Fine-Grained Vessel Traffic Flow Prediction with a Spatio-Temporal Multi-Graph Convolutional Network

Maohan Liang, Ryan Wen Liu, *Member, IEEE,* Yang Zhan, Huanhuan Li, Fenghua Zhu, *Senior Member, IEEE,* and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—The accurate and robust prediction of vessel traffic flow is gaining importance in maritime intelligent transportation system (ITS), such as vessel traffic services, maritime spatial planning, and traffic safety management, etc. To achieve fine-grained vessel traffic flow prediction, we will first generate the maritime traffic network (*which is essentially a graph*), and then propose a graph-driven neural network. In particular, to represent various correlations among spatio-temporal vessel traffic flow, we tend to extract the feature points (i.e., starting, way and ending points) by utilizing the knowledge of vessel positioning data. These feature points are essentially related to the geometrical structures of massive vessel trajectories collected from massive automatic identification system (AIS) records, contributing to the generation of maritime traffic network. We then propose a spatio-temporal multi-graph convolutional network (STMGCN)-based vessel traffic flow prediction method by exploiting multiple types of inherent correlations in the generated maritime graph. The proposed STMGCN mainly contains one spatial multi-graph convolutional layer and two temporal gated convolutional layers, beneficial for extracting spatial and temporal traffic flow patterns. The main benefit of our graph-driven prediction method is that it takes full advantage of the maritime graph and multi-graph learning. Comprehensive experiments have been implemented on realistic AIS dataset to compare our method with several state-of-the-art prediction methods. The fine-grained prediction results have demonstrated our superior performance in terms of both accuracy and robustness.**

***Index Terms*—Traffic flow prediction, maritime traffic network, multi-graph convolutional network, automatic identification system (AIS)**

I. Introduction

THE accurate and robust prediction of vessel traffic flow plays a vital role in maritime intelligent transportation system (ITS) [1]. It is potentially beneficial for reducing ship collision accidents [2], enhancing navigational safety [3], and assisting port management [4], etc. With the widespread equipping of universal shipborne automatic identification system (AIS)[1], the reliable prediction of spatio-temporal vessel traffic flow through large-scale historical AIS data has gained increasing attention within both academia and industry.

The evolution of vessel traffic flow is inevitably affected by several different factors, such as special events (e.g., COVID-19), economic development, environmental changes, and adverse weathers (e.g., haze, low-lightness, snow and rain), etc. Due to these non-linear random effects, it is commonly intractable to accurately and robustly predict the vessel traffic flow using existing prediction methods. In the literature, the widely-used prediction methods could be mainly divided into two types, i.e., traditional computational methods [5], [6] and advanced learning methods [7], [8]. In particular, the traditional methods have the advantages of highly efficient operation and strong interpretability. However, they essentially ignore the inherent spatial-temporal correlations of vessel traffic flow, leading to unstable prediction results in practice. To eliminate these limitations, many studies [9], [10], [11] proposed to extend the concept of graph convolutional network (GCN) to traffic flow prediction. These learning methods creatively incorporate the topological structures of transportation network into the deep neural network. They could provide superior prediction results compared with existing traditional methods [12]. However, unlike traffic traffic network, there is no obvious road network structure, directly extracted from maritime ITS. Therefore, the powerful GCN-enabled traffic flow prediction methods, originally proposed for urban and highway driving scenarios, are not directly applicable in maritime transportation. In addition, existing GCN methods mainly consider the spatial accessibility between nodes in transportation networks, and ignore other correlational factors, e.g., historical, temporal and semantic correlations, etc. It is thus difficult to further improve GCN-based prediction results under complex traffic conditions. In summary, to achieve satisfactory vessel traffic flow prediction performance, it is necessary to handle two important problems as follows

1) How to reliably extract the maritime traffic network (essentially a graph) from massive historical AIS data?
2) How to develop more powerful GCN methods to accurately achieve fine-grained vessel traffic flow prediction?

This work was partially supported by the National Nature Science Foundation of China (Nos.: 51609195 and U1811463). (Corresponding authors: Ryan Wen Liu and Fenghua Zhu.)

M. Liang, R. W. Liu and Y. Zhan are with the School of Navigation, Wuhan University of Technology, Wuhan 430063, China, and also with the National Engineering Research Center for Water Transport Safety, Wuhan 430063, China (e-mail: {mhliang, wenliu, zhanyang}@whut.edu.cn).

H. Li is with the School of Engineering, Technology & Maritime Operations, Liverpool John Moores University, Liverpool L3 3AF, UK (e-mail: H.Li2@ljmu.ac.uk).

F. Zhu and F.-Y. Wang are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: fenghua.zhu@ia.ac.cn; feiyue@ieee.org).

[1]The AIS is an automatic tracking system that adopts transceivers on vessels. In particular, the AIS equipment can transmit dynamic and static information such as maritime mobile service identification (MMSI), vessel size, position (i.e., longitude and latitude), speed, heading, etc.

In this work, to conveniently measure the spatial-temporal correlations of vessel traffic flow, we first extract the maritime traffic network from large-scale historical AIS data. A spatio-temporal multi-graph convolutional network (STMGCN) is then proposed to accurately and robustly predict the vessel traffic flow within the maritime traffic network. The proposed STMGCN is capable of exploring the internal spatio-temporal correlation structures of nodes in the traffic network. The main contributions of this work are summarized as follows

1) *Extraction of Maritime Traffic Network*. To better represent the correlations among vessel traffic flow, a big data-driven computational method is proposed to automatically extract the maritime traffic network from large-scale historical AIS records.
2) *Multi-Graph Convolutional Neural Network*. To take full advantage of the correlations of traffic flow in maritime graph, an advanced multi-graph convolutional network is presented accordingly, leading to satisfactory prediction results under complex traffic conditions.
3) *Numerous Experiments with Realistic Dataset*. Comprehensive experiments on realistic AIS dataset have demonstrated that the proposed graph-based framework obviously outperformed several state-of-the-art prediction methods in terms of accuracy and robustness.

Our fine-grained traffic flow prediction method mainly benefits from the extracted maritime traffic network and STMGCN. Numerical experiments on realistic AIS data have illustrated the superior prediction performance of our graph-driven learning method under several different scenarios.

II. Related Works

In this section, we will briefly review the recent progresses from several perspectives, i.e., general traffic flow prediction methods (i.e., modeling and learning methods), and specific prediction methods for vessel traffic flow.

A. Modeling-Based Traffic Flow Prediction

Many traditional methods, which mainly include the auto-regressive integrated moving average (ARIMA) model [13], low-rank modeling [14], HoltWinter [15], data-driven filtering [16], and their extended versions [17], [18], have been developed to perform traffic flow prediction. The primary trend and periodicity features of original traffic flow can be captured through linear models, whose major benefits are simplicity and efficiency in practice. However, they are highly sensitive to chaotic and fractal characteristics of traffic, leading to low-accuracy prediction in practice. The Kalman filtering [16] has become a typical linear method to predict traffic flow. It has the capacity of continuously updating the estimate of state variables, which could improve the prediction accuracy and robustness [19]. In addition, the support vector regression (SVR)-based computational framework [20] has been widely utilized in traffic flow prediction in the presence of small samples. In practical applications, the original traffic flow data changes periodically and shows high spatio-temporal correlations. Therefore, the low-rank modeling [14] has also been introduced to reliably predict traffic flow.

B. Learning-Based Traffic Flow Prediction

Due to the strong feature learning capacity of neural network, different learning methods have attracted significant attention in traffic flow prediction. The learning methods have strong non-linear representation ability, potentially leading to high-accuracy prediction results. The representative methods mainly include the back-propagation neural network (BPNN) [21], [22], recurrent neural network (RNN) [23], [24], gated recurrent unit (GRU) [25], [26], [27], convolutional neural network (CNN) [28], [29], [30], stacked auto-encoder (SAE) [31], diffusion convolutional recurrent neural network (DCRNN) [32], and GCN [33], etc. The early studies mainly focused on the temporal correlation of traffic flow [22], [34], resulting in limited prediction reliability. To enhance the prediction performance, CNN has been widely adopted to capture the spatial correlation [12], [35]. Essentially, the CNN-based prediction methods are only appropriate in the Euclidean space. With the recent evolution of GCN [36], [37], various methods have shown satisfactory prediction results in the non-Euclidean space, such as, temporal graph convolutional network (T-GCN) [27], syntax and knowledge-based GCN (SK-GCN) [38], etc. In the field of road traffic flow prediction, the GCN-driven prediction methods have been widely exploited [27], [39], [40] since Kipf *et al.* [41] simplified the graph convolution. The spatio-temporal graph convolutional neural network (STGCN) was then proposed to capture both spatial and temporal correlations [42]. In STGCN, the entire convolutional structures were employed to capture the temporal dynamic behaviors of traffic flow along the time axis. In addition, a spatial graph convolutional layer was introduced to capture the spatial dynamic correlations. Similar network structures have been successfully adopted in GCN-driven computational frameworks [43], [44], [45], [46].

C. Vessel Traffic Flow Prediction

With the rapid developments of shipping business, regional and global economy, the reliable prediction of vessel traffic flow has attracted significant attention from both industry and academia. For example, to promote the accuracy of short-term prediction, He *et al.* [47] constructed a novel Kalman method by combining the regression analysis and Kalman filtering. Yu *et al.* [48] proposed to exploit three data mining approaches to estimate the delay or advance of vessel arrivals by utilizing the traffic-specific knowledge collected from historical shipping data. An improved particle swarm optimization (PSO)-back-propagation method [49] has also been presented to accurately predict the vessel traffic flow in the port water area. In addition, a novel multi-layer perception (MLP) network was developed to improve long-term vessel speed prediction results [50]. Li *et al.* [51] developed a novel hybrid method for optimizing SVR parameters to enhance traffic flow prediction. In the literature, the existing prediction methods do not take into account both spatial and temporal correlations among vessel traffic flow data in specific water areas. Therefore, it is computationally intractable to achieve fine-grained spatio-temporal prediction results under complex traffic conditions.

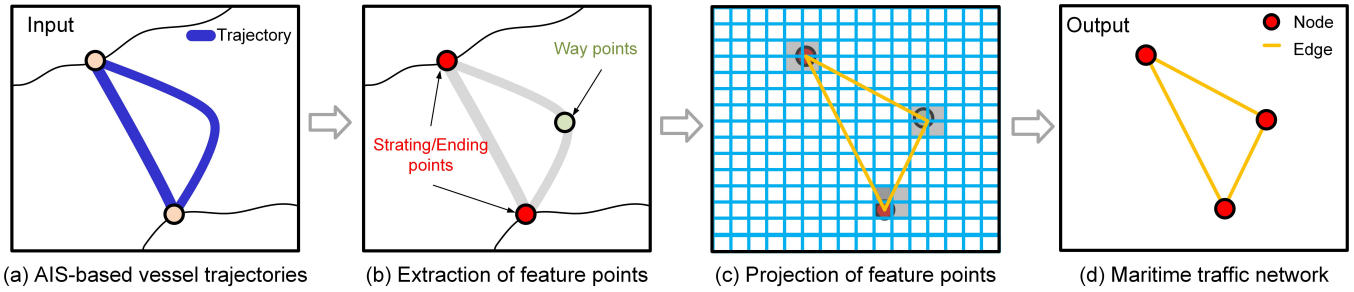


Fig. 1. The flowchart of generation of maritime traffic network. From left to right: (a) Collection of original AIS-based vessel trajectories, (b) Extraction of feature points, (c) Projection of feature points, and (d) Generation of maritime traffic network.

Algorithm 1 Extraction of Maritime Traffic Network.

Input: The set of trajectory, T_n ; The grid size, $M * N$;
Output: The maritime traffic network, $\mathcal{G}_O$;

for each vessel trajectory T_i in T_n **do**
  Extract starting, ending and way points from T_i by the DP
  Add starting, ending and way points into $\mathcal{P}$
end for
The research area is divided into $M \times N$ grid cells, and each cell is assigned a grid code that uniquely identifies the row and column ID to which it belongs
$\mathcal{P}$ is clustered into C clusters by DBSCAN
for each cluster F^c with $1 \leq c \leq C$ **do**
  for feature points $p_j \in F^c$ **do**
    Calculate the grid cell ID of each point
  end for
end for
Node $\mathcal{N}$ is defined as the region where grid cells contain the same cluster of feature points
The edges E are the connection relationships of the nodes
The adjacency matrix A_o^{ij} is calculated by Eq. (6)
return $\mathcal{G}_O = (\mathcal{N}, E, \mathcal{A}_O)$;

III. Extraction of Maritime Traffic Network

The internal spatio-temporal correlation among vessel traffic flow is usually highly related to the maritime traffic network. Accurate extraction of traffic nodes and edges is therefore an important prerequisite for effective prediction of vessel traffic flow. This section will introduce the details on how to extract the maritime traffic network from spatio-temporal AIS data.

A. Definitions

We will formally define the basic concepts of vessel trajectory, feature points (including starting, ending and way points), and maritime traffic network considered in this work.

Definition 1: Vessel Trajectory: The vessel trajectory T is always represented through a series of timestamped points $p_n = [\text{lon}_n, \text{lat}_n, t_n]$ with $n \in \{1, 2, \ldots, N\}$ recorded by AIS devices, i.e., $T = \{p_1, p_2, \ldots, p_N\}$ with N being the number of timestamped points in T. Here, t_n, lat_n and lon_n represent the time stamp, latitude, and longitude, respectively.

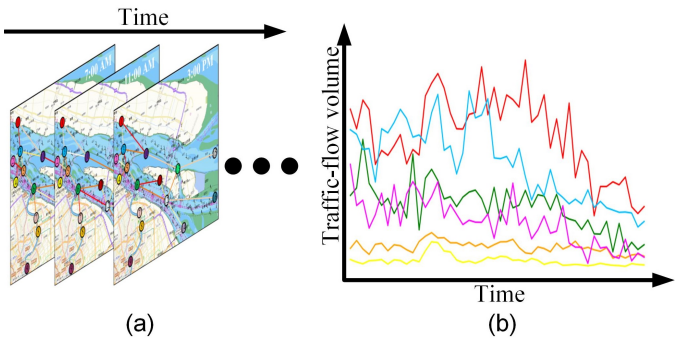


Fig. 2. The visual example of vessel traffic flow data. (a) The spatio-temporal structure of vessel traffic data, where the data at each time slice form a graph. (b) The vessel traffic flow observations at different graph nodes.

Definition 2: Starting and Ending Points: The starting and ending points, respectively, represent the coordinates where vessels begin and stop in the same shipping routes.

Definition 3: Way Points: The way points are isolated positions within the vessel trajectory where a vessel obviously changes its course.

Definition 4: Feature Points: The feature points are the clusters of starting, ending, and way points. Generally, the feature points contain fruitful vessel motion information.

Definition 5: Maritime Traffic Network: The maritime traffic network (a.k.a., maritime graph) is defined as an undirected graph $\mathcal{G} = (\mathcal{N}, E, \mathcal{A})$, visually illustrated in Fig. 1. Here, $\mathcal{N}$ denotes a finite set of nodes, E denotes a set of edges, indicating the connectivity between adjacent nodes, and $\mathcal{A} \in R^{N \times N}$ denotes the adjacency matrix. Each node in $\mathcal{G}$ records N measurements with the same sampling frequency. It means that each node generates a feature vector of length N at each time slice, shown by the solid lines in Fig. 2. More details on how to generate the maritime traffic network will be discussed in Section IV-B3.

The maritime traffic network will be generated through the massive historical AIS-based vessel trajectories. Based on this maritime traffic network, the fine-grained vessel traffic flow prediction could be guaranteed accordingly.

B. Extraction of Feature Points

To generate the maritime traffic network given in Definition 5, it is necessary to robustly and accurately extract the feature

points. In this work, the popular Douglas-Peucker (DP)-based compression algorithm, which captures the vessel directional changes based on spatial trajectory information, is directly used to roughly detect the starting, ending, and way points of vessel trajectories. The implementation of DP algorithm is abstractly formulated as follows

$$\{p_i^s, p_i^w, p_i^e\} \leftarrow \mathrm{DP}(T_i), \tag{1}$$

for $i = 1, 2, \cdots, I$. Here, I is the total number of collected vessel trajectories, T_i denotes the i-th vessel trajectory, p_i^s, p_i^w and p_i^e denote the starting, way, and ending points, respectively. The DP algorithm is iteratively performed to extract the starting, ending, and way points for each vessel trajectory.

These recorded starting, ending, and way points are coarse and inevitably suffer from undesirable biases and noise. To avoid these detrimental consequences, we exploit the density-based spatial clustering of applications with noise (DBSCAN) algorithm [52], [53] to distinguish outliers and cluster all the starting, ending, and way points into feature points. Among common spatial clustering algorithms, DBSCAN is always used because it does not need to set the number of clusters in advance, and is resistant to noise points. In addition, DBSCAN can find arbitrarily-shaped clusters. The implementation of DBSCAN illustrated in Fig. 1 (b), which can be abstractly described as follows

$$[F^1, \cdots, F^C] + O \leftarrow \mathrm{DBSCAN}\left(\mathcal{P}\right), \tag{2}$$

where F^c with $1 \leq c \leq C$ denotes the c-th cluster of feature points, $\mathcal{P}$ denotes the set of all starting, ending, and way points extracted from Eq. (1), and O denotes the outliers. More details and explanations on DP and DBSCAN algorithms can be found in Ref. [52].

C. Projection of Feature Points

The nodes of the maritime traffic network should be a region but not a precise coordinate point. To accurately construct the maritime traffic network, we project the feature points in grids, shown in Fig. 1 (c). We first divide the experimental water area into an $L \times W$ grid and project the feature points into this grid. The grids with the same cluster of feature point projections can be defined as a network node. Specifically, let $F^c = \{p_1, p_2, \cdots, p_N\}$ denote the c-th cluster of feature points, where N is the number of trajectory feature points. The p_n in F^c contains longitude lon_n^c, latitude lat_n^c, and time t_n^c. The mathematical relationship between $(\mathrm{lat}_n^c, \mathrm{lon}_n^c)$ in Cartesian coordinates and (l_n^c, w_n^c) in projected grids $\mathcal{J}$ are given by

$$l_n^c = \left\langle \frac{\mathrm{lat}_n^c - \mathrm{lat}_{\min}}{\mathrm{lat}_{\max} - \mathrm{lat}_{\min}} \cdot (L-1) \right\rangle + 1 \in [1, L], \tag{3}$$

$$w_n^c = \left\langle \frac{\mathrm{lon}_n^c - \mathrm{lon}_{\min}}{\mathrm{lon}_{\max} - \mathrm{lon}_{\min}} \cdot (W-1) \right\rangle + 1 \in [1, W], \tag{4}$$

where $\mathrm{lat}_{\max\,/\,\min}$ and $\mathrm{lon}_{\max\,/\,\min}$, respectively, denote the maximum/minimum latitude and longitude in the experimental water area, and $\langle\cdot\rangle$ denotes the cell operation. When the number of feature points (l_n, w_n) with projected coordinates equals d, the intensity of element (l_n^c, w_n^c) in $\mathcal{J}$ becomes $\mathcal{J}\left(l_n^c, w_n^c\right) = d$, essentially denoting the magnitude of projected image $\mathcal{J}$. The binary trajectory grid $\mathcal{X}$ is then generated by the intermediate image $\mathcal{J}$, whose definition is given by

$$\mathcal{X}_{ij} = \begin{cases} c, & \text{if } \mathcal{J}\left(w_n^c, h_n^c\right) > \hat{d}, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where $\hat{d}$ denotes a predefined threshold, which is empirically selected as 3 in this work.

D. Generation of Maritime Traffic Network

In this work, the node n_c of the maritime traffic network is defined as the grids in $\mathcal{X}_{ij}$ equal to c. The edges of the maritime traffic network are defined as the reachable relationship between adjacent nodes. If two nodes n_i and n_j connect in the origin graph, the corresponding value in the adjacency matrix $\mathcal{A}_O^{ij}$ is set to 1. Therefore, the adjacency matrix of the localized original maritime traffic network $\mathcal{G}_O$ can be formulated as follows

$$\mathcal{A}_O^{ij} = \begin{cases} 1, & \text{if } n_i \text{ connects to } n_j, i \neq j, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where n_i and n_j, respectively, denote the i- and j-th node in the localized original maritime traffic network. The maritime traffic network is visually illustrated in Fig. 1 (d). In this paper, node observations are obtained by counting the vessel traffic flow passing the node per unit time span.

IV. STMGCN: Spatio-Temporal Multi-Graph Convolutional Network

This section mainly focuses on how to model different types of spatio-temporal correlations through multiple graphs and how to predict vessel traffic flow based on STMGCN. The vessel traffic flow prediction is a classic time-series prediction task, i.e., estimation of the most likely traffic measurements in the following H time steps based on previous M traffic observations. In this paper, we regard the traffic information on the maritime traffic network as the attribute features of the node in the network. Thus, the prediction framework in this paper can be given by

$$[v_{t+1}, \cdots, v_{t+H}] = \mathcal{F}\left([v_{t-M+1}, \cdots, v_t]; \mathcal{G}_D, \mathcal{G}_I, \mathcal{G}_C\right), \tag{7}$$

where $v_t \in R^N$ represents the node observations (i.e., vessel traffic flow) in maritime traffic network $\mathcal{G}$ in the unit time span t, $\mathcal{F}\left(\cdot\right)$ denotes the mapping function. The $\mathcal{G}_D$, $\mathcal{G}_I$, and $\mathcal{G}_C$, respectively, represent the distance, interaction, and correlation graphs. For the sake of better understanding, we will first briefly introduce the basic concept of GCN, and then propose the STMGCN-based vessel traffic flow prediction method.

A. Overview of Graph Convolutional Network

CNN has received tremendous success in various applications, such as visual navigation [54], computer vision [55], and time series forecasting [56], etc. However, the classical CNN is only applicable in the Euclidean space. Therefore, it is intractable to perform CNN-based vessel traffic flow prediction in the (non-Euclidean) maritime graph. As an

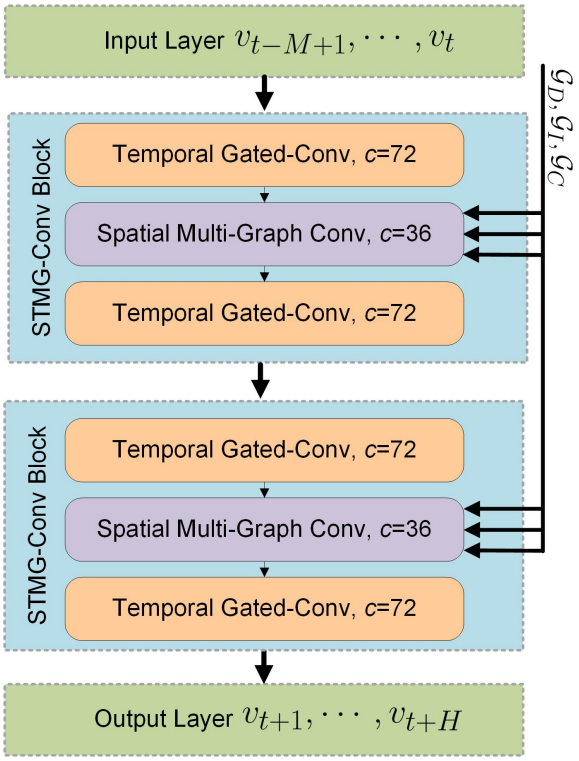


Fig. 3. The flowchart of spatio-temporal multi-graph convolutional network (STMGCN). The STMGCN contains two spatio-temporal multi-graph convolutional blocks (STMG-Conv blocks) and a fully connected output layer. There are one spatial multi-graph convolutional layer and two temporal gated convolutional layers in STMG-Conv block. The residual connection and bottleneck strategy are applied inside each STMG-Conv block.

advanced extension of CNN in the graph domain, GCN is capable of fully exploiting graph structure to learn the node representation of the generated graph. It has been successfully adopted in many non-Euclidean space applications, including trajectory prediction [57], natural language processing [58], and recommended systems [59], etc.

In this work, the convolution on graph is calculated in the Fourier domain. Let $U \in \mathbb{R}^{n \times n}$ denote the Fourier basis matrix of eigenvectors of the normalized Laplacian $\mathcal{L}$. We design to express the Laplace matrix $\mathcal{L}$ for a graph as follows

$$\mathcal{L} = I_n - D^{-\frac{1}{2}} \hat{\mathcal{A}} D^{-\frac{1}{2}} = U \Lambda U^T \in \mathbb{R}^{n \times n}, \tag{8}$$

with $D_{ii} = \sum_j \hat{\mathcal{A}}_{ij} \in \mathbb{R}^{n \times n}$. Here, $I_n \in \mathbb{R}^{n \times n}$ and $\hat{\mathcal{A}} \in \mathbb{R}^{n \times n}$, respectively, denote the identity matrix and weighted adjacency matrix of maritime graph $\mathcal{G}$.

The spectral graph convolution operator "$*\mathcal{G}$" is defined as the multiplication of signal $x \in \mathbb{R}^n$ with a kernel Θ. The Fourier transform for x is defined as $\hat{x} = U^T x$. Based on these definitions, x can be filtered by Θ with multiplication between Θ and $U^T x$. Therefore, the spectral graph convolution in this work is defined as follows

$$\Theta * \mathcal{G} x = \Theta(\mathcal{L}) x = \Theta\left(U \Lambda U^T\right) x = U \Theta(\Lambda) U^T x. \tag{9}$$

In general, the computational cost of kernel Θ in graph convolution is very high. To improve computational efficiency, Defferrard *et al.* [60] developed a quick localized technique

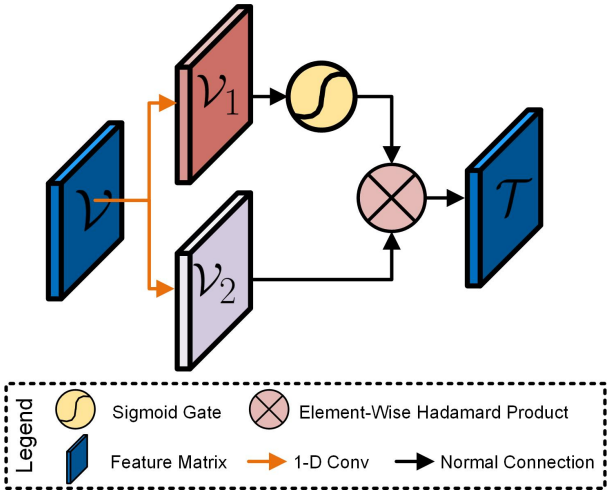


Fig. 4. Temporal gated convolutional layer. In this layer, the input feature matrix is processed by a 1-D causal convolutional layer with a Z width kernel and nonlinear GLUs.

to localize the filter and decrease the number of spectral graph convolution parameters. With the fast localized spectral filtering, the spectral graph convolution is given by

$$\Theta * \mathcal{G} x = \Theta(\mathcal{L}) x \approx \sum_{k=0}^{\mathcal{K}-1} \theta_k \beta_k \left(\hat{\mathcal{L}}\right) x, \tag{10}$$

with $\beta_k\left(\hat{\mathcal{L}}\right) \in \mathbb{R}^{n \times n}$ denoting the k-th order Chebyshev polynomial evaluated at the scaled Laplacian $\hat{\mathcal{L}} = 2\mathcal{L}/\lambda_{max} - I_n$. Here, $\theta \in \mathbb{R}^{\mathcal{K}}$, $\mathcal{K}$ and λ_{max}, respectively, denote the vector of polynomial coefficients, the kernel size of graph convolution, and the maximum eigenvalue of $\mathcal{L}$. The approximate expansion of the Chebyshev polynomial has been adopted to handle the spectral graph convolution. The computational cost of Eq. (8) can be accordingly reduced to $\mathcal{O}(\mathcal{K} \mid \varepsilon)$ as Eq. (9) shows.

B. Spatio-Temporal Multi-Graph Convolutional Network

1) Network Architecture. The detailed architecture of our STMGCN is introduced in this section. As shown in Fig. 3, STMGCN is mainly made up of two spatio-temporal multi-graph convolutional blocks (STMG-Conv block). Each STMG-Conv block consists of two temporal gated convolutional layers and one spatial multi-graph convolutional layer. As shown in Fig. 4, we use the temporal gated convolutional layer to extract the temporal correlation of vessel traffic flow. The spatial multi-graph convolutional layer is exploited to learn the spatial correlation of graph nodes, illustrated in Fig. 5. The temporal and spatial convolutional layers are alternatively performed to better fuse the spatio-temporal correlations.

2) Gated CNN for Extracting Temporal Correlations. As illustrated in Fig. 4, we employ a 1-D temporal convolutional layer to capture the temporal correlations of original traffic flow. The temporal convolutional layer contains a 1-D causal convolutional layer with a kernel of width $\mathcal{K}_t$ and nonlinear gated linear units (GLUs). Given the input of temporal convolutional layer $\mathcal{V} \in \mathbb{R}^{n \times m \times c_{in}}$, where n, m, c_{in}, respectively, denote the width, length and input channel dimensions. The

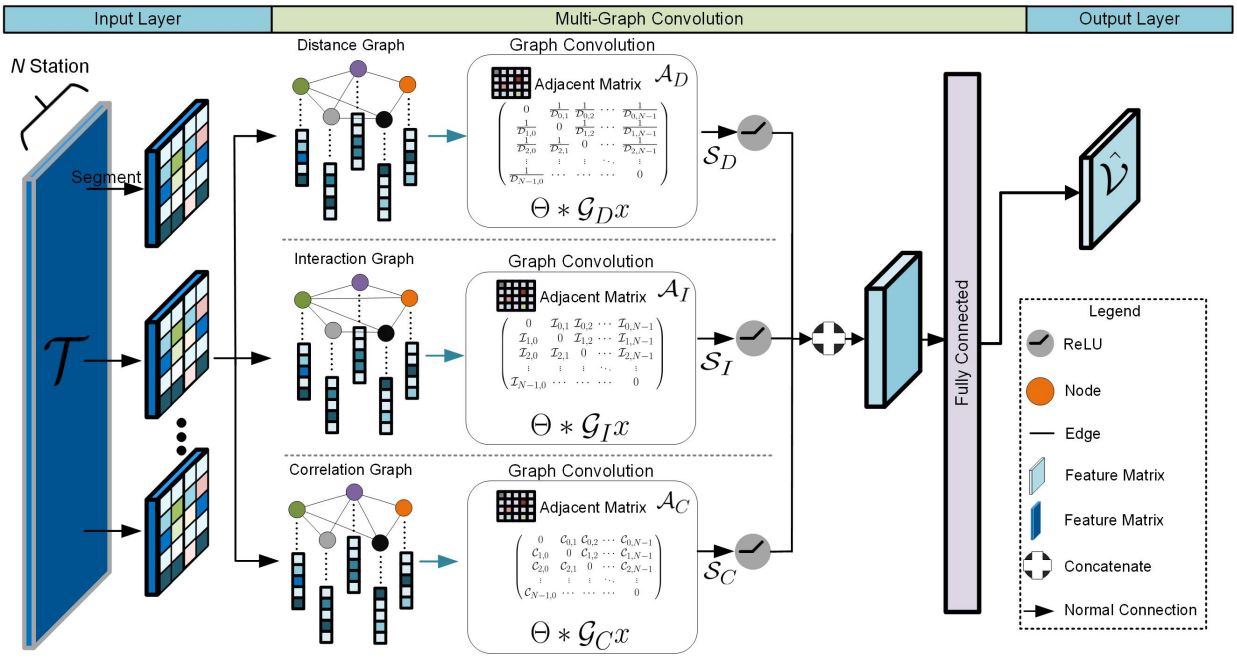


Fig. 5. Spatial multi-graph convolutional layer. In this layer, each input feature is processed by three graph convolutional layers. Then, the feature matrix will be concatenated together. Finally, the concatenated feature matrix is mapped to the same size as the input features using fully-connected layers.

convolutional kernel $\Gamma \in \mathbb{R}^{\mathcal{K}_t \times c_{in} \times 2c_{out}}$ maps $\mathcal{V}$ to an output element $[\mathcal{V}_1, \mathcal{V}_2] \in \mathbb{R}^{n \times (m-\mathcal{K}_t+1) \times (2c_{out})}$ ($\mathcal{V}_1$, $\mathcal{V}_2$ are split into half with the same size of channels), where c_{out} denotes the output channel dimensions. The temporal gated convolutional is calculated by

$$\mathcal{T} = \Gamma *_\tau \mathcal{V} = \mathcal{V}_1 \odot \sigma(\mathcal{V}_2) \in \mathbb{R}^{n \times (m-\mathcal{K}_t+1) \times c_{out}}, \quad (11)$$

where $\mathcal{V}_1, \mathcal{V}_2 \in \mathbb{R}^{(m-\mathcal{K}_t+1) \times (2c_{out})}$ denote the separate inputs of GLU, and $\odot$ denotes the elementwise Hadamard product. The sigmoid gate $\sigma(\mathcal{V}_2)$ is exploited to recognize the compositional structure and dynamic variations in the time series from the current state input $\mathcal{V}_1$. The sigmoid gate $\sigma(\mathcal{V}_2)$ decides which current status inputs $\mathcal{V}_1$ are important for recognizing the compositional structure and dynamic variations in the time series.

3) Gated Multi-Graph CNN for Extracting Spatial Correlations. To fuse spatial correlations into our network, the spatial multi-graph convolutional layer is developed to jointly process graph-structured time series, shown in Fig. 3. This layer can be stacked and expanded according to the scale and complexity of original vessel traffic flow.

As shown in Fig. 3, the spatial multi-graph convolutional layer is a link between two temporal gated convolutional layers. To capture node interactions, we use the spatial multi-graph convolutional layer to perform the convolution operation. The STMGCN relies heavily on graph construction. If the produced graphs are unable to adequately describe the correlations between graph nodes, it would become harmful to model learning, and even decrease the prediction performance. Based on this concept, we construct three spatial correlation graphs (i.e., distance graph $\mathcal{G}_D$, interaction graph $\mathcal{G}_I$, and correlation graph $\mathcal{G}_C$) to describe the spatial correlation between nodes in the extracted maritime graph, which will be detailedly discussed as follows.

Distance Graph. From the Waldo Tobler's First Law of Geography [61], it is easy to observe that any two nodes near each other in a maritime graph may share the similar navigation behaviors. Motivated by this assumption, we design to reconstruct the distance graph $\mathcal{G}_D(\mathcal{N}, E, \mathcal{A}_D)$ to describe the distance between nodes. In particular, The reciprocal of the distance is used to express the weight between two nodes. Thus, the closer the nodes, the higher the weight. According to the Algorithm 1, we use the centroids of graph nodes to calculate the distance between different nodes. Let $\mathcal{D}_{ij}$ denote the Euclidean distance between the centroids of nodes n_i and n_j. The adjacent matrix $\mathcal{A}_D$ of distance graph $\mathcal{G}_D$ can be represented as follows

$$\mathcal{A}_D^{ij} = \begin{cases} \frac{1}{\mathcal{D}_{ij}} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}, \quad (12)$$

where $\mathcal{A}_D^{ij}$ represents the weight of distance graph $\mathcal{G}_D$ corresponding to the i-th row and j-th column.

Interaction Graph. The historical traffic flow between nodes has a crucial influence on constructing the spatial correlation graphs. For instance, if there exist many navigation records between the nodes n_i and n_j, these two nodes n_i and n_j then tend to influence each other regarding the dynamic vessel traffic flow patterns. Following this logic, we create an interaction graph $\mathcal{G}_I(\mathcal{N}, E, \mathcal{A}_I)$ to represent how frequently two nodes interact with each other based on historical navigation records. The traffic flow between any two nodes can be easily counted, according to the method proposed in Section III. Then, the

TABLE I
THE VARIANTS OF STMGCN USING IN THIS PAPER. THE '✓' INDICATES THE STGCN OR STMGCN VARIANTS CONSIDERING THE CORRESPONDING SPATIAL CHARACTERISTICS. THE "X" AND "Y", RESPECTIVELY, INDICATE THE STGCN AND STMGCN.

	Methods									
Graph Types	X-O	X-D	X-I	X-C	Y-OD	Y-OI	Y-OC	Y-DI	Y-DC	Y-IC
$\mathcal{G}_O$	✓				✓	✓	✓			
$\mathcal{G}_D$		✓			✓			✓	✓	
$\mathcal{G}_I$			✓			✓		✓		✓
$\mathcal{G}_C$				✓			✓		✓	✓

adjacent matrix $\mathcal{A}_I$ of $\mathcal{G}_I$ can be represented as follows

$$\mathcal{A}_I^{ij} = \begin{cases} \mathcal{I}_{ij} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}, \tag{13}$$

where $\mathcal{I}_{ij}$ denotes the number of maritime traffic flow records between the i-th and j-th nodes.

Correlation Graph. The historical usage (i.e., inflow or outflow) of each node in each time segment plays an important role in the construction of spatial correlation graphs. To better consider the historical usage of nodes, we construct a correlation graph $\mathcal{G}_C(\mathcal{N}, E, \mathcal{A}_C)$ through the inflow or outflow correlation between nodes. In this paper, we utilize the well-known Pearson coefficient to determine the correlation. Pearson coefficient is a popular test statistics that is used to measure the statistical relationship, or association, between two datasets. Let X and Y denote the traffic flow volume at node n_i and node n_j, the Pearson coefficient is then given by

$$\mathcal{C}_{ij} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}, \tag{14}$$

where $Cov(\cdot,\cdot)$ denotes the covariance between two vectors. Here, σ_X and σ_Y, respectively, denote the standard deviations of X and Y. Then, we can represent the adjacent matrix $\mathcal{A}_C$ of $\mathcal{G}_C$ as follows

$$\mathcal{A}_C^{ij} = \begin{cases} \mathcal{C}_{ij} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}, \tag{15}$$

Spatial Multi-Graph Convolution. The process of spatial multi-graph convolution is shown in Fig. 5. Given the input data of the graph convolutional layer as $\mathcal{T} \in \mathbb{R}^{n \times m \times c_{in}}$, where n, m, c_{in}, respectively, denote the size of the spatial, temporal, and channel dimensions. We can represent the propagation rule of spatial graph convolution as follows

$$\mathcal{S} = \psi(\Theta * \mathcal{G}\mathcal{T}) \in \mathbb{R}^{n \times m \times c_{out}}, \tag{16}$$

where $\psi(\cdot)$ denoting the ReLU activation function, and Θ denotes the spectral kernel of graph convolution. Each traffic flow segment is input to three GCN modules, together with the corresponding maritime graphs (i.e., $\mathcal{G}_D$, $\mathcal{G}_I$, and $\mathcal{G}_C$), resulting in three feature matrices, denoted as $\mathcal{S}_D$, $\mathcal{S}_I$, and $\mathcal{S}_C$. We use the Chebyshev polynomial to estimate kernels to speed up the spectral graph convolution operation in this work. The three feature matrices are concatenated as the input of a fully connected layer, and the output $\hat{\mathcal{V}}$ of the final spatial multi-graph convolution is calculated by

$$\hat{\mathcal{V}} = \psi(W[\mathcal{S}_D, \mathcal{S}_I, \mathcal{S}_C] + \mathcal{B}) \in \mathbb{R}^{n \times m \times c_{out}}, \tag{17}$$

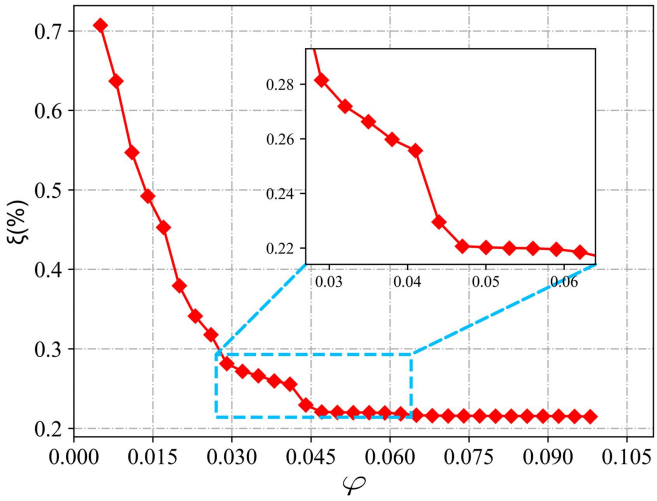


Fig. 6. Influences of parameter φ on detection of starting, ending, and way points. Here, ξ denotes the ratio of compression (the proportion of remaining timestamped points).

where $[\cdot,\cdot,\cdot]$ represents the concatenation operation, W and $\mathcal{B}$, respectively, denote the weighting matrix and bias.

4) Spatio-Temporal Multi-Graph Convolutional Block. As previously mentioned, each STMG-Conv block consists of two temporal gated convolutional layers and one spatial multi-graph convolutional layer. In this paper, the temporal gated convolutional layer and spatial multi-graph convolutional layer are run interleaved. Given the input traffic measurements $\mathcal{V}^l$, one STMG-Conv block computes the predicted traffic measurements $\mathcal{V}^{l+1}$ by the following equations

$$\mathcal{T}^l = \Gamma *_\tau \mathcal{V}^l, \tag{18}$$

$$\hat{\mathcal{V}}^l = W\left[\psi\left(\Theta * \mathcal{G}_D\mathcal{T}^l\right), \psi\left(\Theta * \mathcal{G}_I\mathcal{T}^l\right), \psi\left(\Theta * \mathcal{G}_C\mathcal{T}^l\right)\right] + \mathcal{B}, \tag{19}$$

$$\mathcal{V}^{l+1} = \Gamma *_\tau \hat{\mathcal{V}}^l. \tag{20}$$

In this paper, the STMGCN consists two STMG-Conv blocks. Thus, we will repeat this process twice to get the predicted traffic measurements. Theoretically, the STMG-Conv module can be extended to more other models and applications.

V. EXPERIMENTS AND ANALYSIS

In this section, various experiments will be conducted on realistic data to validate the proposed method. Firstly, we will briefly describe the research water regions and data used in this paper. The quantitative metrics and comparison baselines are then presented in detail. Finally, we implement

TABLE II

COMPARISONS OF BASELINE METHODS FOR VESSEL TRAFFIC FLOW PREDICTION. IN THIS PAPER, EACH PREDICTION MODEL IS RUN TEN TIMES TO DECREASE RANDOMNESS. THE RMSE, MAE AND R^2 (MEAN $\pm$ STANDARD DEVIATION) ARE SIMULTANEOUSLY EXPLOITED TO EVALUATE THE VESSEL TRAFFIC PREDICTION RESULTS.

Unit Time Spans	Evaluation Metrics	Models							
		HA [62]	SVR [20]	BPNN [22]	LSTM [23]	GRU [25]	T-GCN [27]	DCRNN[32]	STMGCN
1h	RMSE↓	9.03 ± 7.19	8.66 ± 7.76	8.47 ± 7.73	7.71 ± 6.69	7.84 ± 6.79	7.61 ± 5.84	7.35 ± 6.91	$\mathbf{6.46 \pm 0.96}$
	MAE↓	5.94 ± 4.59	5.48 ± 4.80	6.25 ± 5.98	5.48 ± 4.77	5.63 ± 4.89	4.73 ± 3.24	4.09 ± 5.01	$\mathbf{3.63 \pm 0.43}$
	R^2↑	*	0.17 ± 0.27	0.17 ± 0.29	0.24 ± 0.30	0.19 ± 0.32	0.42 ± 0.27	0.50 ± 0.60	$\mathbf{0.66 \pm 0.08}$
2h	RMSE↓	9.03 ± 7.19	8.69 ± 7.73	8.75 ± 8.01	7.89 ± 6.94	8.16 ± 7.37	8.10 ± 6.54	8.37 ± 7.25	$\mathbf{7.19 \pm 1.47}$
	MAE↓	5.94 ± 4.59	5.46 ± 4.76	6.45 ± 6.28	5.62 ± 4.98	6.00 ± 5.71	5.14 ± 3.78	6.11 ± 5.33	$\mathbf{3.86 \pm 0.44}$
	R^2↑	*	0.15 ± 0.26	0.11 ± 0.30	0.18 ± 0.32	0.17 ± 0.31	0.36 ± 0.24	0.30 ± 0.60	$\mathbf{0.60 \pm 0.09}$
3h	RMSE↓	9.03 ± 7.19	8.77 ± 7.74	8.86 ± 8.09	8.09 ± 7.03	8.37 ± 7.38	8.41 ± 6.77	8.37 ± 8.05	$\mathbf{7.59 \pm 2.19}$
	MAE↓	5.94 ± 4.59	5.51 ± 4.79	6.51 ± 6.35	5.78 ± 5.07	6.13 ± 5.61	5.32 ± 3.95	5.10 ± 6.02	$\mathbf{4.16 \pm 0.72}$
	R^2↑	*	0.13 ± 0.26	0.10 ± 0.31	0.12 ± 0.35	0.12 ± 0.34	0.33 ± 0.28	0.32 ± 0.44	$\mathbf{0.56 \pm 0.11}$

TABLE III

COMPARISONS OF VESSEL TRAFFIC FLOW PREDICTION ON RE FOR DIFFERENT COMPETING METHODS. EACH PREDICTION MODEL IS RUN TEN TIMES TO DECREASE RANDOMNESS.

Nodes	Models							
	HA [62]	SVR [20]	BPNN [22]	LSTM [23]	GRU [25]	T-GCN[27]	DCRNN[32]	STMGCN
1	0.8380 ± 0.000	0.8030 ± 0.163	0.8303 ± 0.072	0.8197 ± 0.103	0.8603 ± 0.064	0.4378 ± 0.015	0.3675 ± 0.055	$\mathbf{0.2711 \pm 0.008}$
2	0.8382 ± 0.000	0.8023 ± 0.138	0.8255 ± 0.109	0.6255 ± 0.092	0.7390 ± 0.067	0.5743 ± 0.056	0.3614 ± 0.039	$\mathbf{0.2539 \pm 0.009}$
3	0.4940 ± 0.000	0.5100 ± 0.124	0.4661 ± 0.136	0.4374 ± 0.079	0.4423 ± 0.073	0.3179 ± 0.062	0.5470 ± 0.026	$\mathbf{0.2998 \pm 0.005}$
4	0.5340 ± 0.000	0.4867 ± 0.063	0.5099 ± 0.049	0.4844 ± 0.063	0.4810 ± 0.062	0.4153 ± 0.053	0.3852 ± 0.018	$\mathbf{0.3227 \pm 0.012}$
5	0.3329 ± 0.000	0.3021 ± 0.054	0.4060 ± 0.055	0.4027 ± 0.046	0.3777 ± 0.052	0.3294 ± 0.044	0.3550 ± 0.024	$\mathbf{0.2650 \pm 0.020}$
6	0.8529 ± 0.000	0.7652 ± 0.150	0.8046 ± 0.126	0.7165 ± 0.134	0.7465 ± 0.137	0.4185 ± 0.122	0.5059 ± 0.118	$\mathbf{0.3260 \pm 0.019}$
7	1.0547 ± 0.000	0.9617 ± 0.129	1.2194 ± 0.105	1.0325 ± 0.134	1.3391 ± 0.157	0.4808 ± 0.110	1.3598 ± 0.097	$\mathbf{0.4723 \pm 0.011}$
8	0.6807 ± 0.000	0.7443 ± 0.126	0.6788 ± 0.091	0.6363 ± 0.082	0.6748 ± 0.075	0.5474 ± 0.071	0.6833 ± 0.039	$\mathbf{0.4438 \pm 0.015}$
9	0.6895 ± 0.000	0.7233 ± 0.080	0.6492 ± 0.109	0.5617 ± 0.061	0.6492 ± 0.081	0.5328 ± 0.072	0.9284 ± 0.079	$\mathbf{0.4706 \pm 0.015}$
10	0.7591 ± 0.000	0.8348 ± 0.083	0.8405 ± 0.078	0.7191 ± 0.057	0.7591 ± 0.059	0.5824 ± 0.054	0.6330 ± 0.042	$\mathbf{0.4455 \pm 0.015}$
11	0.8331 ± 0.000	0.8699 ± 0.108	0.7741 ± 0.131	0.8214 ± 0.121	0.8204 ± 0.103	0.6247 ± 0.112	0.6571 ± 0.149	$\mathbf{0.5475 \pm 0.021}$
12	0.8140 ± 0.000	0.8180 ± 0.089	0.8046 ± 0.089	0.6852 ± 0.080	0.7029 ± 0.072	0.5834 ± 0.070	0.4507 ± 0.039	$\mathbf{0.4392 \pm 0.028}$
13	0.7910 ± 0.000	0.8365 ± 0.247	0.8255 ± 0.149	0.7234 ± 0.148	0.8405 ± 0.135	0.7244 ± 0.121	0.4984 ± 0.035	$\mathbf{0.4213 \pm 0.015}$
14	0.9346 ± 0.000	0.8201 ± 0.083	0.8303 ± 0.075	0.8197 ± 0.047	0.8603 ± 0.069	0.4199 ± 0.042	0.5516 ± 0.035	$\mathbf{0.3928 \pm 0.028}$
15	0.7714 ± 0.000	0.7511 ± 0.157	0.7011 ± 0.153	0.9189 ± 0.110	1.0672 ± 0.106	0.5940 ± 0.095	0.9083 ± 0.052	$\mathbf{0.4362 \pm 0.036}$
16	0.7435 ± 0.000	0.8173 ± 0.136	0.7189 ± 0.105	0.9011 ± 0.134	0.9672 ± 0.137	0.5537 ± 0.105	0.6396 ± 0.047	$\mathbf{0.3579 \pm 0.028}$

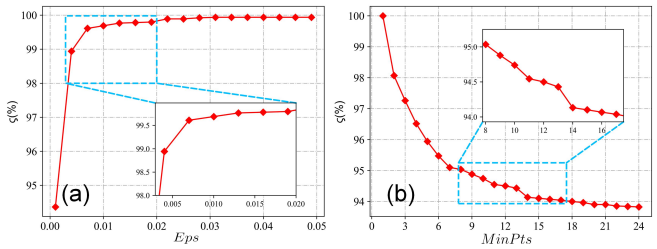


Fig. 7. Influences of DBSCAN parameters (Eps and $MinPts$) on extraction of feature points. Here, ς denotes the ratio of utilization (the proportion of remaining timestamped points). (a) The influences of Eps, (b) The influences of $MinPts$.

experiments on realistic scenarios to demonstrate the superior performance of our maritime traffic network extraction methods and STMGCN-based vessel traffic flow prediction methods.

A. Dataset Description

We collected original vessel trajectories from the terrestrial AIS base stations in the South Channel of Yangtze River Estuary, within longitude $121°23' - 121°59'$ east and latitude $31°07' - 31°34'$ north. The temporal sampling rate of the collected data is 2-30 s. The dataset includes a total of $49,532$ vessel trajectories and $61,994,075$ trajectory data points. The time period used was from 31 July to 31 August 2017. For the sake of better comparisons, the qualities of original trajectories were improved directly by utilizing the existing methods proposed in our previous work [52]. We implemented all experiments in PyTorch 1.0 with the NVIDIA GeForce RTX 2060 GPU, Intel Core i7-9750H CPU (2.6 GHz), 16 GB of RAM, and Windows 10 64-bit operating system.

B. Evaluation Metrics and Baselines

1) Evaluation Metrics. To quantitatively evaluate the prediction results, the root mean squared error (RMSE), coefficient of determination (R^2), mean absolute error (MAE), and relative error (RE) were jointly employed in our experiments. In particular, the proposed model and all baseline model were run 10 times to reduce the randomness in prediction evaluation. In addition, we also calculated the mean and variance of each evaluation metrics to measure the traffic prediction in terms of accuracy and robustness. Generally, the smaller values of RMSE, MAE and RE indicate better prediction performance. $R^2 \in [0, 1]$ shows how well the data fit the regression model (goodness of fit). Theoretically, the more accurate prediction results could be obtained with the higher R^2.

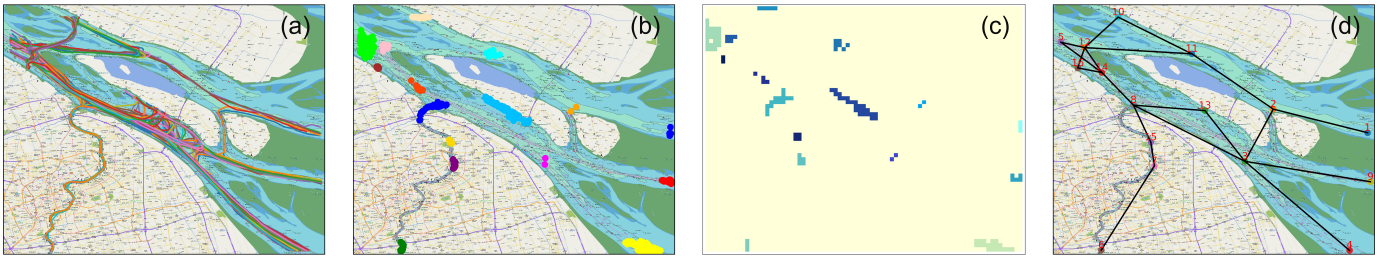


Fig. 8. Extraction results of maritime traffic network. From left to right: (a) Visualization results of vessel trajectories in experimental water area, (b) extraction results of feature points, (c) projection results of feature points, and (d) generation results of maritime traffic network.

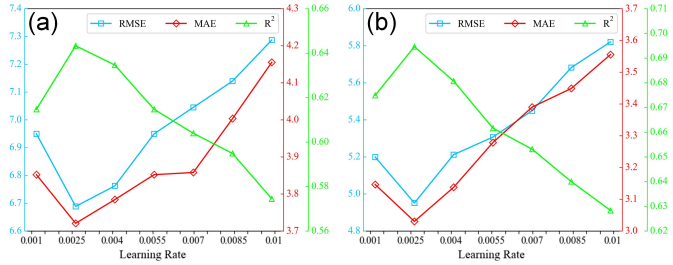


Fig. 9. Comparisons of predicted performance over different learning rates in the training and test sets. (a) The changes of RMSE, MAE, and R^2 on the training set. (b) The changes of RMSE, MAE, and R^2 on the test set.

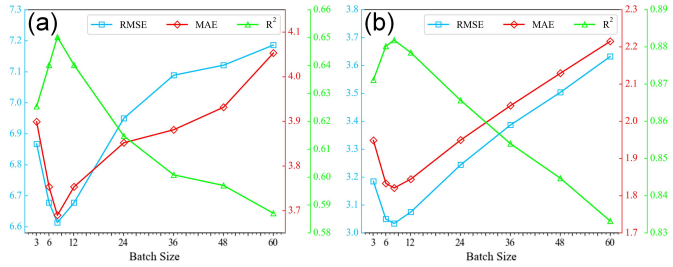


Fig. 10. Comparisons of predicted performance over different learning rates in the training and test sets. (a) The results of RMSE, MAE, and R^2 on the training set. (b) The results of RMSE, MAE, and R^2 on the test set.

2) Baselines. To fairly evaluate the performance of our STMGCN, we compared the proposed method with recent novel baseline methods, including history average (HA) [62], SVR [20], BPNN [22], long short-term memory (LSTM) [23], GRU [25], T-GCN[27], and DCRNN[32]. It is worth noting that we employ the default parameter from the original implementations for all baseline methods. In addition, we attempt to evaluate the effectiveness of the different spatial correlation graphs of STMGCN in our experiments, i.e., the four maritime graphs described in Sections III-D and IV-B3. This paper compares 10 variants of STMGCN, as shown in Table I.

C. Maritime Traffic Network Extraction Results

1) Parameter Settings. Both DP and DBSCAN methods are employed to assist in the extraction of maritime traffic networks, as illustrated in Section III-B. The extraction results are highly dependent on the input settings used in DP and DBSCAN. It is thus important to choose the compression threshold (i.e., φ) for DP compression, the neighbourhood radius (i.e., Eps), and the minimal number of points inside the Eps radius (i.e., $MinPts$) for optimal DBSCAN clustering. Fig. 6 depicts the effects of DP thresholds $\varphi \in [0, 0.10)$ on AIS-based vessel trajectory compression. It is apparent that when φ increases, the fraction of surviving timestamped points decreases dramatically. A larger proportion could retain more timestamped points in vessel trajectories, resulting in redundant way points, whereas a smaller proportion may result in the removal of certain way points owing to trajectories being oversimplified. We empirically set $\varphi = 0.045$ to extract the feature points by compressing the vessel trajectories to obtain a reasonable equilibrium.

During the generation of maritime traffic networks, DBSCAN could assist in removing undesired outliers from typical feature points. It should be noted that the parameters Eps and $MinPts$ have a considerable influence on the automated detection of outliers. However, it is difficult to determine the ideal settings directly. This paper proposes exhaustive experiments to empirically identify the best values of Eps and $MinPts$. Eps and $MinPts$, in particular, span the ranges $[0, 0.05]$ and $[2, 98]$, respectively. The impact of different parameters Eps and $MinPts$ on automatic detection of outliers is clearly depicted in Fig. 7. As Eps increases, the proportion of retained feature points increase and then gradually settle. In comparison, when $MinPts$ rises, the proportion of the remaining feature points varies slowly. According to the experimental results, we empirically choose the optimal parameters $Eps = 0.01$ and $MinPts = 14$ to guarantee detection stability. In addition, we manually set the spatial interval $\Delta = 0.0075°$ (i.e., grid size $L \times W = 79 \times 60$).

2) Network Extraction Results. In this work, we developed a data-driven multi-step computational framework to extract the maritime traffic networks based on massive historical AIS data, as illustrated in Section III. Fig. 8 shows the extraction result of the maritime traffic network in South Channel of Yangtze River Estuary. Fig. 8 (a) demonstrates the original vessel trajectories considered in this work. As shown in Fig. 8 (b), the feature points, including starting, ending, and way points, are sufficiently estimated via DP compression and DBSCAN. Different colored points in Fig. 8 (b) represent different clusters of feature points. Benefiting from the feature point extraction process, it becomes more feasible to accurately obtain nodes of the graph in Fig. 8 (c). The result of the maritime traffic network is shown in Fig. 8 (d). In particular, the points with numbers and the black lines

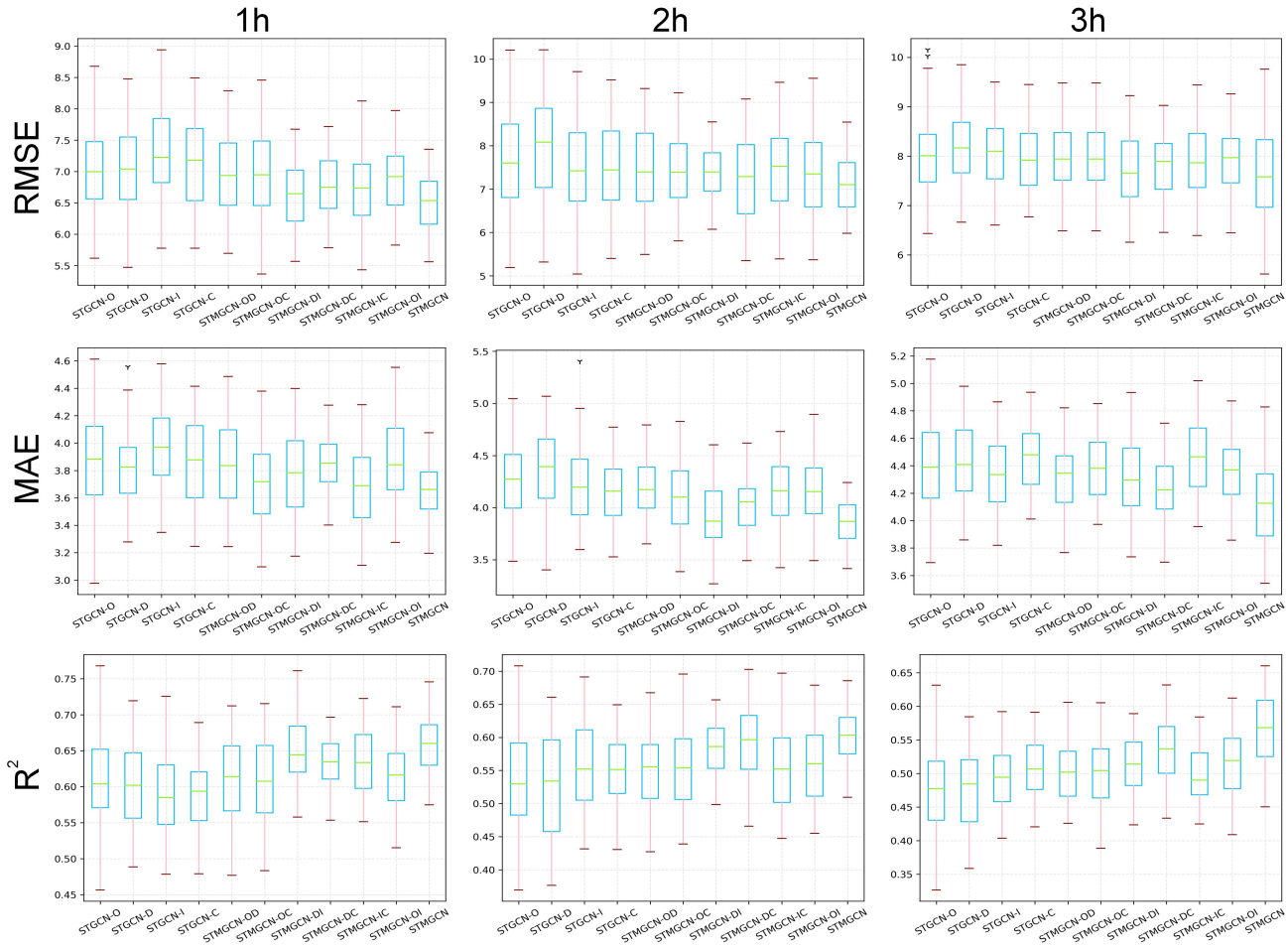


Fig. 11. Comparisons of predicted performance (RMSE, MAE, and R^2) over different unit time spans (1 h/2 h/3 h). Each prediction model is run ten times to decrease randomness.

in Fig. 8 (d), respectively, denote the nodes and edges. It can be seen that the original maritime traffic network contains 16 nodes and 20 edges. The number of vessels passing through each node in a certain unit time span (1 h/2 h/3 h) is counted to obtain the vessel traffic flow dataset.

D. Vessel Traffic Flow Prediction Results

1) Network Settings. In this subsection, the optimal STMGCN hyperparameters are selected through extensive comparative experiments. In STMGCN, the MSE is set as the loss function. Both the graph convolution kernel size $\mathcal{K}$ and temporal convolution kernel size $\mathcal{K}_t$ are set to 3 in the model. All tests use 12 hours as a historical time window, i.e., 12 observation data are utilized to estimate vessel traffic flow in the next 1 hour, 2 hours, and 3 hours. In our experiments, 80% of the traffic flow data is selected as the training set, and the remaining 20% is exploited as the test set.

Since the hyperparameters (i.e., initial learning rate, batch size) of a deep learning model have a significant impact on its performance, we compare the different parameters of network settings and select the model parameters with the optimal performance. It is worth mentioning that, despite progress made in recent years, selecting an optimal learning rate (LR) in STMGCN remains a difficult task. LR controls how much the network parameters are adjusted to minimize the network loss function. If LR is too small, the deep learning model will hardly converge for a specific model, resulting in a protracted training period and lower performance. However, if LR is too high, it will lead to loss vanishing so that valuable features cannot be excavated. In this paper, we recommend manually experimenting with several LR values to find the optimal LR. As shown in Fig. 9, extensive tests have been carried out to explore the effect of various LR values on the proposed method training. The x-axis in Fig. 9 indicates the LR values, and the y-axis indicates the change in different metrics. As shown in Fig. 9 (a) and (b), if the LR value is equal to 0.0025, it can obtain the optimal prediction results on both the test and training datasets. Furthermore, if the learning rate is greater than 0.0025, the R^2 decreases with increasing learning rate, MAE and RMSE increase with increasing learning rate. Thus, the initial LR is set to 0.0025. In addition, we choose the batch size from $[3, 6, 8, 12, 24, 36, 48, 60]$ and analyze the results in prediction precision. Fig. 10 shows that when the batch size is 8, the prediction precision error is lowest, and the precision is highest in both the training and test sets. Therefore, the batch size is set to 8 in all experiments.

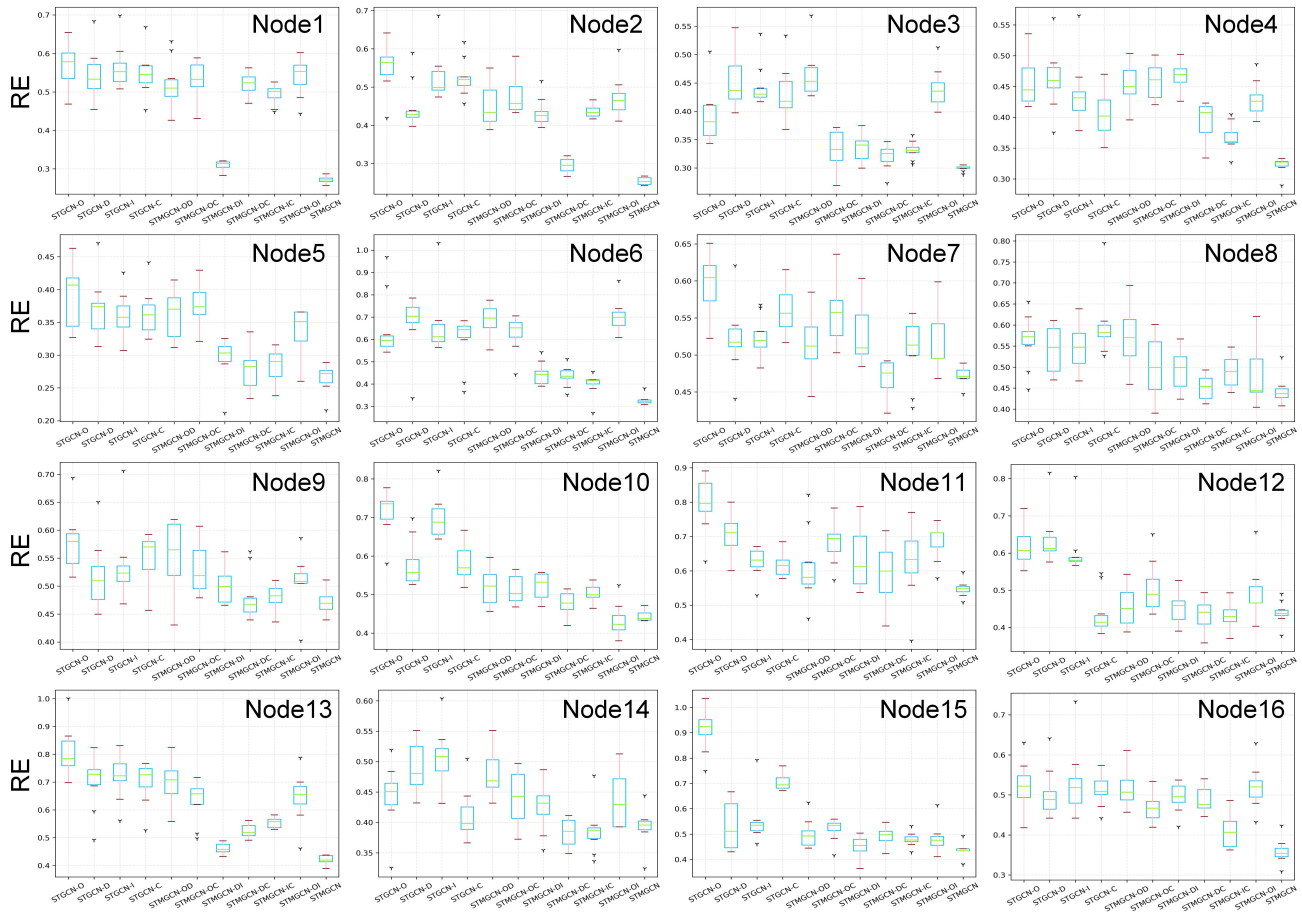


Fig. 12. Comparisons of RE for variants of STMGCN at 16 nodes. Each prediction method is performed ten times to decrease randomness.

2) Experiment Results. Table II demonstrates the prediction results of our STMGCN and other state-of-the-art methods for 1 hour, 2 hours, and 3 hours prediction tasks. We employ $*$ to represent negligibly small values and mark the best results in bold. As shown in Table II, STMGCN generates the most satisfactory prediction results for all evaluation metrics. It proves that STMGCN is effective for spatio-temporal vessel traffic flow prediction. In addition, it can be observed that the STMGCN obtains a minimum standard deviation under all evaluation indexes. It demonstrates that STMGCN has strong learning and stability properties, leading to superior prediction results. To further evaluate the accuracy and robustness, RE is employed to evaluate the proposed method and other competing methods for all graph nodes, shown in Table III. It can be seen from the Table III that the prediction results obtained by traditional methods are worse than learning models. Our graph-driven learning method has the capacity of yielding the best prediction results for all nodes in the maritime traffic network.

3) Ablation Experiments. In this subsection, we attempt to evaluate the effectiveness of graphs in STMGCN. The prediction results of single-graph (i.e., origin, distance, interaction or correlation graph) and multi-graph fusion variants are shown in Figs. 11 and 12. As observed in Fig. 11, the two-graph fusion-based variants obtain better prediction results than single-graph models. It is worth noting that STGCN-DC and STGCN-IC show better prediction results in two-graph fusion-based variants, which illustrates that the distance, interaction and correlation graphs can enhance the prediction accuracy. Owing to the superior performance of multi-graph fusion, our STMGCN performs well in predicting high-accuracy vessel traffic flow over different unit time spans. To further evaluate the prediction methods, we analyze the prediction results for all graph nodes. These prediction results are evaluated using the RE, creating the boxplots shown in Fig. 12. As observed from Fig. 12, STMGCN generates the most accurate and robust prediction results for all graph nodes.

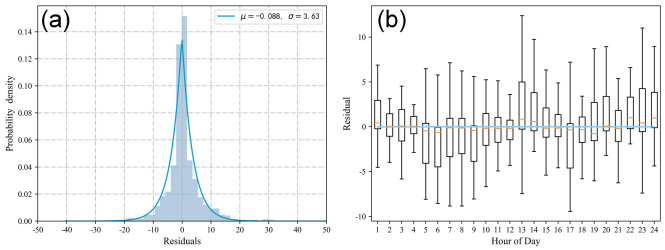


Fig. 13. Residual analysis. (a) Residual distribution of all predicted results. (b) Boxplots of residuals at different time periods.

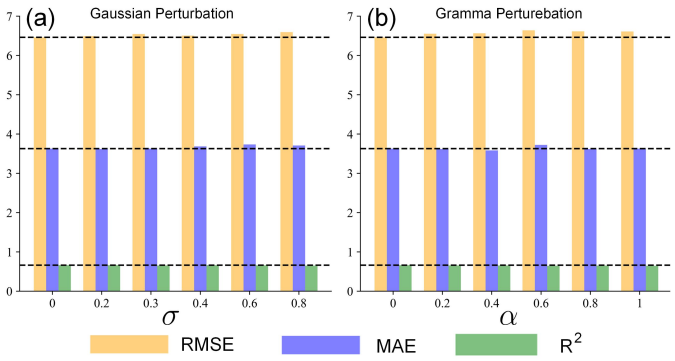


Fig. 14. Perturbation analysis. (a) Adding Gaussian perturbation to the input data. (b) Adding Poisson perturbation to the input data.

4) Residual Analysis. The residuals are an important indicator of whether a model is systematically correct, therefore this subsection analyzes the residuals of the proposed prediction method. The residual r can be defined as the ground-truth value subtracting the predicted value—that is, $r = x - \hat{x}$. The fitted Laplace distribution with a location parameter of -0.088 (i.e., a mean μ of -0.088) and scale parameter of 3.63 (i.e., variance σ of 3.63) is shown in Fig. 13 (a). The residual distributions with a mean of -0.088 indicate that the trained STMGCN has successfully learned the spatio-temporal correlations of vessel traffic flow in the maritime traffic network. The temporal information, such as the time of day, may influence the accuracy of traffic flow prediction. As shown in Fig. 13 (b), we estimate the residuals with regard to hour by drawing a box plot. The residuals, shown in Fig. 13 (b), are distributed around the zero lines, illustrating that the proposed method can fit the vessel traffic flow data and obtain good prediction performance in all-day.

5) Perturbation Analysis and Robustness. In the real world, noise is unavoidable during data collection and transmission. We investigate the robustness of STMGCN using perturbation analysis to test its noise immunity. Gaussian and Gamma noise can accurately reflect many systems, and are mathematically easy to deal with [63], [64]. Therefore, this work uses the Gaussian and Gamma noise to evaluate the robustness of our prediction method. As shown in Fig. 14, random noise with the Gaussian distribution $N \in (0, \sigma^2)$ and the Gamma distribution $G(\alpha, 0.5)$ are added to the input data. Fig. 14 (a) depicts the effects of adding Gaussian noise to the dataset, where the x-axis indicates σ, the y-axis indicates the change in each assessment measure, and colors indicate different metrics. Fig. 14 (b) depicts the effects of adding Gamma noise to the dataset. Regardless of noise distribution is Gaussian or Gamma, the change of metrics is extremely small in Fig. 14. It indicates that the STMGCN is a robust prediction method and can deal with high-noise problems.

VI. Conclusion and Future Perspectives

In this work, a graph-based learning framework was proposed for fine-grained vessel traffic flow prediction. To enhance the prediction accuracy, we first extract the regional maritime traffic networks from massive historical AIS data. The STMGCN was then constructed for traffic flow prediction, which has the advantage of considering multiple spatial correlations. Finally, comprehensive prediction experiments have been carried out on realistic dataset to demonstrate the superiority of the proposed method. Experimental results shown that the multi-graph fusion strategy can tremendously improve the prediction accuracy and robustness under different traffic conditions.

However, there still exists room for improvement. The traffic flow prediction performance is highly dependent on the selection of input parameters. Therefore, to guarantee more satisfactory prediction results, we will develop automatic methods to adaptively select the optimal parameters for both traffic network extraction and STMGCN-based traffic flow prediction. In addition, vessel traffic flow prediction is often influenced by abnormal events, e.g., severe weather, and traffic accidents, etc. It is thus necessary to further extend our STMGCN to learn the relationship between traffic flow prediction and abnormal events in practical applications.

References

[1] R. W. Liu, J. Chen, Z. Liu, Y. Li, Y. Liu, and J. Liu, "Vessel traffic flow separation-prediction using low-rank and sparse decomposition," in *Proc. ITSC*, 2017, pp. 516-521.

[2] P. R. Lei, "Mining maritime traffic conflict trajectories from a massive AIS data," *Knowl. Inf. Syst.*, vol. 62, no.1, pp. 259-285, Jan. 2020.

[3] R. Zhen, Y. Jin, Q. Hu, Z. Shao, and N. Nikitakos, "Maritime anomaly detection within coastal waters based on vessel trajectory clustering and Naïve Bayes classifier," *J. Navig.*, vol. 70, no. 3, pp. 648-670, Jan. 2017.

[4] M. Liang, Y. Zhan, and R. W. Liu, "MVFFNet: Multi-view feature fusion network for imbalanced ship classification," *Pattern Recognit. Lett.*, vol. 151, pp. 26-32 Nov. 2021.

[5] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6164-6173, Apr. 2009.

[6] Y. S. Jeong, Y. J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1700-1707, Dec. 2013.

[7] L. Chen, L. Zheng, J. Yang, D. Xia, and W. Liu, "Short-term traffic flow prediction: From the perspective of traffic flow decomposition," *Neurocomputing*, vol. 413, no. 6, pp. 444-456, Nov. 2020.

[8] C. Chen, Z. Liu, S. Wan, J. Luan, and Q. Pei, "Traffic flow prediction based on deep learning in internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3776-3789, Jun. 2021.

[9] B. Yu, Y. Lee, and K. Sohn, "Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN)," *Transp. Res. Part C Emerging Technol.*, vol. 114, pp. 189-204, May 2020.

[10] W. Zhang, F. Zhu, Y. Lv, C. Tan, W. Liu, X. Zhang, F.-Y Wang, "AdapGL: An adaptive graph learning algorithm for traffic prediction based on spatiotemporal neural networks," *Transp. Res. Part C Emerging Technol.*, vol. 139, p. 103659, Jun. 2022.

[11] S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan, "GSTNet: Global spatial-temporal network for traffic flow prediction, in *Proc. IJCAI*, 2019, pp. 2286-2293.

[12] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based Conv-LSTM networks for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6910-6920, Nov. 2021.

[13] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box Jenkins techniques," *Transp. Res. Rec.*, no. 722, pp. 1-9, Apr. 1979.

[14] Y. Yu, Y. Zhang, S. Qian, S. Wang, Y. Hu, and B. Yin, "A low rank dynamic mode decomposition model for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6547-6560, Oct. 2021.

[15] H. Grubb and A. Mason, "Long lead-time forecasting of UK air passengers by HoltWinters methods with damped trend," *Int. J. Forecasting*, vol. 17, no. 1, pp. 71-82, Mar. 2001.

[16] A. Stathopoulos and G. M. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transp. Res. Part C Emerging Technol.*, vol. 11, no. 2, pp. 121-135, Apr. 2003.

[17] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664-672, Nov. 2003.

[18] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 653-662, Apr. 2015.

[19] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. Part B Methodol.*, vol. 18, no. 1, pp. 1-11, Feb. 1984.

[20] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial-temporal correlation for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2001-2013, Jun. 2019.

[21] X. Chen, S. Wu, C. Shi, Y. Huang, Y. Yang, R. Ke, and J. Zhao, "Sensing data supported traffic flow prediction via denoising schemes and ANN: a comparison," *IEEE Sens. J.*, vol. 20, no. 23, pp. 14317-14328, Dec, 2020.

[22] M.-C. Tan, S. C. Wong, J.-M. Xu, Z. R. Guan, and Z. Peng, "An aggregation approach to short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 60-69, Mar. 2009.

[23] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297-305, Nov. 2018.

[24] L. Liu et al., "Dynamic spatial-temporal representation learning for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7169-7183, Nov. 2021.

[25] R. W. Liu, M. Liang, J. Nie, W. Y. B. Lim, Y. Zhang, and M. Guizani, "Deep learning-powered vessel trajectory prediction for improving smart-traffic services in maritime Internet of Things," *IEEE Trans. Net. Sci. Eng.*, to be published, 2022, doi: 10.1109/TNSE.2022.3140529.

[26] R. W. Liu, M. Liang, J. Nie, W. Y. B. Lim, Y. Zhang, and M. Guizani, "STMGCN: Mobile edge computing-empowered vessel trajectory prediction using spatio-temporal multi-graph convolutional network," *IEEE Trans. Ind. Inf.*, to be published, 2022, doi: 10.1109/TII.2022.3165886.

[27] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848-3858, Sep. 2020.

[28] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transp. Res. Part C Emerging Technol.*, vol. 90, pp. 166-180, May 2018.

[29] Y. Zhang, Y. Zhou, H. Lu, and H. Fujita, "Traffic network flow prediction using parallel training for deep convolutional neural networks on spark cloud," *IEEE Trans. Ind. Inf.*, vol. 16, no. 12, pp. 7369-7380, Dec. 2020.

[30] M. Liang, R. W. Liu, S. Li, Z. Xiao, X. Liu, and F. Lu, "An unsupervised learning method with convolutional auto-encoder for vessel trajectory similarity computation," *Ocean Eng.*, vol. 225, p. 108803, Apr. 2021.

[31] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865-873, Apr. 2015.

[32] Y. Li, R. Yu, C. Shahabi and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting", in *Proc. ICLR*, 2018, pp. 1-16.

[33] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. IJCAI*, 2018, pp. 3634-3640.

[34] D. Park and L. R. Rilett, "Forecasting freeway link travel times with a multilayer feedforward neural network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 14, no. 5, pp. 357-367, Dec. 2002.

[35] X. Ma, J. Zhang, B. Du, C. Ding, and L. Sun, "Parallel architecture of convolutional bi-directional lstm neural networks for network-wide metro ridership prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2278-2288, Jun. 2019.

[36] X. Geng, X. Wu, L. Zhang, Q. Yang, Y. Liu, and J. Ye, "Multi-modal graph interaction for multi-graph convolution network in urban spatio-temporal forecasting," 2019, *arXiv:1905.11395.* [Online]. Available: http://arXiv:1905.11395.

[37] A. K. Malik, R. Gao, M. A. Ganaie, M. Tanveer, P. N Suganthan, "Random vector functional link network: recent developments, applications, and future directions" 2022, *arXiv:2203.11316* [Online]. Available: http://arXiv:2203.11316.

[38] J. Zhou, J. X. Huang, Q. V. Hu, and L. He, "Sk-gcn: Modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification," *Knowledge Based Syst.*, vol. 205, p. 106292, Oct. 2020.

[39] X. Geng et al., "Spatio-temporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI*, 2019, pp. 1-8,

[40] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883-4894, Nov. 2020.

[41] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. ICLR*, Toulon, France, 2017, pp. 1-14.

[42] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. IJCAI*, 2018, pp. 3634-3640.

[43] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He. "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proc. AAAI*, 2019, pp. 890-897.

[44] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI*, 2019, pp. 922-929.

[45] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proc. AAAI*, 2020, pp. 914-921.

[46] Z. Li et al., "A multi-stream feature fusion approach for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1456-1466, Feb. 2022.

[47] W. He, C. Zhong, M. Sotelo, X. Chu, X. Liu, and Z. Li, "Short-term vessel traffic flow forecasting by using an improved Kalman model," *Cluster Comp.*, vol. 22, no. 10, pp. 1-10, Dec. 2019.

[48] J. Yu, G. Tang, X. Song, X. Yu, Y. Qi, D. Li, et al., "Ship arrival prediction and its value on daily container terminal operation," *Ocean Eng.*, vol. 157, pp. 73-86, Jun. 2018.

[49] Z.-G. Zhang, J.-C. Yin, N.-N. Wang, and Z.-G. Hui, "Vessel traffic flow analysis and prediction by an improved PSO-BP mechanism based on AIS data," *Evolving Syst.*, vol. 10, no. 3, pp. 397-407, Sept. 2018.

[50] S. Gan, S. Liang, K. Li, J. Deng, and T. Cheng, "Long-term ship speed prediction for intelligent traffic signaling," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 82-91, Jan. 2017.

[51] M.-W. Li, D.-F. Han, and W.-I. Wang, "Vessel traffic flow forecasting by RSVR with chaotic cloud simulated annealing genetic algorithm and KPCA," *Neurocomputing*, vol. 157, pp. 243-255, Jun. 2015.

[52] R. W. Liu, J. Nie, S. Garg, Z. Xiong, Y. Zhang, and M. S. Hossain, "Data-driven trajectory quality improvement for promoting intelligent vessel traffic services in 6G-enabled maritime IoT systems," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5374-5385, Apr. 2021.

[53] H. Li, J. Liu, Z. Yang, R.W. Liu, K. Wu, and Y. Wan, "Adaptively constrained dynamic time warping for time series classification and clustering," *Inf. Sci.*, vol.534, pp.97-116, Sep. 2020

[54] Y. Liu, W. Dong, D. Gong, L. Zhang, and Q. Shi, "Deblurring natural image using super-Gaussian fields," in *Proc. ECCV*, 2018, pp. 452-46.

[55] Y. Lu, Y. Guo, and M. Liang, "CNN-enabled visibility enhancement framework for vessel detection under haze environment," *J. Adv. Transp.*, vol.2021, p.5598390, May 2021.

[56] R. Gao, J. Liu, X. Bai, and K. F. Yuen, "Annual dilated convolution neural network for newbuilding ship prices forecasting," *Neural Comput. Appl.*, Mar. 2022.

[57] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "SocialSTGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. CVPR*, 2020, pp. 14412-14420.

[58] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," in *Proc. Conf. Empirical Methods Natural Lang.*, 2017, pp. 1506-1515.

[59] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for Web-scale recommender systems," in *Proc. SIGKDD*, 2018, pp. 974-983.

[60] M. Defferrard, X. Bresson, and P. Van der Gheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. NIPS*, 2016, pp. 3844-3852.

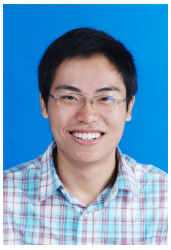
[61] G.Sofia, "Combining geomorphometry, feature extraction techniques and Earth-surface processes research: The way forward," *Geomorphology*, vol. 355, p. 107055, Apr. 2020.

[62] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, "Citywide cellular traffic prediction based on densely connected convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1656-1659, Aug. 2018.

[63] S. Nadarajah and S. Kotz, "On the generation of Gaussian noise," *IEEE Trans. Signal Process.*, vol. 55, no. 3, pp. 1172-1172, Mar. 2007

[64] J. Jiang, L. Zhang and J. Yang, "Mixed noise removal by weighted encoding with sparse nonlocal regularization", *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2651-2662, Apr. 2014.

Maohan Liang received his master degree in Traffic and Transportation Engineering from the Wuhan University of Technology, Wuhan, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Navigation, Wuhan University of Technology, Wuhan, China, and studying as an exchange student with the Nanyang Technological University, Singapore. His research interests include trajectory data mining, deep learning, and intelligent transportation systems.



Ryan Wen Liu (M'15) received the B.Sc. degree (Hons.) in Information and Computing Science from the Department of Mathematics, Wuhan University of Technology, Wuhan, China, in 2009, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2015. He is currently an Associate Professor with the School of Navigation, Wuhan University of Technology. He was a Visiting Scholar with the Agency for Science, Technology and Research, Singapore. His research interests include computer vision data mining, and intelligent transportation system.

Yang Zhan received the B.Sc. degree in Computer Science and Technology from the Wuhan University of Technology, Wuhan, China, in 2022. He is currently pursuing Ph.D. degree with the School of Artificial Intelligence, OPtics, and ElectroNics (iOPEN), Northwestern Polytechnical University, Xian, China. His research interests include deep learning, data mining, and multimodal machine learning.

Huanhuan Li received the M.Sc. degree in Applied Mathematics from the School of Science in 2015 and the Ph.D. degree in Traffic Information Engineering and Control from the School of Navigation in 2020, Wuhan University of Technology, Wuhan, China. She was a research fellow at Nanyang Technological University in Singapore from 2020 to 2021. She is currently a research fellow with the School of Engineering in Liverpool John Moores University. Her research interests include maritime big data mining and maritime transportation safety.

Fenghua Zhu received the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2008. He is currently an Associate Professor with the State Key Laboratory for Management and Control of Complex Systems, China. His research interests are artificial transportation systems and parallel transportation management systems.

Fei-Yue Wang (S'87-M'89-SM'94-F'03) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York in 1990. He joined the University of Arizona in 1990 and became a professor and Director of the Robotics and Automation Laboratory (RAL) and Program in Advanced Research for Complex Systems (PARCS). In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Oversea Chinese Talents Program from the State Planning Council and "100 Talent Program" from CAS, and in 2002, was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory of Management and Control for Complex Systems. Dr. Wang's current research focuses on methods and applications for parallel systems, social computing, and knowledge automation. He was the Founding Editor-in-Chief of the International Journal of Intelligent Control and Systems (1995−2000), Founding EiC of IEEE ITS Magazine (2006−2007), EiC of IEEE Intelligent Systems (2009−2012), and EiC of IEEE Transactions on ITS (2009−2016). Currently he is EiC of China's Journal of Command and Control. Since 1997, he has served as General or Program Chair of more than 20 IEEE, INFORMS, ACM, ASME conferences. He was the President of IEEE ITS Society (2005−2007), Chinese Association for Science and Technology (CAST, USA) in 2005, the American Zhu Kezhen Education Foundation (2007−2008), and the Vice President of the ACM China Council (2010−2011). Since 2008, he is the Vice President and Secretary General of Chinese Association of Automation. Dr. Wang is elected Fellow of IEEE, INCOSE, IFAC, ASME, and AAAS. In 2007, he received the 2nd Class National Prize in Natural Sciences of China and awarded the Outstanding Scientist by ACM for his work in intelligent control and social computing. He received IEEE ITS Outstanding Application and Research Awards in 2009 and 2011, and IEEE SMC Norbert Wiener Award in 2014.