE↓	0.0333	0.0337	0.0287	0.0293	0.0134	0.0146	0.0136	**0.0108**
	RMSE↓	0.3539	0.4083	0.3081	0.3349	0.1920	0.2112	0.1605	**0.1568**
	R^2↑	0.9875	0.9834	0.9905	0.9888	0.9963	0.9955	0.9974	**0.9975**
C_6	MAPE↓	0.0297	0.0276	0.0262	0.0263	0.0104	0.0098	0.0119	**0.0041**
	RMSE↓	0.4270	0.4157	0.3784	0.3833	0.1715	0.1802	0.2616	**0.0805**
	R^2↑	0.9934	0.9937	0.9948	0.9947	0.9989	0.9988	0.9975	**0.9997**

Table 14
Prediction results of VTF data based on 19 different models under 3 indicators in the CJ Water area.

Dataset	Performance Indexes	Methods										
		ARIMA	SVM	BPNN	WNN	RNN	LSTM	GRU	Bi-LSTM	Bi-GRU	CNN–BiLSTM	CNN–BiGRU
CJ_1	MAPE↓	0.2020	0.1455	0.1175	0.1086	0.1049	0.0922	0.0965	0.0413	0.0469	0.0296	0.0318
	RMSE↓	3.9798	2.7300	2.7540	2.1264	1.7780	1.6117	1.6551	0.8103	0.9298	0.5640	0.6106
	R^2↑	0.5589	0.7924	0.7887	0.8740	0.9119	0.9276	0.9237	0.9817	0.9759	0.9911	0.9896
CJ_2	MAPE↓	0.2992	0.1928	0.2515	0.1702	0.1284	0.1218	0.1284	0.0506	0.0554	0.0328	0.0351
	RMSE↓	4.0895	2.4749	3.1697	2.0680	1.6278	1.5411	1.6160	0.7790	0.8750	0.4788	0.5000
	R^2↑	0.0694	0.6591	0.4409	0.7620	0.8525	0.8678	0.8546	0.9662	0.9573	0.9872	0.9860
CJ_3	MAPE↓	0.1997	0.1645	0.1612	0.1195	0.0913	0.0913	0.0938	0.0388	0.0415	0.0293	0.0315
	RMSE↓	4.9887	3.4306	3.5266	2.4503	1.7956	1.7331	1.7884	1.0926	1.0728	0.9118	0.9073
	R^2↑	0.4154	0.7235	0.7078	0.8589	0.9242	0.9294	0.9248	0.9719	0.9729	0.9804	0.9806
CJ_4	MAPE↓	0.2042	0.1443	0.1614	0.1097	0.1088	0.1129	0.1077	0.0901	0.0765	0.0353	0.0330
	RMSE↓	3.8807	2.2295	2.4523	1.6875	1.6287	1.6836	1.6005	1.4618	1.1650	0.5660	0.5538
	R^2↑	0.5057	0.8368	0.8026	0.9065	0.9129	0.9069	0.9159	0.9298	0.9554	0.9894	0.9899
CJ_5	MAPE↓	0.3016	0.1605	0.1594	0.1440	0.1015	0.1019	0.1035	0.0431	0.0539	0.0359	0.0293
	RMSE↓	5.7008	3.3911	3.3883	2.8916	1.8905	1.8730	1.8983	0.9753	1.2546	0.7288	0.6668
	R^2↑	0.1075	0.6842	0.6847	0.7703	0.9018	0.9036	0.9010	0.9738	0.9567	0.9854	0.9877
CJ_6	MAPE↓	0.2066	0.1214	0.1320	0.1144	0.1015	0.0962	0.0957	0.0282	0.0375	0.0242	0.0292
	RMSE↓	4.5956	2.4741	2.6552	2.2338	2.0325	1.8053	1.8358	0.5421	0.7453	0.4860	0.5704
	R^2↑	0.3578	0.8138	0.7856	0.8482	0.8743	0.9009	0.8975	0.9910	0.9831	0.9928	0.9901
CJ_7	MAPE↓	0.1828	0.1132	0.0954	0.1061	0.1089	0.0888	0.0970	0.0410	0.0473	0.0327	0.0377
	RMSE↓	3.3532	2.3787	2.0723	1.9274	1.8427	1.5806	1.6790	0.7660	0.9019	0.8021	0.8304
	R^2↑	0.5192	0.7580	0.8163	0.8411	0.8548	0.8931	0.8794	0.9749	0.9652	0.9724	0.9705
CJ_8	MAPE↓	0.1928	0.1579	0.1630	0.1157	0.0946	0.1005	0.0908	0.0898	0.0765	0.0393	0.0443
	RMSE↓	3.6088	2.8993	3.0291	2.0924	1.7058	1.7154	1.6506	1.5532	1.3567	0.7034	0.8127
	R^2↑	0.2363	0.5071	0.4620	0.7432	0.8293	0.8274	0.8402	0.8585	0.8920	0.9709	0.9612

Dataset	Performance Indexes	Methods							
		CNN-LSTM	CNN-GRU	LSTM-GCN	GRU-GCN	BiLSTM-GCN	BiGRU-GCN	ICLSGNet	MPTSRNet
CJ_1	MAPE↓	0.0371	0.0396	0.0276	0.0309	0.0204	0.0236	0.0200	**0.0101**
	RMSE↓	0.6777	0.7300	0.6019	0.6527	0.5481	0.5855	0.4517	**0.1841**
	R^2↑	0.9872	0.9851	0.9899	0.9881	0.9916	0.9904	0.9943	**0.9990**
CJ_2	MAPE↓	0.0435	0.0463	0.0331	0.0367	0.0232	0.0265	0.0309	**0.0107**
	RMSE↓	0.5346	0.5615	0.4369	0.4676	0.4028	0.4237	0.4753	**0.1563**
	R^2↑	0.9840	0.9824	0.9893	0.9878	0.9909	0.9900	0.9874	**0.9986**
CJ_3	MAPE↓	0.0367	0.0390	0.0287	0.0285	0.0177	0.0213	0.0381	**0.0098**
	RMSE↓	0.9614	0.9620	0.8363	0.8437	0.7746	0.8117	0.9264	**0.3119**
	R^2↑	0.9782	0.9782	0.9835	0.9832	0.9859	0.9845	0.9798	**0.9977**
CJ_4	MAPE↓	0.0447	0.0419	0.0327	0.0305	0.0185	0.0219	0.0197	**0.0084**
	RMSE↓	0.6847	0.6591	0.5020	0.4769	0.3390	0.3794	0.4066	**0.1770**
	R^2↑	0.9846	0.9857	0.9917	0.9925	0.9962	0.9952	0.9945	**0.9989**
CJ_5	MAPE↓	0.0438	0.0382	0.0352	0.0290	0.0234	0.0207	0.0361	**0.0146**
	RMSE↓	0.8293	0.7549	0.6591	0.5796	0.5060	0.4656	0.7611	**0.3895**
	R^2↑	0.9811	0.9843	0.9880	0.9907	0.9929	0.9940	0.9840	**0.9958**
CJ_6	MAPE↓	0.0329	0.0371	0.0260	0.0282	0.0182	0.0208	0.0208	**0.0103**
	RMSE↓	0.6057	0.6826	0.4936	0.5285	0.4010	0.4346	0.4959	**0.2207**
	R^2↑	0.9888	0.9858	0.9925	0.9915	0.9951	0.9942	0.9925	**0.9985**
CJ_7	MAPE↓	0.0394	0.0436	0.0330	0.0310	0.0268	0.0237	0.0225	**0.0092**
	RMSE↓	0.8554	0.8884	0.7819	0.7558	0.7421	0.7151	0.4631	**0.1629**
	R^2↑	0.9687	0.9662	0.9738	0.9755	0.9764	0.9781	0.9908	**0.9988**
CJ_8	MAPE↓	0.0454	0.0489	0.0247	0.0272	0.0215	0.0202	0.0185	**0.0110**
	RMSE↓	0.7915	0.8777	0.4689	0.5063	0.4295	0.4102	0.4111	**0.2402**
	R^2↑	0.9632	0.9548	0.9871	0.9849	0.9891	0.9901	0.9900	**0.9966**

CJ water areas, respectively. In both figures, the loss curves across all datasets (C_1-C_6 for CFDP in Fig. 16 and CJ_1-CJ_8 in Fig. 17) demonstrate a consistent decrease in loss values as the number of iterations increases, which is a clear indicator of the model's good convergence.

In Fig. 16, datasets C_1 through C_6 show a steady and rapid decline in loss values early in the training process, with the curves flattening out as they approach zero, suggesting that the model effectively minimises the error without signs of bouncing or instability. Similarly, Fig. 17 shows the loss curves for datasets CJ_1 through CJ_8, also decreasing smoothly and stabilising at very low values. This pattern indicates that the model not only learns effectively from the training data but also reaches a point of stability, which mitigates concerns about overfitting.

The absence of any upward trends or significant fluctuations in the latter stages of these curves suggests that the MPTSRNet model is robust, achieving a stable and reliable fit to the data without overfitting. This is further supported by the low final values of the loss functions, indicative of high model accuracy and generalisability across different maritime datasets. Such behaviour is crucial for practical deployment, where the model's ability to generalise and perform consistently under varying conditions is paramount.

5.7. Ablation experiments

This section investigates the effectiveness of the MPTNSR model by comparing different configurations through ablation studies. By systematically excluding specific components—spatial, temporal, or semantic modelling—the impact of each feature on the overall performance is analysed. Various combination strategies are applied while maintaining a consistent network structure to ensure a fair comparison. The evaluations are conducted across all datasets using the following variants:

(1) **MPTNSR**: Incorporates spatial, temporal, and semantic modelling, representing the complete model. This configuration uses CNN for spatial feature extraction, BiLSTM for capturing bidirectional temporal dependencies, and GCN for relational and semantic learning. MPTNSR represents the full potential of the model in handling spatiotemporal and semantic dynamics.

(2) **MPTNSR-w/o-spatial**: Excludes spatial feature modelling, utilising only temporal modelling and semantic learning. By omitting CNN, the model cannot capture localised spatial features or dependencies, which may reduce its ability to represent regional variations in VTF data. Instead, temporal and semantic information is modelled using configurations such as LSTM-GCN, BiLSTM-GCN, GRU-GCN, and BiGRU-GCN. This setup evaluates the contribution of spatial features to prediction accuracy.

(3) **MPTNSR-w/o-temporal**: Omits temporal modelling, relying solely on spatial and semantic modelling. This variant corresponds to CNN-GCN. However, CNN-GCN is designed for handling spatiotemporal features, and the absence of temporal modelling limits GCN to processing only static relationships and CNN to capturing local static features. As a result, CNN-GCN becomes incapable of supporting dynamic prediction tasks. Without temporal modelling, it fails to capture complex, dynamic patterns, such as variations in VTF over time.

(4) **MPTNSR-w/o-semantic**: Excludes semantic modelling, relying on spatial and temporal modelling. The exclusion of GCN prevents the model from learning relational dependencies and interactions between different channels. It uses models like CNN-LSTM, CNN-BiLSTM, CNN-GRU, CNN-BiGRU, and ICLSNet (CNN-LSTM-DTW) to combine spatial and temporal information without incorporating inter-channel relationships. This configuration helps analyse the importance of semantic learning in improving prediction outcomes.

The ablation studies validate the critical role of each component in the overall performance of the MPTNSR model. By integrating

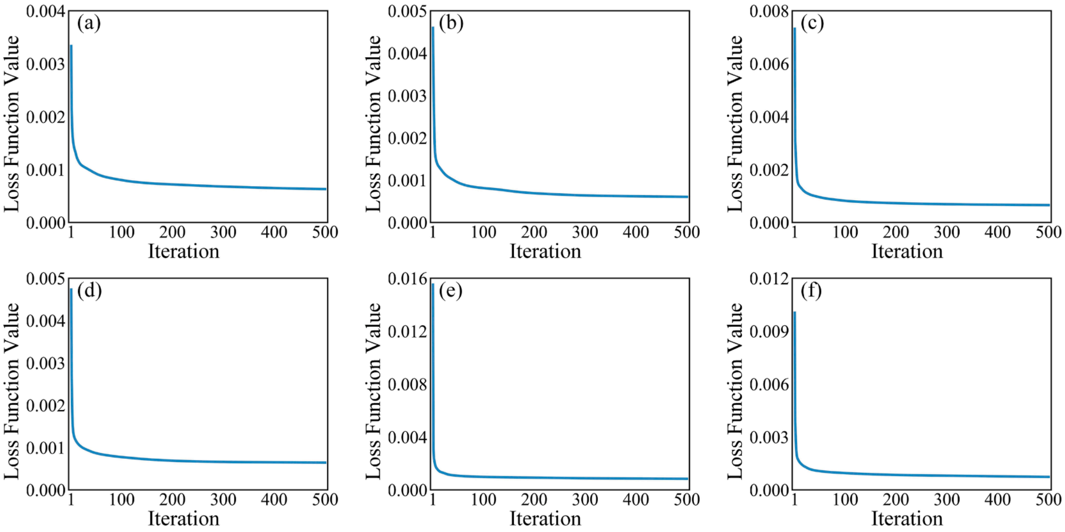


Fig. 16. The results of the loss function curve in the CFDP water area. (a) – (f) indicate datasets C_1 – C_6, respectively.

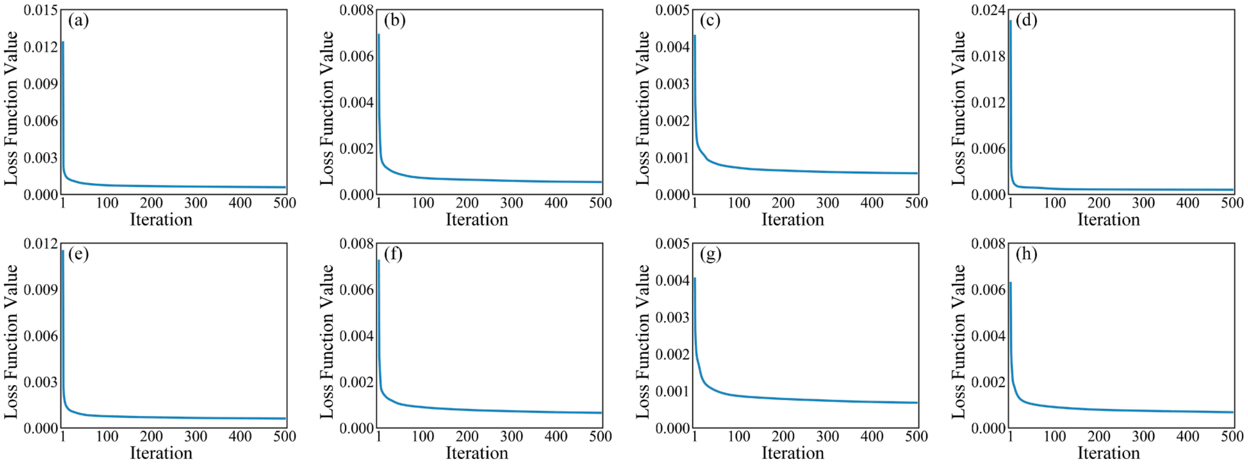


Fig. 17. The results of the loss function curve in the CFDP water area. (a) – (h) indicate datasets CJ_1 – CJ_8, respectively.

spatial, temporal, and semantic features, the complete model effectively captures the intricate dynamics of VTF, leading to superior accuracy and reliability. The results, summarised in Tables 13 and 14, highlight that removing any single component significantly degrades the model's ability to deliver accurate and stable predictions, as demonstrated by higher MAPE and RMSE values and lower R^2 scores.

Furthermore, the distinct contributions of spatial, temporal, and semantic modelling in port operations, open-sea navigation, and complex maritime networks provide a stronger justification for the MPTNSR model's architecture and highlight its practical benefits in maritime navigation and traffic forecasting. The detailed impacts are outlined below.

(1) Impact of spatial features in maritime traffic scenarios.

Spatial features play a crucial role in capturing regional variations, port congestion patterns, and navigational constraints. In structured maritime environments, such as straits, estuaries, and port approaches, vessel movements are heavily influenced by localised spatial dependencies. The removal of spatial modelling (MPTNSR-w/o-spatial) significantly degrades the model's ability to account for traffic bottlenecks, restricted waterways, and varying ship densities across regions.

For instance, in port areas (e.g., CFDP dataset), where vessel interactions are concentrated, and manoeuvring decisions are dictated by tight spatial constraints, the absence of CNN-driven feature extraction leads to higher RMSE and MAPE values. Without spatial awareness, models such as LSTM-GCN and GRU-GCN struggle to differentiate between traffic-congested zones and open-sea navigation, making them ineffective in regional traffic flow prediction. The lack of spatial information also prevents the model from capturing localised vessel movement patterns, reducing its ability to reflect real-time variations in maritime traffic density.

(2) Impact of temporal features in dynamic traffic conditions.

Temporal dependencies are essential in modelling evolving vessel movement patterns, seasonal variations in maritime traffic, and periodic port congestion cycles. The exclusion of temporal modelling (MPTNSR-w/o-temporal) significantly reduces the model's ability to capture historical trends and sequential dependencies, impairing its effectiveness in predicting dynamic vessel behaviour.

This limitation is particularly pronounced in open-sea navigation scenarios (e.g., CJ dataset), where vessel movement follows cyclical patterns influenced by weather conditions, tidal effects, and operational schedules. Without BiLSTM-driven temporal modelling, CNN-GCN-based models fail to identify recurring congestion waves, fluctuating ship traffic densities, and risk-prone navigation windows. The absence of temporal awareness results in discontinuous and short-term predictions, which are less useful for longitudinal vessel traffic forecasting and maritime route optimisation.

(3) Impact of semantic features in complex maritime networks.

Semantic modelling, enabled by GCN, is essential for understanding relational dependencies between vessels, navigational regulations, and multi-source maritime data integration. The exclusion of semantic modelling (MPTNSR-w/o-semantic) leads to substantial performance degradation, particularly in high-density maritime networks where vessel interactions, port regulations, and external factors affect movement patterns collectively.

For example, in major port clusters (e.g., CJ water areas), vessel traffic is dictated by traffic control measures, ship berthing schedules, and multi-vessel interactions. Without GCN-based relational learning, models like CNN-LSTM and CNN-GRU, which rely solely on spatial and temporal learning, struggle to capture interdependent vessel movements such as anchorage waiting patterns, collision avoidance strategies, and formation-based navigation. This shortcoming may result in inaccurate and unreliable predictions, particularly in congested and regulation-intensive waterways.

(4) The complete MPTNSR model: Robustness across maritime scenarios.

By seamlessly integrating spatial, temporal, and semantic learning, the MPTNSR model achieves state-of-the-art predictive accuracy across a wide range of maritime environments. This integration enables context-aware vessel traffic prediction, making it adaptable to various operational scenarios.

In port areas, MPTNSR effectively models localised congestion patterns and vessel manoeuvring behaviours, offering precise short-term traffic forecasts. In open-sea navigation, it captures long-term temporal dependencies and fluctuated traffic densities, ensuring robust voyage predictions. In complex maritime networks, its relational learning capabilities enhance interaction-aware predictions, improving maritime situational awareness.

The MPTNSR model consistently achieves lower MAPE and RMSE values, with all the associated R^2 scores nearing 1.0, demonstrating its exceptional predictive accuracy across multiple datasets. The ablation results confirm that the integration of spatial, temporal, and semantic features enhances robustness, enabling the model to outperform all variants and effectively capture complex maritime traffic dynamics. This validates MPTNSR as a reliable tool for predictive tasks in maritime navigation and traffic forecasting.

5.8. Time complexity analysis

The computation time of the proposed MPTNSR model on a Central Processing Unit (CPU) for the two water areas (CFDP and CJ) is detailed in Table 15. While the MPTNSR model achieves the best prediction performance, its running time reflects the complexity of its computations. For the CFDP channels, the running time ranges from approximately 2269.98 (C_3) to 2723.08 s (C_1), with relatively consistent time across channels, suggesting a balanced computational complexity. In contrast, the CJ channels exhibit a slightly broader range, from 2313.30 (CJ_6) to 2782.18 s (CJ_5), indicating that certain channels (e.g., CJ_5 and CJ_3) require more computational resources due to their associated larger datasets or more complex inter-channel relationships.

The CFDP channels generally show slightly higher average computation time compared to most CJ channels, except for outliers like CJ_5, which exhibit significantly higher demands. This difference might reflect the increased complexity or denser data structures in CFDP datasets. Despite these variations, the model demonstrates consistent computational balance across most channels, with only a few exceptions, showing higher variance.

The MPTNSR model's significant computation time reflects its complexity, integrating periodic, temporal, and semantic features to achieve exceptional predictive accuracy. While the model performs well, enhancing its computational efficiency is essential for practical applications, particularly in real-time and large-scale scenarios. Transitioning to GPU-based computation addresses these high computational demands, enabling parallel processing to significantly reduce running times and improve the model's scalability.

To further optimise the proposed MPTNSR model's performance, experiments were conducted using multiple GPUs. The running times for the CFDP and CJ water areas, as shown in Table 14, highlight the impact of using one, two, and four GPUs. The key findings are summarised below:

(1) Single GPU (30 % speedup)

The single GPU setup significantly reduces the computational time compared to no GPU acceleration, achieving approximately 30 % time reduction. This demonstrates the efficiency of parallelised computation on a GPU for tasks like this.

(2) Two GPUs (42 % speedup)

Adding a second GPU further reduces the computation time but only provides a moderate additional time reduction beyond the single GPU. The speedup ratio from one to two GPUs is approximately 12 %, suggesting diminishing returns due to factors like communication overhead and workload synchronisation between GPUs.

(3) Four GPUs (70 % speedup)

Table 15
The running time of the proposed MPTNSR model on a CPU in two water areas (measured in seconds).

Channel	CFDP (CPU)	CFDP (One GPU)	CFDP (Two GPUs)	CFDP (Four GPUs)	Channel	CJ (CPU)	CJ (One GPU)	CJ (Two GPUs)	CJ (Four GPUs)
C_1	2723.08	1927.82	1531.19	807.47	CJ_1	2699.27	1898.36	1519.45	791.25
C_2	2318.98	1656.65	1326.64	673.59	CJ_2	2331.14	1667.81	1307.73	678.02
C_3	2269.98	1587.86	1305.55	646.63	CJ_3	2595.19	1811.19	1543.56	771.24
C_4	2319.31	1586.79	1303.16	688.51	CJ_4	2352.41	1687.25	1335.51	616.03
C_5	2307.11	1645.99	1309.39	671.56	CJ_5	2782.18	1902.86	1658.04	773.86
C_6	2311.86	1607.70	1310.29	636.70	CJ_6	2313.30	1633.81	1363.51	625.58
–	–				CJ_7	2314.22	1610.41	1379.18	708.87
–	–				CJ_8	2594.90	1854.36	1542.74	665.63

The use of four GPUs shows the biggest running time reduction, with a 70 % reduction rate compared to no GPU usage. However, the incremental gain from two to four GPUs (28 %) is still less than the initial improvement achieved with the first GPU. This indicates that while additional GPUs enhance performance, the efficiency gain diminishes as the number of GPUs increases.

These findings suggest that four GPUs provide the optimal balance between performance and efficiency for the MPTNSR model. Adding more GPUs might yield further reductions in computation time, but the incremental benefits would likely be marginal compared to the increased cost and complexity. These results underscore the importance of optimising hardware resources to enhance the MPTNSR model's efficiency and scalability, making it more practical for real-world applications.

Although the training process for the GPU-based MPTNSR model takes between 600 and 800 s due to its architecture and comprehensive feature integration, the testing phase and future real-world applications could be significantly faster. This distinction highlights that while the model requires substantial computational time during the initial training phase to learn complex patterns and relationships, its efficiency during inference allows for rapid processing of the test dataset. The model can deliver near-instantaneous prediction in practical applications, making it highly competitive for real-time scenarios where quick decision-making is critical. The trade-off between long training time and fast testing ensures both robustness and practicality in deployment.

5.9. *Prediction performance on low-quality data*

To further assess the effectiveness of the proposed MPTNSR model in addressing low-quality data scenarios, comparative experiments were conducted using both normal and noisy datasets. The noisy dataset was generated by introducing noise to the normal VTF dataset, where the noise follows a normal distribution with a mean of 1 and a standard deviation of 5. These experiments evaluate the robustness and performance of the MPTNSR model under varying data quality conditions.

Figs. 18 and 19 depict the VTF data distribution across two water areas under normal and noisy conditions, respectively. From these figures, it is evident that adding noise increases the variability and dispersion of the VTF data, as shown by the presence of outliers and a wider range in the box plots.

The comparative performance of the MPTNSR model on normal and noisy VTF datasets is evaluated using the REMean index, as shown in Figs. 19 and 20 for the CFDP and CJ water areas. The radar plots provide a clear visualisation of the model's performance under both conditions. Despite the increased errors introduced by noisy data, the MPTNSR model demonstrates strong adaptability, with errors remaining within a reasonable range for most channels and time intervals. These results highlight the robustness and practicality of the MPTNSR model in handling noisy data, making it a reliable choice for real-world applications in complex and dynamic maritime environments.

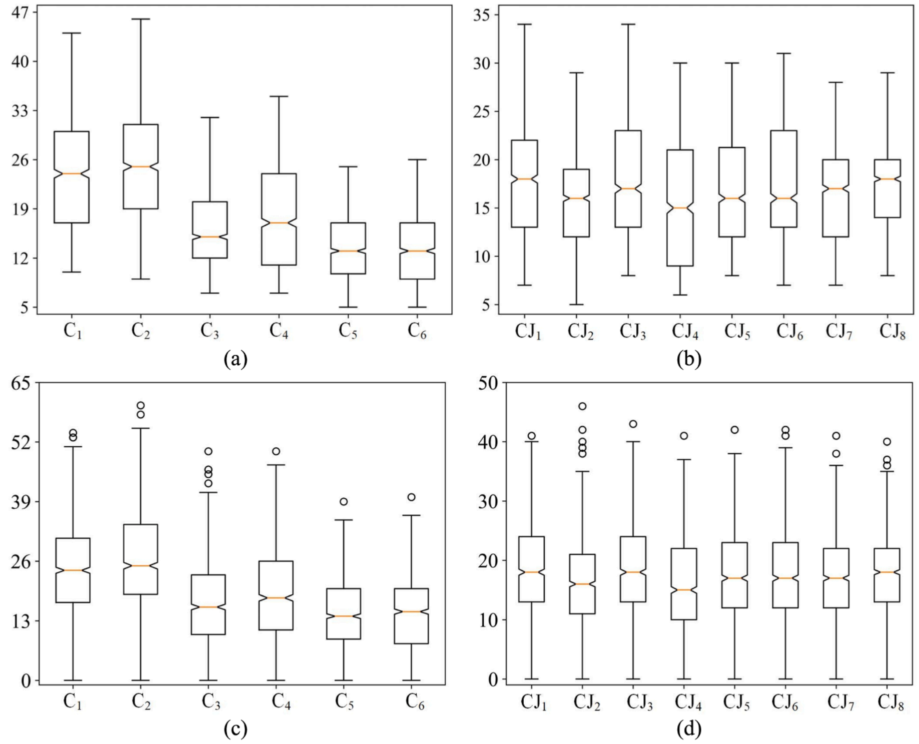


Fig. 18. Distribution of normal and noisy VTF data in two water areas: (a) and (b) represent the normal VTF data distribution across six channels in the CFDP area and eight channels in the CJ area, respectively; (c) and (d) represent the noisy VTF data distribution across six channels in the CFDP area and eight channels in the CJ area, respectively.

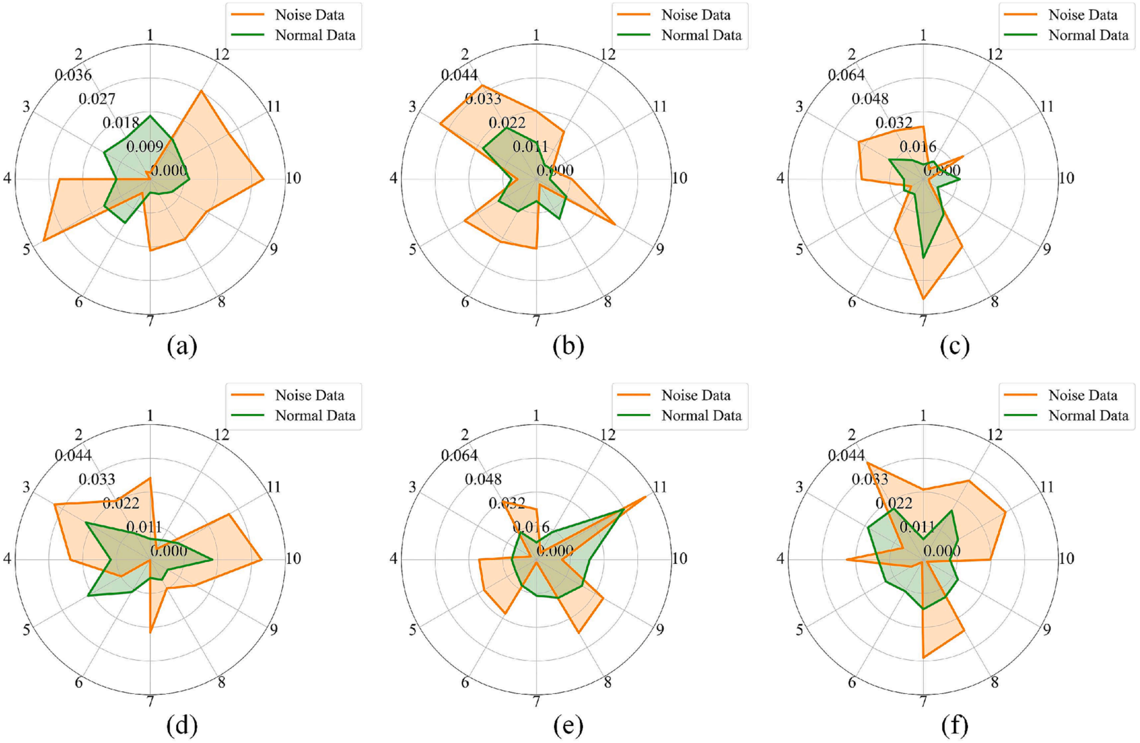


Fig. 19. The comparative results of the proposed MPTNSR model on normal and noisy VTF data based on the REMean index across six channels in the CFDP water area.

5.10. Discussion and implications

The proposed MPTNSR method for VTF prediction demonstrates significant advantages in terms of accuracy, robustness, and practical applicability, offering valuable implications for various stakeholders, including port companies, maritime regulatory authorities, and research institutions. The integration of CNN-BiLSTM and GCN ensures a holistic approach, capturing both local and global metrics to address the dynamic and interconnected nature of maritime traffic.

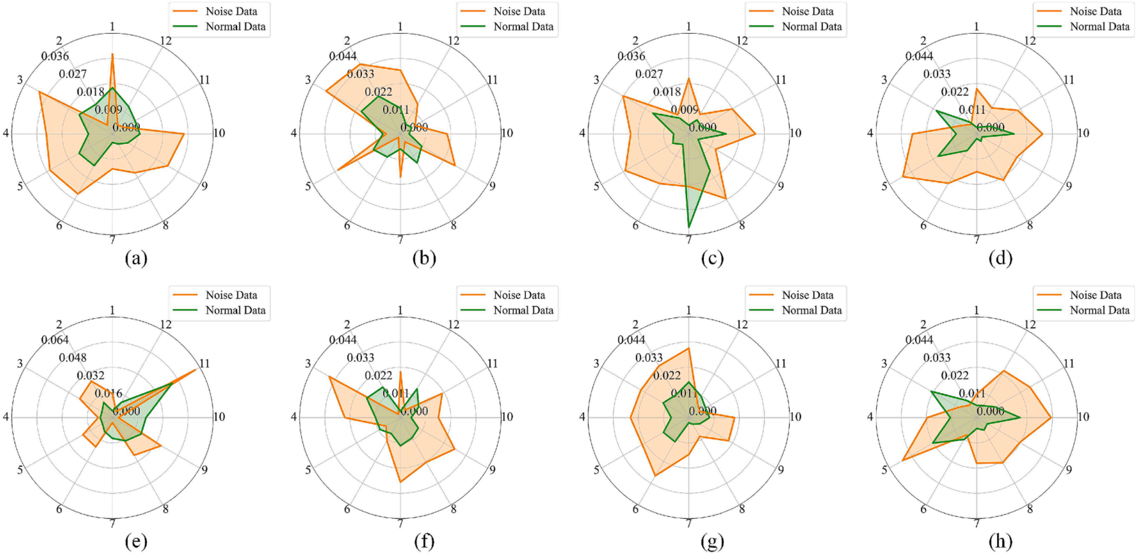


Fig. 20. The comparative results of the proposed MPTNSR model on normal and noisy VTF data based on the REMean index across eight channels in the CJ water area.

The MPTNSR method excels in predictive accuracy, as demonstrated by its ability to handle complex spatiotemporal and semantic relationships in VTF data. Its superior performance across different datasets underscores its robustness and adaptability, making it a reliable tool for predicting vessel movements under diverse operational conditions. The model's capability to incorporate external factors such as weather conditions, traffic peaks, and accidents ensures that predictions remain relevant and actionable, even in unpredictable environments.

By combining CNN for spatial feature extraction, BiLSTM for bidirectional temporal modelling, and GCN for semantic learning, the MPTNSR model achieves seamless integration of key perspectives. This enables it to deliver precise and stable predictions, supporting real-time decision-making and long-term planning. The ability to balance periodic, temporal, and semantic views allows the model to adapt to both short-term fluctuations and long-term trends in VTF data.

The MPTNSR model's enhanced accuracy supports efficient traffic management decisions, reducing congestion in busy waterways and mitigating collision risks. Its ability to predict VTF trends enables proactive traffic control measures, improving overall navigational safety and operational efficiency. For stakeholders, these predictive capabilities translate into tangible benefits, such as optimised scheduling, resource allocation, and infrastructure planning.

The adoption of the MPTNSR method allows port operators to optimise docking schedules, minimise vessel waiting times, and allocate resources such as pilot boats and berthing spaces more effectively. This leads to reduced operational costs, improved service quality, and better capacity planning. Accurate VTF predictions also help anticipate peak traffic periods, ensuring ports are adequately prepared to manage fluctuating volumes.

For regulatory bodies, the MPTNSR method provides a robust framework for monitoring and managing vessel movements. Enhanced predictive accuracy boosts navigational safety by enabling authorities to anticipate and address traffic bottlenecks, adverse weather conditions, and potential accidents proactively. This improves compliance with maritime regulations and enhances overall maritime safety.

The MPTNSR method offers researchers a powerful tool for studying VTF patterns and dynamics. Its ability to handle complex datasets across various scenarios provides a foundation for developing advanced predictive models and exploring the interplay between vessel types, traffic flows, and environmental factors. This contributes to the broader field of intelligent maritime traffic management.

The ability to analyse VTF trends and patterns supports long-term strategic planning for infrastructure investments and policy decisions. By leveraging predictive insights, authorities can optimise port expansions, improve traffic regulations, and ensure sustainable growth in the maritime sector.

The MPTNSR method offers a transformative approach to VTF prediction, providing substantial benefits across operational, safety, and strategic domains. Its advanced modelling capabilities ensure that stakeholders can make informed, data-driven decisions, enhancing the efficiency, safety, and sustainability of maritime operations. The adoption of this method marks a significant step forward in intelligent vessel traffic management, aligning with the goals of modern maritime transportation systems.

6. Conclusions

This paper introduces a new prediction technique, termed MPTNSR, aimed at enhancing VTF prediction capabilities. The MPTNSR model is distinguished by its three-pronged approach: a periodic perspective (harnessed via CNN), a temporal dimension (leveraged through Bi-LSTM), and a semantic angle (mediated by GCN). To further refine the evaluation of prediction efficacy, an innovative loss function is introduced, integrating local measurements (via MSE) with global insights (through PCC). The proposed MPTNSR method introduces three key advantages that enhance the precision and consistency of VTF data prediction. First, it integrates temporal and periodic patterns using a CNN-BiLSTM network, effectively capturing complex temporal dependencies. Second, it employs semantic collaboration through a GCN, enabling the model to map relationships based on VTF variations and geographical placements. Third, a refined loss function incorporating PCC and MSE is introduced, ensuring a balanced evaluation by combining local and global metrics. These advancements significantly improve the accuracy and robustness of VTF data prediction, making the MPTNSR method a highly effective tool for maritime applications.

The MPTNSR method advances collaborative VTF prediction across multiple channels, thanks to its semantic view. This paper compares the proposed MPTNSR method with eighteen other methods using VTF data from six channels in the CFDP water area and eight channels in the CJ water area. The experiments demonstrate that the MPTNSR method achieves better predictive performance, especially in terms of accuracy and stability, for collaborative VTF prediction across multiple channels. The GPU-accelerated MPTNSR method has significantly reduced training time and enhanced scalability.

As a result, the proposed MPTNSR method can significantly support port companies, maritime regulatory authorities, and research institutions in carrying out VTF prediction tasks effectively. To facilitate the practical implementation of this method, future research can focus on two key areas. Firstly, considering the rapid evolution of VTF due to external factors like adverse weather, vessel accidents, or the impact of shock events (e.g., COVID-19), it is important to incorporate these unforeseen circumstances into network training. Secondly, since VTF patterns vary against different vessel types, analysing these variations and integrating them into training networks is insightful. By addressing these aspects, future research can be conducted to improve the adaptability of the proposed MPTNSR method based on real-time VTF data.

CRediT authorship contribution statement

Huanhuan Li: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yu Zhang:** Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis. **Yan Li:** Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jasmine Siu Lee Lam:** Writing – review & editing, Validation, Methodology, Investigation, Formal analysis. **Christian Matthews:** Writing – review & editing, Validation, Methodology, Supervision. **Zaili Yang:** Writing – review & editing, Validation, Resources, Project administration, Methodology, Investigation, Funding acquisition.

Declaration of competing interest

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

Acknowledgement

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 864724) and Royal Society International Exchanges 2021 Cost Share (NSFC) (IEC\NSFC\211211).

Appendix

Table A1
The prediction results (RMSE, MAPE, and R^2) of the proposed MPTNSR method using three different sizes of the local matrix in six datasets in the CFDP water area.

Dataset	Performance indexes	Size of local matrix		
		3 × 3	5 × 5	7 × 7
C_1	MAPE↓	**0.0058**	0.0181	0.0414
	RMSE↓	**0.1847**	0.7287	1.3838
	R^2↑	**0.9993**	0.9895	0.9624
C_2	MAPE↓	**0.0084**	0.0191	0.0921
	RMSE↓	**0.3076**	0.4460	2.3723
	R^2↑	**0.9986**	0.9971	0.9206
C_3	MAPE↓	**0.0120**	0.0173	0.0532
	RMSE↓	**0.4016**	0.6916	1.4829
	R^2↑	**0.9961**	0.9886	0.9476
C_4	MAPE↓	**0.0095**	0.0274	0.0468
	RMSE↓	**0.2753**	0.6184	1.2687
	R^2↑	**0.9989**	0.9948	0.9783
C_5	MAPE↓	**0.0109**	0.0663	0.1017
	RMSE↓	**0.1636**	0.7644	1.2748
	R^2↑	**0.9973**	0.9420	0.8387
C_6	MAPE↓	**0.0072**	0.0346	0.0528
	RMSE↓	**0.1408**	0.5820	1.0084
	R^2↑	**0.9992**	0.9877	0.9633

Table A2
The prediction results (RMSE, MAPE, and R^2) of the proposed MPTNSR method using three different sizes of the local matrix across eight datasets in the CJ water area.

Dataset	Performance indexes	Size of local matrix		
		3 × 3	5 × 5	7 × 7
CJ_1	MAPE↓	**0.0149**	0.0380	0.0462
	RMSE↓	**0.2947**	1.1180	1.5657
	R^2↑	**0.9975**	0.9651	0.9317
CJ_2	MAPE↓	**0.0168**	0.0409	0.0556
	RMSE↓	**0.2542**	0.6879	0.9334
	R^2↑	**0.9964**	0.9736	0.9515
CJ_3	MAPE↓	**0.0102**	0.0145	0.0268
	RMSE↓	**0.3165**	0.4792	0.7718
	R^2↑	**0.9976**	0.9946	0.9860
CJ_4	MAPE↓	**0.0103**	0.0148	0.0215
	RMSE↓	**0.1956**	0.2730	0.4628
	R^2↑	**0.9987**	0.9975	0.9929
CJ_5	MAPE↓	**0.0162**	0.0198	0.0230

(*continued on next page*)

Table A2 (*continued*)

Dataset	Performance indexes	Size of local matrix		
		3 × 3	5 × 5	7 × 7
	RMSE↓	**0.3856**	0.4413	0.5084
	R^2↑	**0.9959**	0.9946	0.9929
CJ_6	MAPE↓	**0.0118**	0.0151	0.0228
	RMSE↓	**0.2540**	0.3379	0.5275
	R^2↑	**0.9980**	0.9965	0.9915
CJ_7	MAPE↓	**0.0120**	0.0198	0.0235
	RMSE↓	**0.2385**	0.3877	0.4809
	R^2↑	**0.9975**	0.9935	0.9901
CJ_8	MAPE↓	**0.0145**	0.0185	0.0287
	RMSE↓	**0.3428**	0.4782	0.7557
	R^2↑	**0.9931**	0.9865	0.9665

Table A3

The prediction results (RMSE, MAPE, and R^2) of the proposed MPTNSR method using five different learning rates in the CFDP water area.

Dataset	Performance indexes	Learning rates				
		0.0001	0.0005	**0.001**	0.005	0.01
C_1	MAPE↓	0.0081	0.0075	**0.0058**	0.0079	0.0162
	RMSE↓	0.2582	0.2185	**0.1847**	0.2546	0.4964
	R^2↑	0.9986	0.9990	**0.9993**	0.9987	0.9951
C_2	MAPE↓	0.0090	0.0056	**0.0084**	0.0243	0.0461
	RMSE↓	0.3916	0.1642	**0.3076**	0.7138	1.0496
	R^2↑	0.9978	0.9996	**0.9986**	0.9928	0.9844
C_3	MAPE↓	0.0144	0.0136	**0.0120**	0.0351	0.0524
	RMSE↓	0.4746	0.4356	**0.4016**	1.0847	1.1685
	R^2↑	0.9946	0.9954	**0.9961**	0.9719	0.9675
C_4	MAPE↓	0.0147	0.0116	**0.0095**	0.0245	0.0469
	RMSE↓	0.4597	0.4031	**0.2753**	0.7260	1.0095
	R^2↑	0.9971	0.9978	**0.9989**	0.9928	0.9862
C_5	MAPE↓	0.0188	0.0113	**0.0109**	0.0134	0.0190
	RMSE↓	0.2133	0.1759	**0.1636**	0.2054	0.2187
	R^2↑	0.9954	0.9969	**0.9973**	0.9958	0.9952
C_6	MAPE↓	0.0119	0.0043	**0.0072**	0.0115	0.0300
	RMSE↓	0.2511	0.0825	**0.1408**	0.1809	0.5505
	R^2↑	0.9977	0.9997	**0.9992**	0.9988	0.9890

Table A4

The prediction results (RMSE, MAPE, and R^2) of the proposed MPTNSR method using five different learning rates in the CJ water area.

Dataset	Performance indexes	Learning rates				
		0.0001	0.0005	**0.001**	0.005	0.01
CJ_1	MAPE↓	0.0313	0.0175	**0.0149**	0.0217	0.0203
	RMSE↓	0.6190	0.3887	**0.2947**	0.4276	0.4260
	R^2↑	0.9893	0.9957	**0.9975**	0.9949	0.9949
CJ_2	MAPE↓	0.0258	0.0203	**0.0168**	0.0284	0.0331
	RMSE↓	0.4001	0.3094	**0.2542**	0.4471	0.5035
	R^2↑	0.9910	0.9946	**0.9964**	0.9888	0.9858
CJ_3	MAPE↓	0.0131	0.0096	**0.0102**	0.0104	0.0140
	RMSE↓	0.4912	0.2581	**0.3165**	0.2749	0.3920
	R^2↑	0.9943	0.9984	**0.9976**	0.9982	0.9963
CJ_4	MAPE↓	0.0225	0.0170	**0.0103**	0.0175	0.0240
	RMSE↓	0.4089	0.3614	**0.1956**	0.3301	0.4880
	R^2↑	0.9945	0.9957	**0.9987**	0.9964	0.9921
CJ_5	MAPE↓	0.0208	0.0190	**0.0162**	0.0234	0.0208
	RMSE↓	0.5983	0.4385	**0.3856**	0.5770	0.5729
	R^2↑	0.9901	0.9947	**0.9959**	0.9908	0.9909
CJ_6	MAPE↓	0.0123	0.0105	**0.0118**	0.0203	0.0120
	RMSE↓	0.2656	0.2452	**0.2540**	0.4113	0.3048
	R^2↑	0.9978	0.9981	**0.9980**	0.9948	0.9971
CJ_7	MAPE↓	0.0262	0.0187	**0.0120**	0.0177	0.0197
	RMSE↓	0.5700	0.4047	**0.2385**	0.3941	0.4062
	R^2↑	0.9861	0.9929	**0.9975**	0.9933	0.9929
CJ_8	MAPE↓	0.0273	0.0251	**0.0145**	0.0249	0.0286
	RMSE↓	0.7241	0.6769	**0.3428**	0.7058	0.8195
	R^2↑	0.9692	0.9731	**0.9931**	0.9707	0.9606

Table A5
The prediction results of the proposed MPTNSR model for five different sequence lengths in the CFDP water area.

Dataset	Performance indexes	Sequence lengths				
		3	6	9	12	15
C_1	MAPE↓	**0.0056**	0.0058	0.0130	0.0205	0.0179
	RMSE↓	**0.1604**	0.1847	0.4307	0.6274	0.4714
	R^2↑	**0.9994**	0.9993	0.9963	0.9922	0.9956
C_2	MAPE↓	**0.0062**	0.0084	0.0206	0.0193	0.0218
	RMSE↓	**0.1712**	0.3076	0.5956	0.5475	0.6520
	R^2↑	**0.9995**	0.9986	0.9949	0.9957	0.9940
C_3	MAPE↓	**0.0089**	0.0120	0.0285	0.0247	0.0215
	RMSE↓	**0.3188**	0.4016	0.8465	0.7758	0.7313
	R^2↑	**0.9975**	0.9961	0.9829	0.9856	0.9872
C_4	MAPE↓	**0.0082**	0.0095	0.0208	0.0286	0.0194
	RMSE↓	**0.2639**	0.2753	0.5959	0.9392	0.5466
	R^2↑	**0.9990**	0.9989	0.9952	0.9881	0.9959
C_5	MAPE↓	**0.0108**	0.0109	0.0163	0.0295	0.0260
	RMSE↓	**0.1568**	0.1636	0.2207	0.4104	0.3264
	R^2↑	**0.9975**	0.9973	0.9951	0.9832	0.9894
C_6	MAPE↓	**0.0041**	0.0072	0.0183	0.0112	0.0239
	RMSE↓	**0.0805**	0.1408	0.3545	0.2376	0.4863
	R^2↑	**0.9997**	0.9992	0.9954	0.9979	0.9914

Table A6
The prediction results of the proposed MPTNSR model for five different sequence lengths in the CJ water area.

Dataset	Performance indexes	Sequence lengths				
		3	6	9	12	15
CJ_1	MAPE↓	**0.0101**	0.0149	0.0171	0.0268	0.0231
	RMSE↓	**0.1841**	0.2947	0.3540	0.5219	0.4515
	R^2↑	**0.9990**	0.9975	0.9965	0.9924	0.9943
CJ_2	MAPE↓	**0.0107**	0.0168	0.0253	0.0240	0.0240
	RMSE↓	**0.1563**	0.2542	0.3439	0.3369	0.3848
	R^2↑	**0.9986**	0.9964	0.9934	0.9936	0.9917
CJ_3	MAPE↓	**0.0098**	0.0102	0.0227	0.0207	0.0196
	RMSE↓	**0.3119**	0.3165	0.6909	0.6347	0.6158
	R^2↑	**0.9977**	0.9976	0.9887	0.9905	0.9910
CJ_4	MAPE↓	**0.0084**	0.0103	0.0153	0.0209	0.0162
	RMSE↓	**0.1770**	0.1956	0.3165	0.4212	0.2885
	R^2↑	**0.9989**	0.9987	0.9967	0.9941	0.9972
CJ_5	MAPE↓	**0.0146**	0.0162	0.0193	0.0257	0.0258
	RMSE↓	**0.3895**	0.3856	0.4686	0.6483	0.6123
	R^2↑	**0.9958**	0.9959	0.9939	0.9884	0.9897
CJ_6	MAPE↓	**0.0103**	0.0118	0.0185	0.0158	0.0203
	RMSE↓	**0.2207**	0.2540	0.4111	0.3612	0.5006
	R^2↑	**0.9985**	0.9980	0.9948	0.9960	0.9923
CJ_7	MAPE↓	**0.0092**	0.0120	0.0247	0.0241	0.0221
	RMSE↓	**0.1629**	0.2385	0.4874	0.4442	0.4396
	R^2↑	**0.9988**	0.9975	0.9898	0.9915	0.9917
CJ_8	MAPE↓	**0.0110**	0.0145	0.0218	0.0273	0.0269
	RMSE↓	**0.2402**	0.3428	0.5218	0.6736	0.6917
	R^2↑	**0.9966**	0.9931	0.9840	0.9733	0.9719

Data availability

Data will be made available on request.

References

Allen, D.M., 1971. Mean Square Error of Prediction as a Criterion for Selecting Variables. Technometrics 13, 469–475. https://doi.org/10.1080/00401706.1971.10488811.

Bao, Y., Huang, J., Shen, Q., Cao, Y., Ding, W., Shi, Z., Shi, Q., 2023. Spatial–Temporal Complex Graph Convolution Network for Traffic Flow Prediction. Eng. Appl. Artif. Intel. 121, 106044. https://doi.org/10.1016/j.engappai.2023.106044.

Benesty, J., Chen, J., Huang, Y., 2008. On the Importance of the Pearson Correlation Coefficient in Noise Reduction. IEEE Trans. Audio Speech Lang. Process. 16, 757–765. https://doi.org/10.1109/TASL.2008.919072.

Bharti, R., Kumar, K., 2023. Short-term traffic flow prediction based on optimized deep learning neural network: pso-bi-lstm. Phys. A: Statist. Mechan. Appl. 625, 129001. https://doi.org/10.1016/j.physa:2023.129001.

Cao, S., Wu, L., Wu, J., Wu, D., Li, Q., 2022. A spatio-temporal sequence-to-sequence network for traffic flow prediction. Inf. Sci. 610, 185–203. https://doi.org/10.1016/j.ins.2022.07.125.

Cao, S., Wu, L., Zhang, R., Wu, D., Cui, J., Chang, Y., 2024. A Spatiotemporal Multiscale Graph Convolutional Network for Traffic Flow Prediction. IEEE Trans. Intell. Transp. Syst. 25, 8705–8718. https://doi.org/10.1109/TITS.2024.3354802.

Chai, T., Draxler, R.R., 2014. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. Geosci. Model Dev. 7, 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014.

Chen, X., Chen, X.M., 2022. A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction. Transp. Res. Part C Emerging Technol. 143, 103820. https://doi.org/10.1016/j.trc.2022.103820.

Cheng, Z., Lu, J., Zhou, H., Zhang, Y., Zhang, L., 2022. Short-Term Traffic Flow Prediction: An Integrated Method of Econometrics and Hybrid Deep Learning. IEEE Trans. Intell. Transp. Syst. 23, 5231–5244. https://doi.org/10.1109/TITS.2021.3052796.

Dey, R., Salem, F.M., 2017. Gate-variants of Gated Recurrent Unit (GRU) neural networks, in: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS). Presented at the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1597–1600. Doi: 10.1109/MWSCAS.2017.8053243.

Do, L.N.N., Vu, H.L., Vo, B.Q., Liu, Z., Phung, D., 2019. An effective spatial-temporal attention based neural network for traffic flow prediction. Transp. Res. Part C Emerging Technol. 108, 12–28. https://doi.org/10.1016/j.trc.2019.09.008.

Doucoure, B., Agbossou, K., Cardenas, A., 2016. Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data. Renew. Energy 92, 202–211. https://doi.org/10.1016/j.renene.2016.02.003.

Forti, N., Millefiori, L.M., Braca, P., Willett, P., 2020. Prediction oof Vessel Trajectories From AIS Data Via Sequence-To-Sequence Recurrent Neural Networks, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Presented at the ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8936–8940. Doi: 10.1109/ICASSP40776.2020.9054421.

Fu, S., Zhong, S., Lin, L., Zhao, M., 2022. A Novel Time-Series Memory Auto-Encoder With Sequentially Updated Reconstructions for Remaining Useful Life Prediction. IEEE Trans. Neural Networks Learn. Syst. 33, 7114–7125. https://doi.org/10.1109/TNNLS.2021.3084249.

Gao, M., Shi, G., Li, S., 2018. Online Prediction of Ship Behavior with Automatic Identification System Sensor Data Using Bidirectional Long Short-Term Memory Recurrent Neural Network. Sensors 18, 4211. https://doi.org/10.3390/s18124211.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2020. Generative adversarial networks. Commun. ACM 63, 139–144. https://doi.org/10.1145/3422622.

Graves, A., 2012. Long Short-Term Memory. In: Graves, A. (Ed.), Supervised Sequence Labelling with Recurrent Neural Networks, Studies in Computational Intelligence. Springer, Berlin, Heidelberg, pp. 37–45. https://doi.org/10.1007/978-3-642-24797-2_4.

Grifoll, M., 2019. A statistical forecasting model applied to container throughput in a multi-port gateway system: the Barcelona-Tarragona-Valencia case. Int. J. Shipp. Transport Logist. 11, 316–333. https://doi.org/10.1504/IJSTL.2019.100453.

Han, Z., Zhao, J., Leung, H., Ma, K.F., Wang, W., 2021. A review of deep learning models for time series prediction. IEEE Sens. J. 21, 7833–7848. https://doi.org/10.1109/JSEN.2019.2923982.

Hao, W., Sun, X., Wang, C., Chen, H., Huang, L., 2022. A hybrid EMD-LSTM model for non-stationary wave prediction in offshore China. Ocean Eng. 246, 110566. https://doi.org/10.1016/j.oceaneng.2022.110566.

Hara, K., Saito, D., Shouno, H., 2015. Analysis of function of rectified linear unit used in deep learning, in: 2015 International Joint Conference on Neural Networks (IJCNN). Presented at the 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. Doi: 10.1109/IJCNN.2015.7280578.

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 Comput 22, 7907–7916. https://doi.org/10.1007/s10586-017-1491-2.

Hou, F., Zhang, Y., Fu, X., Jiao, L., Zheng, W., 2021. The Prediction of Multistep Traffic Flow Based on AST-GCN-LSTM. J. Adv. Transp. 2021, e9513170. https://doi.org/10.1155/2021/9513170.

Hou, J., Luo, C., Qin, F., Shao, Y., Chen, X., 2023. FuS-GCN: Efficient B-rep based graph convolutional networks for 3D-CAD model classification and retrieval. Adv. Eng. Inf. 56, 102008. https://doi.org/10.1016/j.aei.2023.102008.

Hu, H.-X., Hu, Q., Tan, G., Zhang, Y., Lin, Z.-Z., 2024. A Multi-Layer Model Based on Transformer and Deep Learning for Traffic Flow Prediction. IEEE Trans. Intell. Transp. Syst. 25, 443–451. https://doi.org/10.1109/TITS.2023.3311397.

Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., Zhang, H., 2019. Deep learning with long short-term memory for time series prediction. IEEE Commun. Mag. 57, 114–119. https://doi.org/10.1109/MCOM.2019.1800155.

Huang, W., Song, G., Hong, H., Xie, K., 2014. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. 15, 2191–2201. https://doi.org/10.1109/TITS.2014.2311123.

Huang, Y., Zhang, S., Wen, J., Chen, X., 2020. Short-Term Traffic Flow Prediction Based on Graph Convolutional Network Embedded LSTM 159–168. Doi: 10.1061/9780784483152.014.

Jia, C., Ma, J., 2023. Conditional temporal GAN for intent-aware vessel trajectory prediction in the precautionary area. Eng. Appl. Artif. Intel. 126, 106776. https://doi.org/10.1016/j.engappai.2023.106776.

Jin, X.-B., Gong, W.-T., Kong, J.-L., Bai, Y.-T., Su, T.-L., 2022. PFVAE: A Planar Flow-Based Variational Auto-Encoder Prediction Model for Time Series Data. Mathematics 10, 610. https://doi.org/10.3390/math10040610.

Kayacan, E., Ulutas, B., Kaynak, O., 2010. Grey system theory-based models in time series prediction. Expert Syst. Appl. 37, 1784–1789. https://doi.org/10.1016/j.eswa.2009.07.064.

Kong, F., Li, J., Jiang, B., Song, H., 2019. Short-term traffic flow prediction in smart multimedia system for Internet of Vehicles based on deep belief network. Futur. Gener. Comput. Syst. 93, 460–472. https://doi.org/10.1016/j.future.2018.10.052.

Kumar, P., Gupta, G.P., Tripathi, R., Garg, S., Hassan, M.M., 2023. DLTIF: Deep Learning-Driven Cyber Threat Intelligence Modeling and Identification Framework in IoT-Enabled Maritime Transportation Systems. IEEE Trans. Intell. Transp. Syst. 24, 2472–2481. https://doi.org/10.1109/TITS.2021.3122368.

Kumar, R., Kumar, P., Kumar, Y., 2022. Multi-step time series analysis and forecasting strategy using ARIMA and evolutionary algorithms. Int. J. Inf. Tecnol. 14, 359–373. https://doi.org/10.1007/s41870-021-00741-8.

Lee, E., Khan, J., Son, W.-J., Kim, K., 2023. An efficient feature augmentation and LSTM-based method to predict maritime traffic conditions. Appl. Sci. 13, 2556. https://doi.org/10.3390/app13042556.

Li, H., Jiao, H., Yang, Z., 2023a. AIS data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods. Transport. Res. Part e: Logist. Transportat. Rev. 175, 103152. https://doi.org/10.1016/j.tre.2023.103152.

Li, H., Xing, W., Jiao, H., Yang, Z., Li, Y., 2024a. Deep bi-directional information-empowered ship trajectory prediction for maritime autonomous surface ships. Transport. Res. Part e: Logist. Transport. Rev. 181, 103367. https://doi.org/10.1016/j.tre.2023.103367.

Li, H., Xing, W., Jiao, H., Yuen, K.F., Gao, R., Li, Y., Matthews, C., Yang, Z., 2024b. Bi-directional information fusion-driven deep network for ship trajectory prediction in intelligent transportation systems. Transport. Res. Part e: Logist. Transportat. Rev. 192, 103770. https://doi.org/10.1016/j.tre.2024.103770.

Li, H., Yang, Z., 2023. Incorporation of AIS data-based machine learning into unsupervised route planning for maritime autonomous surface ships. Transp. Res. Part E Logist. Transp. Rev. 176, 103171. https://doi.org/10.1016/j.tre.2023.103171.

Li, J., Yang, J., Gao, L., Wei, L., Mao, F., 2022a. In: Dynamic Spatial-Temporal Graph Convolutional GRU Network for Traffic Forecasting, in. Association for Computing Machinery, New York, NY, USA, pp. 19–24. https://doi.org/10.1145/3510362.3510366.

Li, L., Sheng, X., Du, B., Wang, Y., Ran, B., 2020. A deep fusion model based on restricted Boltzmann machines for traffic accident duration prediction. Eng. Appl. Artif. Intel. 93, 103686. https://doi.org/10.1016/j.engappai.2020.103686.

Li, R., Qin, Y., Wang, J., Wang, H., 2023b. AMGB: Trajectory prediction using attention-based mechanism GCN-BiLSTM in IOV. Pattern Recogn. Lett. 169, 17–27. https://doi.org/10.1016/j.patrec.2023.03.006.

Li, W., Logenthiran, T., Woo, W.L., 2019. Multi-GRU prediction system for electricity generation's planning and operation. IET Gener. Transm. Distrib. 13, 1630–1637. https://doi.org/10.1049/iet-gtd.2018.6081.

Li, W.-F., Mei, B., Shi, G.-Y., 2018. Automatic recognition of marine traffic flow regions based on Kernel Density Estimation. J. Mar. Sci. Technol. 26. https://doi.org/10.6119/JMST.2018.02_(1).0014.

Li, H., Jiao, H., Yang, Z., 2023c. Ship trajectory prediction based on machine learning and deep learning: A systematic review and methods analysis. Eng. Appl. Artif. Intell. 126, 107062. https://doi.org/10.1016/j.engappai.2023.107062.

Li, H., Lam, J.S.L., Yang, Z., Liu, J., Liu, R.W., Liang, M., Li, Y., 2022b. Unsupervised hierarchical methodology of maritime traffic pattern extraction for knowledge discovery. Transp. Res. Part C Emerg. Technol. 143, 103856. https://doi.org/10.1016/j.trc.2022.103856.

Li, Y., Li, H., Zhang, C., Zhao, Y., Yang, Z., 2024c. Incorporation of adaptive compression into a GPU parallel computing framework for analyzing large-scale vessel trajectories. Transp. Res. Part C Emerg. Technol. 163, 104648. https://doi.org/10.1016/j.trc.2024.104648.

Li, Y., Liang, M., Li, H., Yang, Z., Du, L., Chen, Z., 2023d. Deep learning-powered vessel traffic flow prediction with spatial-temporal attributes and similarity grouping. Eng. Appl. Artif. Intel. 126, 107012. https://doi.org/10.1016/j.engappai.2023.107012.

Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J., 2022c. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. IEEE Trans. Neural Networks Learn. Syst. 33, 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827.

Liang, M., Liu, R.W., Zhan, Y., Li, H., Zhu, F., Wang, F.-Y., 2022. Fine-Grained Vessel Traffic Flow Prediction With a Spatio-Temporal Multigraph Convolutional Network. IEEE Trans. Intell. Transp. Syst. 23, 23694–23707. https://doi.org/10.1109/TITS.2022.3199160.

Liu, H., Wu, C., Li, B., Zong, Z., Shu, Y., 2025. Research on Ship Anomaly Detection Algorithm Based on Transformer-GSA Encoder. IEEE Trans. Intell. Transp. Syst. 1–12. https://doi.org/10.1109/TITS.2025.3536483.

Liu, H., Zhao, H., Wang, J., Yuan, S., Feng, W., 2022. LSTM-GAN-AE: A Promising Approach for Fault Diagnosis in Machine Health Monitoring. IEEE Trans. Instrum. Meas. 71, 1–13. https://doi.org/10.1109/TIM.2021.3135328.

Liu, Y., Gong, C., Yang, L., Chen, Y., 2020. DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. Expert Syst. Appl. 143, 113082. https://doi.org/10.1016/j.eswa.2019.113082.

Liu, Y., Lyu, C., Liu, X., Liu, Z., 2021. Automatic feature engineering for bus passenger flow prediction based on modular convolutional neural network. IEEE Trans. Intell. Transp. Syst. 22, 2349–2358. https://doi.org/10.1109/TITS.2020.3004254.

Livieris, I.E., Pintelas, E., Pintelas, P., 2020. A CNN–LSTM model for gold price time-series forecasting. Neural Comput. Applic. 32, 17351–17360. https://doi.org/10.1007/s00521-020-04867-x.

Lv, Y., Lv, Z., Cheng, Z., Zhu, Z., Rashidi, T.H., 2023. TS-STNN: Spatial-temporal neural network based on tree structure for traffic flow prediction. Transport. Res. Part e: Logist. Transportat. Rev. 177, 103251. https://doi.org/10.1016/j.tre.2023.103251.

Ma, C., Zhao, Y., Dai, G., Xu, X., Wong, S.-C., 2023. A Novel STFSA-CNN-GRU Hybrid Model for Short-Term Traffic Speed Prediction. IEEE Trans. Intell. Transp. Syst. 24, 3728–3737. https://doi.org/10.1109/TITS.2021.3117835.

Ma, Z., Mei, G., 2022. A hybrid attention-based deep learning approach for wind power prediction. Appl. Energy 323, 119608. https://doi.org/10.1016/j.apenergy.2022.119608.

Makowski, D., Naud, C., Jeuffroy, M.-H., Barbottin, A., Monod, H., 2006. Global sensitivity analysis for calculating the contribution of genetic parameters to the variance of crop model prediction. Reliab. Eng. Syst. Saf. 91, 1142–1147. https://doi.org/10.1016/j.ress.2005.11.015.

Mieczyńska, M., Czarnowski, I., 2021. DBSCAN algorithm for AIS data reconstruction. Procedia Computer Science, Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021 192, 2512–2521. Doi: 10.1016/j.procs.2021.09.020.

Mohamed, A., Dahl, G.E., Hinton, G., 2012. Acoustic Modeling Using Deep Belief Networks. IEEE Trans. Audio Speech Lang. Process. 20, 14–22. https://doi.org/10.1109/TASL.2011.2109382.

Mokhtarimousavi, S., Anderson, J.C., Azizinamini, A., Hadi, M., 2019. Improved support vector machine models for work zone crash injury severity prediction and analysis. Transp. Res. Rec. 2673, 680–692. https://doi.org/10.1177/0361198119845899.

Muruganantham, A., Tan, K.C., Vadakkepat, P., 2016. Evolutionary dynamic multiobjective optimization via kalman filter prediction. IEEE Trans. Cybern. 46, 2862–2873. https://doi.org/10.1109/TCYB.2015.2490738.

Nguyen, N., Quanz, B., 2021. Temporal latent auto-encoder: a method for probabilistic multivariate time series forecasting. Proceed. AAAI Conf. Artif. Intellig. 35, 9117–9125. https://doi.org/10.1609/aaai.v35i10.17101.

Niu, D., Yu, M., Sun, L., Gao, T., Wang, K., 2022. Short-term multi-energy load forecasting for integrated energy systems based on CNN-BiGRU optimized by attention mechanism. Appl. Energy 313, 118801. https://doi.org/10.1016/j.apenergy.2022.118801.

Okutani, I., Stephanedes, Y.J., 1984. Dynamic prediction of traffic volume through Kalman filtering theory. Transp. Res. B Methodol. 18, 1–11. https://doi.org/10.1016/0191-2615(84)90002-X.

Qin, M., Li, Z., Du, Z., 2017. Red tide time series forecasting by combining ARIMA and deep belief network. Knowl.-Based Syst. 125, 39–52. https://doi.org/10.1016/j.knosys.2017.03.027.

Ryu, U., Wang, J., Kim, T., Kwak, S., U, J., 2018. Construction of traffic state vector using mutual information for short-term traffic flow prediction. Transp. Res. Part C Emerging Technol. 96, 55–71. https://doi.org/10.1016/j.trc.2018.09.015.

Sadeghi Gargari, N., Panahi, R., Akbari, H., Ng, A.K.Y., 2022. Long-Term Traffic Forecast Using Neural Network and Seasonal Autoregressive Integrated Moving Average: Case of a Container Port. Transp. Res. Rec. 2676, 236–252. https://doi.org/10.1177/03611981221083311.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S., 2018. Recent Advances in Recurrent. Neural Netw. https://doi.org/10.48550/arXiv.1801.01078.

Shan, L., Liu, Y., Tang, M., Yang, M., Bai, X., 2021. CNN-BiLSTM hybrid neural networks with attention mechanism for well log prediction. Journal of Petroleum Science and Engineering 205, 108838. https://doi.org/10.1016/j.petrol.2021.108838.

Shu, Y., Huang, F., Wu, J., Chen, J., Song, L., Gan, L., Yang, Z., 2025. Research on Ship Following Behavior Based on Data Mining in Arctic Waters. IEEE Trans. Intell. Transp. Syst. 1–11.

Sousa, S.I.V., Martins, F.G., Alvim-Ferraz, M.C.M., Pereira, M.C., 2007. Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations. Environ. Model. Softw. 22, 97–103. https://doi.org/10.1016/j.envsoft.2005.12.002.

Sun, J., Wang, X., Xiong, N., Shao, J., 2018. Learning sparse representation with variational auto-encoder for anomaly detection. IEEE Access 6, 33353–33361. https://doi.org/10.1109/ACCESS.2018.2848210.

Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to Sequence Learning with Neural Networks, in: Advances in Neural Information Processing Systems. Curran Associates, Inc.

Tiwari, S.K., Kumaraswamidhas, L.A., Gautam, C., Garg, N., 2022. An auto-encoder based LSTM model for prediction of ambient noise levels. Appl. Acoust. 195, 108849. https://doi.org/10.1016/j.apacoust.2022.108849.

Turkoglu, M.O., D'Aronco, S., Wegner, J.D., Schindler, K., 2022. Gating Revisited: Deep Multi-Layer RNNs That can be Trained. IEEE Trans. Pattern Anal. Mach. Intell. 44, 4081–4092. https://doi.org/10.1109/TPAMI.2021.3064878.

Wang, H., Zhang, R., Cheng, X., Yang, L., 2022a. Hierarchical Traffic Flow Prediction Based on Spatial-Temporal Graph Convolutional Network. IEEE Trans. Intell. Transp. Syst. 23, 16137–16147. https://doi.org/10.1109/TITS.2022.3148105.

Wang, Q., Liu, W., Wang, X., Chen, X., Chen, G., Wu, Q., 2024. GMHANN: A Novel Traffic Flow Prediction Method for Transportation Management Based on Spatial-Temporal Graph Modeling. IEEE Trans. Intell. Transp. Syst. 25, 386–401. https://doi.org/10.1109/TITS.2023.3306559.

Wang, S., Shao, C., Zhang, J., Zheng, Y., Meng, M., 2022b. Traffic flow prediction using bi-directional gated recurrent unit method. Urban Info 1, 16. https://doi.org/10.1007/s44212-022-00015-z.

Wang, X., Li, J., Zhang, T., 2019. A Machine-Learning Model for Zonal Ship Flow Prediction Using AIS Data: A Case Study in the South Atlantic States Region. J. Mar. Sci. Eng. 7, 463. https://doi.org/10.3390/jmse7120463.

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

Wang, C., Zhang, X., Gao, H., Bashir, M., Li, H., Yang, Z., 2024. Optimizing anti-collision strategy for MASS: A safe reinforcement learning approach to improve maritime traffic safety. Ocean Coast. Manag. 253, 107161. https://doi.org/10.1016/j.ocecoaman.2024.107161.

Weerakody, P.B., Wong, K.W., Wang, G., Ela, W., 2021. A review of irregular time series data handling with gated recurrent neural networks. Neurocomputing 441, 161–178. https://doi.org/10.1016/j.neucom.2021.02.046.

Wei, W., Wu, H., Ma, H., 2019. An AutoEncoder and LSTM-Based Traffic Flow Prediction Method. Sensors 19, 2946. https://doi.org/10.3390/s19132946.

Williams, B.M., Durvasula, P.K., Brown, D.E., 1998. Urban Freeway Traffic Flow Prediction: Application of Seasonal Autoregressive Integrated Moving Average and Exponential Smoothing Models. Transp. Res. Rec. 1644, 132–141. https://doi.org/10.3141/1644-14.

Wolberg, G., 1988. Cubic Spline Interpolation: A Review. Doi: 10.7916/D82Z1DMQ.

Wu, D., Zhu, H., Zhu, Y., Chang, V., He, C., Hsu, C.-H., Wang, H., Feng, S., Tian, L., Huang, Z., 2020. Anomaly Detection Based on RBM-LSTM Neural Network for CPS in Advanced Driver Assistance System. ACM Trans. Cyber-Phys. Syst. 4, 27:1-27:17. Doi: 10.1145/3377408.

Xiao, H., Zhao, Y., Zhang, H., 2023. Predict Vessel Traffic with Weather Conditions Based on Multimodal Deep Learning. J. Mar. Sci. Eng. 11, 39. https://doi.org/10.3390/jmse11010039.

Xiao, Z., Fu, X., Zhang, L., Goh, R.S.M., 2020. Traffic pattern mining and forecasting technologies in maritime traffic service networks: a comprehensive survey. IEEE Trans. Intell. Transp. Syst. 21, 1796–1825. https://doi.org/10.1109/TITS.2019.2908191.

Xiao, Z., Ponnambalam, L., Fu, X., Zhang, W., 2017. Maritime Traffic Probabilistic Forecasting Based on Vessels' Waterway Patterns and Motion Behaviors. IEEE Trans. Intell. Transp. Syst. 18, 3122–3134. https://doi.org/10.1109/TITS.2017.2681810.

Xin, X., Liu, K., Li, H., Yang, Z., 2024. Maritime traffic partitioning: An adaptive semi-supervised spectral regularization approach for leveraging multi-graph evolutionary traffic interactions. Transp. Res. Part C Emerg. Technol. 164, 104670. https://doi.org/10.1016/j.trc.2024.104670.

Xin, X., Yang, Z., Liu, K., Zhang, J., Wu, X., 2023. Multi-stage and multi-topology analysis of ship traffic complexity for probabilistic collision detection. Expert Syst. Appl. 213, 118890. https://doi.org/10.1016/j.eswa.2022.118890.

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

Xu, J., Yuan, S., Shang, J., Wang, J., Yan, K., Yang, Y., 2024. Spatiotemporal Network Based on GCN and BiGRU for Seizure Detection. IEEE J. Biomed. Health Inform. 28, 2037–2046. https://doi.org/10.1109/JBHI.2024.3349583.

Xu, T., Zhang, Q., 2022. Ship traffic flow prediction in wind farms water area based on spatiotemporal dependence. J. Marine Sci. Eng. 10, 295. https://doi.org/10.3390/jmse10020295.

Yan, Z., Yang, H., Wu, Y., Lin, Y., 2023. A multi-view attention-based spatial–temporal network for airport arrival flow prediction. Transport. Res. Part e: Logist. Transport. Rev. 170, 102997. https://doi.org/10.1016/j.tre.2022.102997.

Yang, D., Liao, S., Venus Lun, Y.H., Bai, X., 2023. Towards sustainable port management: data-driven global container ports turnover rate assessment. Transport. Res. Part e: Logist. Transport. Rev. 175, 103169. https://doi.org/10.1016/j.tre.2023.103169.

Yang, J., Jiang, B., Song, H., Yang, X., Lu, W., Liu, H., 2018. No-reference stereoimage quality assessment for multimedia analysis towards internet-of-things. IEEE Access 6, 7631–7640. https://doi.org/10.1109/ACCESS.2018.2791560.

Yang, Y., Liu, Y., Li, G., Zhang, Z., Liu, Y., 2024. Harnessing the power of Machine learning for AIS Data-Driven maritime Research: A comprehensive review. Transport. Res. Part e: Logist. Transport. Rev. 183, 103426. https://doi.org/10.1016/j.tre.2024.103426.

Yang, Z., Tian, Y., Zhou, T., Zhu, Y., Zhang, P., Chen, J., Li, J., 2021. Time-series deep survival prediction for hemodialysis patients using an attention-based Bi-GRU network. Comput. Methods Programs Biomed. 212, 106458. https://doi.org/10.1016/j.cmpb.2021.106458.

Ye, B.-L., Zhang, M., Li, L., Liu, C., Wu, W., 2024. A Survey of Traffic Flow Prediction Methods Based on Long Short-Term Memory Networks. IEEE Intell. Transp. Syst. Mag. 16, 87–112. https://doi.org/10.1109/MITS.2024.3400679.

Yu, H., Meng, Q., Fang, Z., Liu, J., Xu, L., 2022. A review of ship collision risk assessment, hotspot detection and path planning for maritime traffic control in restricted waters. The Journal of Navigation 75, 1337–1363. https://doi.org/10.1017/S0373463322000650.

Yu, Y., Si, X., Hu, C., Zhang, J., 2019. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Comput. 31, 1235–1270. https://doi.org/10.1162/neco_a_01199.

Zha, W., Liu, Y., Wan, Y., Luo, R., Li, D., Yang, S., Xu, Y., 2022. Forecasting monthly gas field production based on the CNN-LSTM model. Energy 260, 124889. https://doi.org/10.1016/j.energy.2022.124889.

Zhan, Y., Hu, D., Wang, Y., Yu, X., 2018. Semisupervised Hyperspectral Image Classification Based on Generative Adversarial Networks. IEEE Geoscience and Remote Sensing Letters 15, 212–216. https://doi.org/10.1109/LGRS.2017.2780890.

Zhang, B., Zhang, H., Zhao, G., Lian, J., 2020. Constructing a PM2.5 concentration prediction model by combining auto-encoder with Bi-LSTM neural networks. Environ. Model. Softw. 124, 104600. https://doi.org/10.1016/j.envsoft.2019.104600.

Zhang, L., Wu, J., Shen, J., Chen, M., Wang, R., Zhou, X., Xu, C., Yao, Q., Wu, Q., 2021. SATP-GAN: self-attention based generative adversarial network for traffic flow prediction. Transportmetrica b: Transport Dynamics 9, 552–568. https://doi.org/10.1080/21680566.2021.1916646.

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

Zhang, W., Wu, Z., 2022. Optimal hybrid framework for carbon price forecasting using time series analysis and least squares support vector machine. J. Forecast. 41, 615–632. https://doi.org/10.1002/for.2831.

Zhang, Y., Xin, D., 2021. Short-term traffic flow prediction model based on deep learning regression algorithm. Int. J. Comput. Sci. Math. 14, 155–166. https://doi.org/10.1504/IJCSM.2021.118796.

Zhao, J., Yan, Z., Chen, X., Han, B., Wu, S., Ke, R., 2022. k-GCN-LSTM: A k-hop Graph Convolutional Network and Long–Short-Term Memory for ship speed prediction. Physica A 606, 128107. https://doi.org/10.1016/j.physa:2022.128107.

Zheng, H., Li, X., Li, Y., Yan, Z., Li, T., 2022. GCN-GAN: Integrating Graph Convolutional Network and Generative Adversarial Network for Traffic Flow Prediction. IEEE Access 10, 94051–94062. https://doi.org/10.1109/ACCESS.2022.3204036.

Zhou, H., Zuo, Y., Li, T., Li, S., 2021. Application of PSO-LSTM for Forecasting of Ship Traffic Flow, in: 2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC). Presented at the 2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), pp. 298–302. Doi: 10.1109/SPAC53836.2021.9539945.

Zhou, X., Liu, Z., Wang, F., Xie, Y., Zhang, X., 2020. Using Deep Learning to Forecast Maritime Vessel Flows. Sensors 20, 1761. https://doi.org/10.3390/s20061761.

Zhu, G., Feng, J., Li, P., Wang, Z., Wu, G., Luo, X., 2023. Multi-condition optimisation design of a hydrofoil based on deep belief network. Ocean Eng. 272, 113846. https://doi.org/10.1016/j.oceaneng.2023.113846.

Zissis, D., Xidias, E.K., Lekkas, D., 2016. Real-time vessel behavior prediction. Evol. Syst. 7, 29–40. https://doi.org/10.1007/s12530-015-9133-5.

Zou, Y., Zhu, T., Xie, Y., Zhang, Y., Zhang, Y., 2022. Multivariate analysis of car-following behavior data using a coupled hidden Markov model. Transp. Res. Part C Emerging Technol. 144, 103914. https://doi.org/10.1016/j.trc.2022.103914.