

Full length article

A deep spatio-temporal meta-learning model for urban traffic revitalization index prediction in the COVID-19 pandemic

Yue Wang^a, Zhiqiang Lv^{a,b}, Zhaoyu Sheng^c, Haokai Sun^c, Aite Zhao^{a,*}^a College of Computer Science and Technology, Qingdao University, Qingdao, China^b Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China^c Institute of Ubiquitous Networks and Urban Computing, Qingdao, China

ARTICLE INFO

Keywords:

Urban computing

Traffic revitalization index prediction

COVID-19 pandemic

Meta-learning

Spatio-temporal correlation

ABSTRACT

The COVID-19 pandemic is a major global public health problem that has caused hardship to people's normal production and life. Predicting the traffic revitalization index can provide references for city managers to formulate policies related to traffic and epidemic prevention. Previous methods have struggled to capture the complex and diverse dynamic spatio-temporal correlations during the COVID-19 pandemic. Therefore, we propose a deep spatio-temporal meta-learning model for the prediction of traffic revitalization index (DeepMeta-TRI) using external auxiliary information such as COVID-19 data. We conduct extensive experiments on a real-world dataset, and the results validate the predictive performance of DeepMeta-TRI and its effectiveness in addressing underfitting.

1. Introduction

The COVID-19 outbreak has brought about major changes in urban functions and people's normal production and life, with significant negative impacts on the economy and transportation [1]. During the Spring Festival 2020, the railway, road, waterway, and civil aviation nationwide handled 1.48 billion passenger trips, down 50.3% from the same period in 2019 [2]. The urban traffic revitalization index (TRI), which is close to 1 under normal conditions at the end of 2019, plummeted as a result of the direct impact of the COVID-19 pandemic. TRI falls within the ambit of traffic vitality in urban vitality, which is a key indicator of the health and orderliness of urban functions, production, and life, proposed by Didi in cooperation with the National Engineering Laboratory of Big Data Analysis and Application Technology of Peking University and the China Center for Information Industry Development. In [3], the urban vitality index was evaluated in five aspects: transportation, society, commerce, tourism, and culture-education. [4] proposed a model to evaluate the social vitality recovery level based on taxi travel data. [5] described the average weekly recovery rate of urban traffic flow during the COVID-19 pandemic. [6] studied the recovery of urban transportation systems from massive events. The above studies illustrate the importance of TRI. With the anti-epidemic measures implemented by the government and the recovery of work in the post-epidemic period, urban production and life have gradually recovered. Coupled with the continuous improvement of the transportation hub system that plays a role in promoting transportation

development, TRI shows a stepwise increase, as shown in Fig. 1, and TRI is proportional to the activity of urban traffic activities.

Gross Domestic Product (GDP) is one of the most important metrics to measure the development of the national economy, which is affected by the development of infrastructure. As a carrier of the national economy, urban transportation plays an extremely vital role in economic development. Fig. 2 shows the GDP and the average TRI for the first and second quarters, from which we can see that with the improvement of the domestic epidemic prevention and control situation, TRI in the second quarter is higher than that in the first quarter, and because GDP is affected by the traffic vitality, the GDP in the second quarter is higher than that in the first quarter. There is a dynamic synchronization between TRI and GDP.

With the alleviation of the COVID-19 pandemic, economic recovery in the post-epidemic period has become one of the government's top priorities. The recovery of urban transportation is crucial in the process of stable economic activities. To perceive the recovery and states of urban traffic, the analysis and prediction of TRI are indispensable. The prediction of TRI can be deployed and applied to transportation-related platforms to present real-time and predicted data to the public scientifically, objectively, and intuitively, as well as indirectly reflect the economic development and the trend of the COVID-19 epidemic in the city. TRI can also be provided as data support and reference for local governments to further formulate and adjust policies related to urban traffic management and epidemic prevention.

* Corresponding author.

E-mail address: zhaoaite@qdu.edu.cn (A. Zhao).

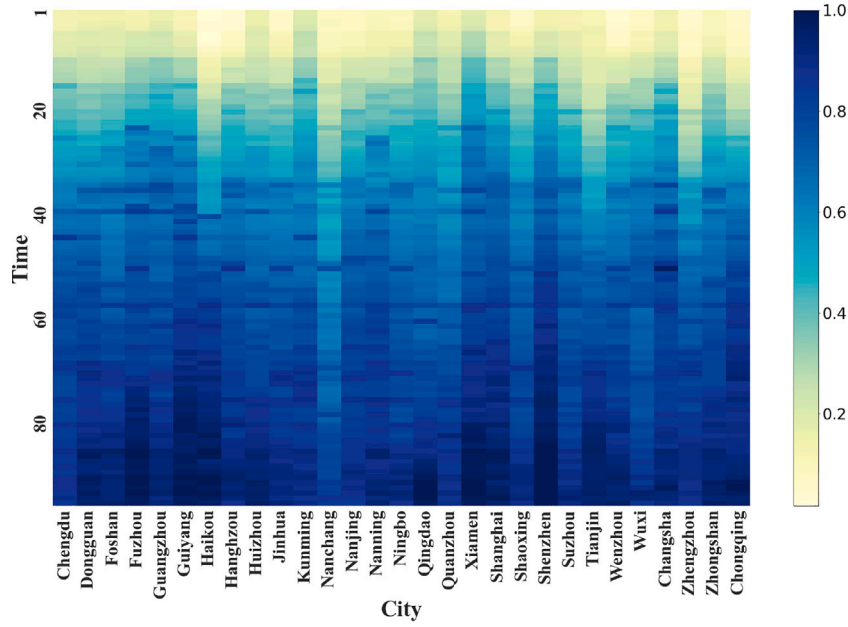


Fig. 1. Visualization of traffic revitalization index trends.

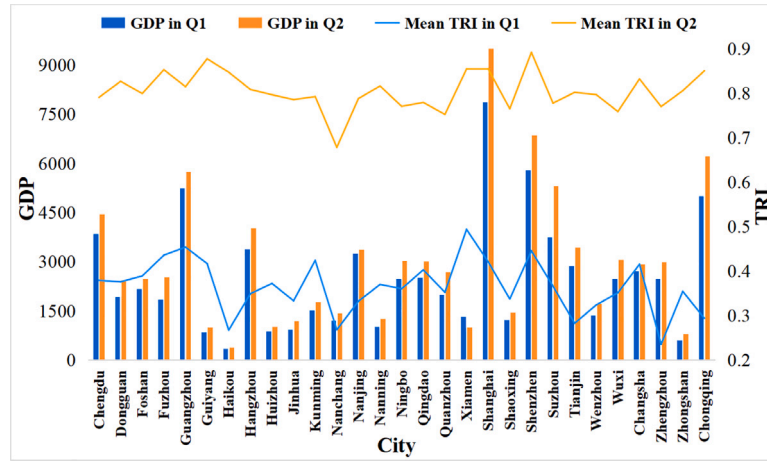


Fig. 2. Trend of GDP and TRI.

Traffic revitalization index forecasting falls within the field of traffic forecasting [7–10], which has been extensively researched and developed in the past decades. Assuming the traffic is static, the existing time series models, such as linear regression [11] and autoregressive integrated moving average models (ARIMA [12]), are based on linear models to capture the temporal dependence. However, real-world traffic states exhibit complex nonlinear relationships and are typically non-static in terms of geographic location, travel time, and weather. Therefore, they cannot model complex nonlinear spatio-temporal relationships. Machine learning models, such as support vector regression, k-nearest neighbor, and artificial neural networks (ANN) [13], build empirical functions in a data-driven manner and more accurately characterize nonlinear spatio-temporal correlations. Nevertheless, they are not efficient in mining deep spatio-temporal correlations in massive traffic data and still cannot capture sophisticated nonlinear spatio-temporal dependencies effectively. Research and development in deep learning [14–17] have enabled the modeling of complex spatio-temporal dependencies, and applications of deep learning networks to traffic prediction have been deployed in real-world cities [18,19]. Recurrent neural networks (RNNs) have been widely used for various

sequence prediction tasks. As variants of RNNs, long short-term memory (LSTM) and gated recurrent unit (GRU) are capable of capturing long-term nonlinear temporal dependencies. To model spatial correlations, researchers applied convolutional neural networks (CNNs) [20] to capture dependencies in Euclidean space. Models incorporating the above methods [21] can learn both spatial and temporal dependencies. However, CNNs are limited to dealing with regular grid structures (e.g., images and videos) and do not consider non-Euclidean correlations. In a bid to more precisely capture the spatial features of irregular traffic networks, graph convolutional networks (GCNs) [22] have been used to model topological features in non-Euclidean structures. The research on meta-learning [23] (e.g., metric-based meta-learning and optimization-based meta-learning), has become an essential driving force to enhance the overall effectiveness of deep learning.

The dynamics changes of spatio-temporal correlations during COVID-19 lead to the inability of existing models to effectively capture the potential relationships between different spatial locations and times, while the integration of external auxiliary information is required. We propose a deep spatio-temporal meta-learning model for traffic revitalization index prediction (DeepMeta-TRI), which copes with the complex and diverse dynamic spatio-temporal correlations

under the COVID-19 pandemic. DeepMeta-TRI consists of a temporal convolution module (TCM), a meta graph convolution network module (MetaGCN), and a meta temporal convolution module (MetaTCM). The predictive performance is further improved mainly by fusing TRI context and external auxiliary information (e.g., COVID-19 data, traffic hubness, weather, etc.) through a multi-scale post-fusion approach and meta-learning based on weight generation. The main contributions of our research are as follows:

- We propose a new deep spatio-temporal meta-learning network. To the best of our knowledge, this is the first one to integrate meta-learning as well as external auxiliary information such as the COVID-19 pandemic and urban traffic hubness, to predict urban traffic revitalization index.
- To model the complex and diverse spatio-temporal correlations, we design a meta graph convolution network module and a meta temporal convolution module. The feature extraction from external auxiliary information with TRI context is processed by the meta gating fusion module and then the parameter weights of the spatial and temporal modules are generated by the meta learner, respectively.
- We evaluate several baseline methods and DeepMeta-TRI using a real-world statistically derived urban traffic revitalization index dataset and conduct ablation experiments to verify the effectiveness of each split internal module. The experimental results show that our model has significant strengths in metrics and tackling underfitting.

2. Related work

2.1. Modeling methods

Traditional forecasting models are usually modeled with the help of statistical analysis. The seasonal autoregressive integrated moving average model [24] considers the relationship between historical and current data for forecasting based on the periodicity and trend of the data. The Kalman filter model [25] uses state equations to derive predictions. Machine learning models, such as artificial neural networks and support vector regression [13], obtain nonlinear changes in the data by back propagation (or radial basis function) and features of historical data, respectively. However, the above methods are difficult to handle large amounts of data and model complex nonlinear spatio-temporal relationships.

Deep learning-based models [26–28] have been widely applied to different spatio-temporal forecasting tasks. RNNs have been successfully employed for sequence learning tasks, and their variants LSTM [29] and GRU [30] are able to model long-term temporal dependence. The crowd flow is predicted by convolution-based residual networks in [31]. [32] uses traffic networks as images for traffic speed prediction based on CNN. However, a single network cannot efficiently and explicitly model spatio-temporal correlation. [33] combines CNN and RNN to capture spatio-temporal dependencies. [34] employs attention-weighted traffic flow features to learn dynamic spatio-temporal representations and explores the role of external factors, such as weather and holiday information. STDN [35] embeds multiple external information (e.g., weather, events, etc.) in the spatio-temporal dimensions and uses the dynamic similarity between locations and periodic attention shifting mechanisms to predict taxi demand at the regional level.

However, unlike the spatially localized structured data, graph data has an arbitrary range and complex topology. CNNs are suitable for processing grid-like regular data, ignoring the topology of the traffic network, therefore, failing to characterize the spatial dependence of traffic in essence. Graph Convolutional Networks (GCNs) [22] can model non-Euclidean spatially structured data and extract spatial features by aggregating the features of neighbors in the traffic graph. [36]

incorporates GCN and RNN-based models to capture spatio-temporal dependencies and further improve prediction accuracy. [37] combines a variety of external factors such as metro stations, bus stop information, etc., enabling the integration of spatio-temporal characteristics of knowledge and data. T-GCN [38] integrates GCN and GRU to present steady-state prediction results in different time intervals for long-term traffic prediction tasks. A3TGCN [39] introduces an attention mechanism, GCN and GRU to capture global changes. The above methods have a large number of parameters due to the introduction of RNN, which leads to a decrease in computational performance. A model composed entirely of convolution can greatly improve computational performance. [40] proposes a model that incorporates GCN and is able to compute long-range dependencies. STGCN [41] combines GCN with Conv1D based on sensor data to simulate spatio-temporal correlation for traffic prediction. ASTGCN [42] includes three temporal attributes, i.e., recent, daily-periodic, and weekly-periodic dependencies, using temporal and spatial attention mechanisms, GCN, and Conv1D to capture spatio-temporal dynamics. The three models mentioned above, although composed entirely of convolution, do not take into account that the traffic state is influenced by complex and diverse external information.

2.2. Meta-learning

The purpose of the weight-based meta-learning approach is to learn and initialize valid weight parameters of the network. In the literature [43], a weight function is learned implicitly in the gradient update to adjust the weights of the samples. [44] proposes a method for learning the parameters of a deep model in one shot, which can generalize a complete deep discriminative model from a single supervised example to identify other instances of the same object class. In [45], a weighting function is explicitly learned to adjust the weights of the samples in the gradient update, while optimizing the noise label and category imbalance. [46] performs data augmentation and weighting by learning a data manipulation method. [47] is a method that uses a supernet to generate weights for another network and is often adopted for compression or multitask learning. [48] provides a meta-multitask learning sequence modeling approach that utilizes a shared meta-network to capture the meta-knowledge of semantic combinations. [49] proposes a small-sample learning algorithm based on weight generation, using a generator to generate attention weight parameters for classifiers.

There are also a couple of meta-learning studies related to the graph structure. [50] employs local subgraphs for meta-learning. [51] can aggregate graph-level information by directly learning graph representations and generate weights directly by running inference on a graph neural network to amortize the search cost. [52] uses a message-passing-based approach to pass labeled supported samples to unlabeled samples via graph inference. [53] is applied to capture semantic higher-order relations and uses an attention mechanism to learn the personalized meta-graph weights for each node.

Distinct from the approaches mentioned above, we employ a combined spatio-temporal model to capture the nonlinear spatio-temporal dependencies in traffic, which is capable of processing non-Euclidean data, overcoming problems such as the slow training speed of traditional recurrent neural networks, and capturing long-term historical information. Since external factors have complex effects on TRI, we integrate external auxiliary information in DeepMeta-TRI. To leverage the value of external auxiliary information, we first analyze the external data and pre-process them, and then use a multi-scale post-fusion approach instead of directly integrating the weights extracted from the external auxiliary information with TRI. The multi-scale post-fusion approach is utilized to fuse the external auxiliary information with the TRI context and further feed them into the meta learner based on the weight generation to learn the parameter weights of the network, which is more efficient for the integration of semantic features at different levels. A unique problem with the urban traffic revitalization index during COVID-19 has finally fully lived up to its predictions.

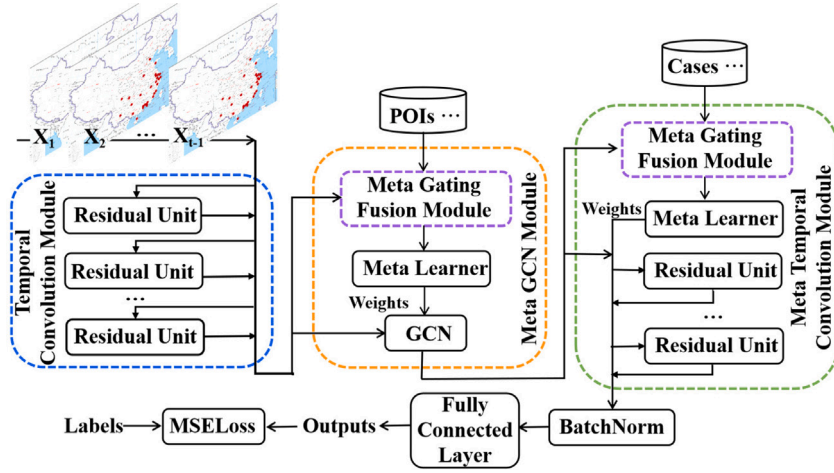


Fig. 3. DeepMeta-TRI framework.

3. Methodologies

3.1. Problem statement

The task of this work is to learn a function $F\Theta(\cdot)$ that utilizes historical data of the TRI X_{in} , external auxiliary information E_x and graph structure G to predict the TRI X_{out} at time t , as shown in Eq. (1).

$$X_{out} = F\Theta(X_{in}, E_x, G) \quad (1)$$

Definition 1 (Input and Output). $X_{in} = (X_{t-\tau}, X_{t-\tau+1}, \dots, X_{t-1}) \in R^{N \times T}$ is the set of input data consisting of recent consecutive periods. $X_{out} = X_t \in R^{N \times T}$ is the output data, where N is the number of nodes and T is the number of time steps required for prediction.

Definition 2 (External Auxiliary Information). E_x is the set of external auxiliary information, including the similarity of Points of Interest (POIs) $P_{i,j} \in R^{N \times N}$ at spatial locations i and j , the COVID-19 data $e_t^c \in R^{N \times T \times M}$, urban traffic hubness $e_t^{hubs} \in R^{N \times T}$, weather conditions $e_t^w \in R^{N \times T}$, temperature $e_t^{tem} \in R^{N \times T}$ and holiday $e_t^h \in R^{N \times T}$ at time t . M is the category of the COVID-19 data.

Definition 3 (Transportation Network). G is an undirected graph representing the topological relationship between city regions, denoted as $G = \{V, E, A\}$. $V = \{v_1, \dots, v_N\}$ is the set of nodes of this undirected graph. $E = \{e_{i,j} \mid 1 < i, j < N\}$ is the set of edges between nodes in the undirected graph, and A is the adjacency matrix $A_D \in R^{N \times N}$ of the distance-based undirected graph.

3.2. Method design

3.2.1. Framework overview

Fig. 3 illustrates the overall structure of DeepMeta-TRI. To begin with, we use a temporal convolution module (TCM) to extract the long-term temporal dependence of TRI and low-level abstract features. In addition, we design a meta graph convolution module (MetaGCN) to model diverse spatial correlations. The external static auxiliary information and the output of TCM are fused in the MetaGCN's meta gating fusion module and fed to the meta learner. The parameter weights of the network generated by the meta learner are fed into the GCN along with the output of the TCM. Then, a meta temporal convolution module (MetaTCM) is designed to model dynamic and diverse temporal correlations. The external dynamic auxiliary information and the output of MetaGCN are fused in the MetaTCM's meta gating fusion module and fed to the meta learner. The parameter weights of the network

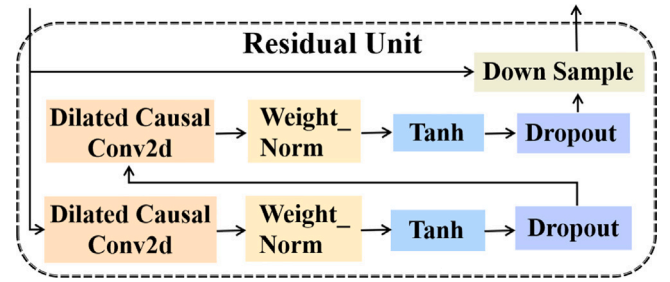


Fig. 4. Structure of the residual unit.

generated by the meta learner are fed to the residual unit jointly with the output of the MetaGCN. Furthermore, the output of MetaTCM is normalized using BatchNorm to avoid the gradient explosion and gradient disappearance caused by a network composed entirely of convolution and to speed up the convergence of the network. Finally, the results are output by the fully connected layer.

3.2.2. Temporal convolution module

TCM is utilized to extract the low-level features of the TRI, which consists of several residual units as shown in Fig. 4.

Causal Convolution Since this module deals with sequence modeling to ensure that historical data are not missed, we employ causal convolution [54]. The causal convolution focuses only on the historical information and is a one-way time-constrained structure, which allows the TRI features of the current node in the hidden layer to be associated with all the historical information, enhancing the temporal dependence of the sequence. The causal convolution at x_t is computed in Eq. (2),

$$(F * X)_{(x_t)} = \sum_{i=1}^K f_i \cdot x_{t-K+i} \quad (2)$$

where $F = \{f_1, f_2, \dots, f_K\}$ is the filter and $X = \{x_1, x_2, \dots, x_T\}$ is the sequence. However, the modeling length of causal convolution over time is limited by the size of the convolution kernel, and to obtain more distant dependencies, more convolution layers need to be stacked, which raises problems such as training complexity and gradient disappearance.

Dilated Convolution In deep networks, to ensure a lower computational effort while expanding the perceptual field, we adopt dilated convolution [55]. The filter can be applied to regions larger than the length of the filter itself by skipping some of the inputs (interval sampling). The interval size is controlled by the $dilationrate = O(2^i)$,

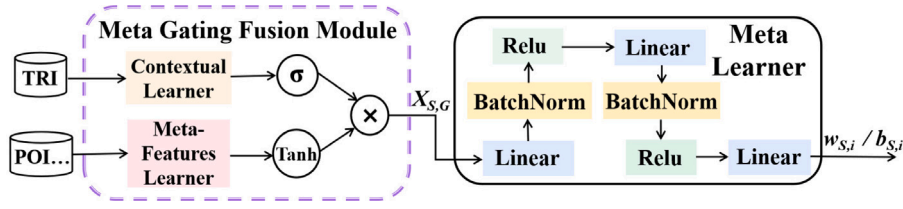


Fig. 5. Structure of meta gating fusion module and meta learner.

which allows the receptive field to grow exponentially as the network deepens, without introducing additional parameters. Each layer in the dilated convolution is padded with $padding = (k-1)d$, $receptive\ field = (k-1)d + 1$, k being the convolution kernel size, and d being the dilatation rate. Increasing d or k enlarges the receptive field. For the data that is greatly affected by time series like traffic vitality, dilated convolution can extract the temporal features of traffic vitality more completely. The formula of the dilated convolution at a dilatation rate equals to d is shown in Eq. (3).

$$(F *_{d} X)_{(x_i)} = \sum_{i=0}^{K-1} f_i \cdot x_{i-d-i} \quad (3)$$

The deeper the network the more abstract and semantically informative the features extracted, but simply increasing the depth would lead to gradient disappearance or gradient explosion. Therefore, we apply residual connection, which allows historical data to be used effectively over time.

3.2.3. Meta graph convolutional network module

As the TRI is affected by the spatial location, we propose the meta graph convolution network module (MetaGCN), which consists of a meta gating fusion module, a meta learner, and the GCN, where the meta fusion gating module and the meta learner are shown in Fig. 5.

Meta Fusion Gating Module This module as a component of MetaGCN is mainly used to provide the relationship between contextual information about TRI and external auxiliary information to improve the predictive performance. Numerous previous works [41,42] assume that the traffic conditions in one area are influenced by the traffic conditions in nearby areas. However, spatial correlation does not depend entirely on the distance between geographical locations. If two regions far from each other have comparable POI distributions, they also have similar patterns of trend changes. POIs are points of information or points of interest in geometric information systems and map services. POIs have been shown to have a strong correlation with traffic travel [56]. We classify POIs by representative 23 categories, obtain the number of each type of POI in a region and form a vector whose dimension is the number of categories of POIs. Then the similarity $P_{i,j}$ of POIs between regions is calculated by using the Pearson correlation coefficient, which is defined as follows:

$$P_{i,j} = \text{Pearson}(P_i, P_j) \in [0, 1] \quad (4)$$

where P_j and P_i are the POI vectors of regions i and j , respectively.

Instead of directly integrating the weights generated from external auxiliary information with TRI, we obtain the weights after a multi-scale post-fusion approach, i.e., fusing the separately processed TRI and external auxiliary information before extracting the parameter weights of the network, which is more conducive to the fusion of semantic features at different levels. Since temporal correlation is time-varying, we use Conv2d to learn the contextual information of the data after the processing of TCM. The fully connected network is used as a meta-features learner to extract external auxiliary information as meta-features of the metadata to learn its impact on each region. The expression of the MetaGCN's meta gating fusion module is as follows,

$$X_{S,G} = \text{Tanh}(FCN(P_{i,j})) \odot \sigma(\text{Conv2d}(TRI)) \quad (5)$$

where $X_{S,G}$ is the output of the meta gating fusion module of MetaGCN, $\text{Tanh}(\cdot)$ is the hyperbolic tangent function that ensures that the output value is between -1 and 1 , $FCN(\cdot)$ is the fully connected network that acts as a meta-features learner to learn the features of $P_{i,j}$. \odot is the element-wise product, and $\sigma(\cdot)$ is the sigmoid function that determines the proportion of information passed to the next layer.

Meta Learner The meta learner consists of a three-layer fully connected network, which includes nonlinear, batch processing, and activation functions. The output of the meta gating fusion module is input to two weight-based generation meta learners to generate the parameter weights for the network. The output feature of the meta learners with parameter weights are fed into the GCN and further combined with the TCM to capture different spatial correlations. The parameter weights are calculated as shown in Eq. (6),

$$w_{S,i}, b_{S,i} = g_{S,w}(X_{S,G}), g_{S,b}(X_{S,G}) \quad (6)$$

where $X_{S,G}$ is the output of the meta gating fusion module. $w_{S,i}$ and $b_{S,i}$ are the parameter weights of the network learned by meta learners $g_{S,w}$ and $g_{S,b}$, respectively.

Graph Convolutional Network Since regions (nodes) are connected in the form of topological graphs and spatial features are mainly manifested in the existence of interactions between different nodes, we use GCN for information transfer at the region level. GCN has the ability to handle highly nonlinear data in non-Euclidean spaces and can effectively extract complex spatial correlations.

GCN learns a function by mapping nodes v_i in the graph. v_i is influenced by adjacent or distant nodes by aggregating their features x_i and the features x_j of neighboring nodes to continuously generate new representations of v_i until equilibrium is reached. GCN operation mainly takes advantage of the fact that the Laplacian matrix can perform a feature decomposition of the graph information. To incorporate the influence of nodes on themselves in the computation, we employ an advanced version of the Laplacian matrix in Eq. (7),

$$L = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \quad (7)$$

where $\tilde{A} = A + I_n$ denotes the distance-based adjacency matrix after adding the self-join and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ represents the degree distribution of the nodes of A .

For large-scale graph structures, however, the eigendecomposition of Laplacian matrices is inefficient [22]. Therefore, we approximate the Laplacian matrix with K -order Chebyshev polynomials to reduce the time complexity. Meanwhile, it implicitly avoids the computation of the graph Fourier basis and ensures that the current node only considers the influence of nodes in the range K on itself. The formula is shown in Eq. (8),

$$x * g(\theta) = \sum_{k=0}^K \theta_k T_k(\tilde{L})x \quad (8)$$

where x is the feature matrix of the output of the temporal convolution module. $T_k(\tilde{L}) = 2\tilde{L} \cdot T_{k-1}(\tilde{L}) - T_{k-2}(\tilde{L})$ indicates the recursive definition of the Chebyshev polynomial, $T_1(\tilde{L}) = \tilde{L}$, $T_0(\tilde{L}) = 1$, $\tilde{L} = \frac{2}{\lambda_{\max}} L - I_n$.

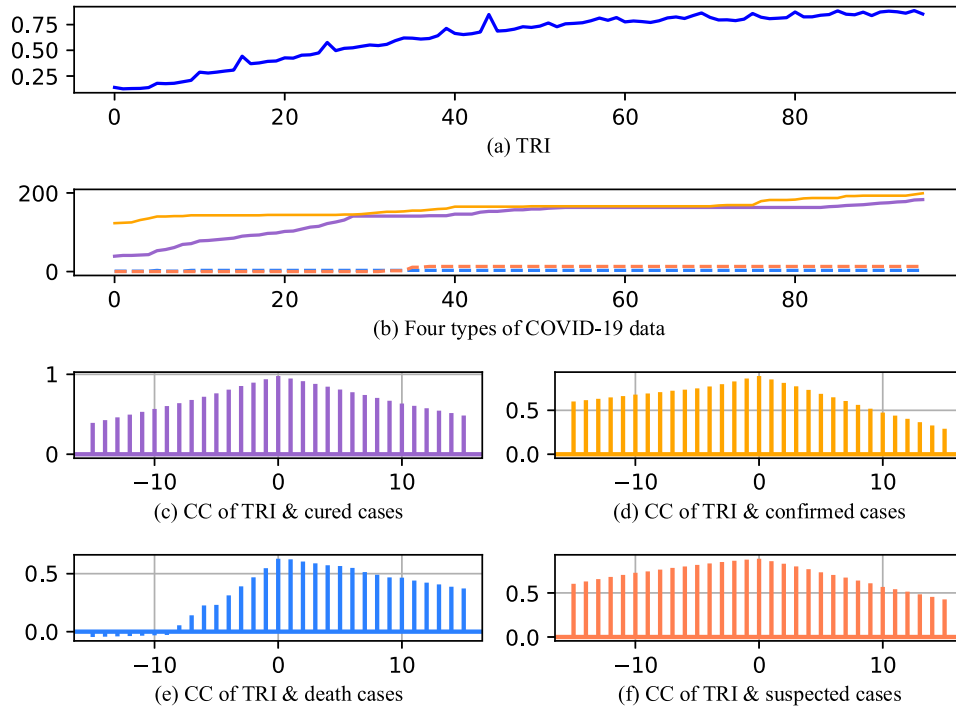


Fig. 6. Cross-correlation of TRI and COVID-19 data.

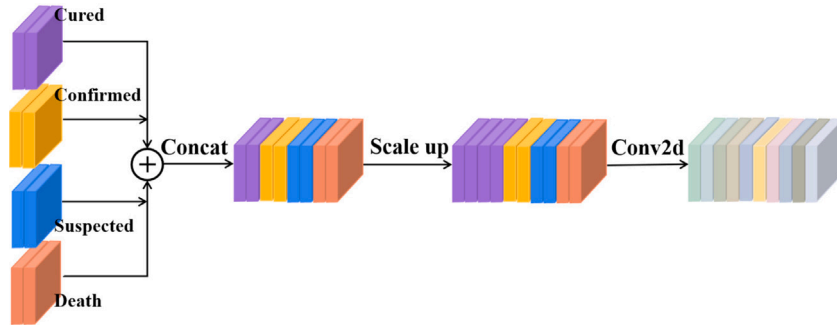


Fig. 7. Visualization of the steps for pre-processing COVID-19 data.

3.2.4. Meta temporal convolution module

For modeling complex and diverse temporal correlations and higher-level abstract features, the meta temporal convolution module (MetaTCM) adds a meta gating fusion module and the meta learner to the TCM, and their structures are similar to those mentioned in MetaGCN. The input of the MetaTCM's meta gating fusion module is the output of the MetaGCN with external dynamic auxiliary information such as COVID-19 data, traffic hubness, and weather conditions.

The emergence of the COVID-19 pandemic has seriously affected the traffic operations of various cities. The transportation capacity of many cities has declined, and the vitality of transportation has also been greatly weakened. Fig. 6 shows the cross-correlation (CC) between TRI and the four selected types of COVID-19 data (i.e., the number of cured cases, confirmed cases, deaths, and suspected cases), which represents the degree of correlation between the two temporal series. Figs. 6(a) and 6(b) show the visualization of the trend of the TRI and the four types of COVID-19 data over time, respectively. The X-axis in Figs. 6(c)–6(f) is the lag coefficient, the Y-axis corresponds to the degree of correlation between 6(a) and 6(b), and each vertical line indicates the correlation coefficient at the lag time. It is evident that the degree of correlation between both sets of data is significant, which further illustrates the importance of integrating the COVID-19 data.

As can be seen in Fig. 6, there is a significant correlation between the selected four types of COVID-19 data and TRI, and in particular, there is a stronger dynamic correlation between the number of cured cases and TRI in comparison. Therefore, the dimensionality of the channel is increased to indicate the number of cured cases after concatenating the four types of data, with a final ratio of 2:1:1:1. Then Conv2d with a kernel size of 3×1 is applied to extract their preliminary features, and Fig. 7 shows a more visualized operation.

Meta Gating Fusion Module & Meta Learner The external dynamic auxiliary information involved in the meta gating fusion module includes traffic hubness, weather conditions, temperatures, and holidays, in addition to the COVID-19 data. Traffic hubness promotes the organic connection of multiple modes of transportation within a certain area and contributes considerably to the growth of the city's TRI. Unusual weather conditions, for instance, heavy rainfall and snowstorms, affect people's travel characteristics and reduce traffic capacity, leading to a decrease in urban traffic vitality. The above dynamic information is first processed by the meta-features learner hierarchically to get the COVID-19 pandemic e^c , traffic hubness e^{hub} , weather e^w , temperature e^{tem} , and holiday e^h respectively before concatenating to obtain E_D in Eq. (9). Then E_D is fused with TRI context to get $X_{T,G}$, as shown in Eq. (10),

$$E_D = e^c \oplus e^{hub} \oplus e^w \oplus e^{tem} \oplus e^h \quad (9)$$

$$X_{T,G} = \text{Tanh}(FCN(E_D)) \odot \sigma(\text{Conv2d}(TRI)) \quad (10)$$

where $X_{T,G}$ is the output of the Meta Gating Fusion Module, $\text{Tanh}(\cdot)$ is the hyperbolic tangent function, $FCN(\cdot)$ is the fully connected network that acts as a meta-features learner to learn the features of E_D , \odot is the element-wise product, and σ is the sigmoid function.

The meta learner in MetaTCM consists of a three-layer fully connected network, similar to that in MetaGCN. The parameter weights generated by the meta learners and the output of MetaGCN are jointly fed into the residual unit to learn the multi-scale spatio-temporal features. The parameter weights are calculated as shown in Eq. (11),

$$w_{T,i}, b_{T,i} = g_{T,w}(X_{T,G}), g_{T,b}(X_{T,G}) \quad (11)$$

where $X_{T,G}$ is the output of the meta gating fusion module. $w_{T,i}$ and $b_{T,i}$ are the parameter weights of the network learned by meta learners $g_{T,w}$ and $g_{T,b}$, respectively.

4. Experiments

4.1. Experimental settings

4.1.1. Dataset

The urban traffic revitalization index (TRI) [57] in this experiment is based on the time range of February 10, 2020, to June 30, 2020, and the spatial range is 29 major cities in China. TRI is obtained by fitting, cross-validating, and weighting urban traffic trajectories, road congestion data, and commuting data from the Didi platform, which can reflect trends in traffic activity and urban recovery.

COVID-19 data include four types (the number of confirmed cases, suspected cases, cured cases, and deaths) for 29 cities with the same time range as TRI. The COVID-19 data are collected from the National Health Commission of the People's Republic of China [58].

1. Number of confirmed cases: Number of persons with clinical symptoms of COVID-19 and epidemiological history, as well as a positive nucleic acid test or confirmed by other laboratory test results.

2. Number of suspected cases: The number of persons diagnosed based on clinical symptoms and epidemiological history of COVID-19 and other actual circumstances.

3. Number of cured cases: Number of persons who had two consecutive negative nucleic acid tests with an interval of at least one day, as well as those who had significant improvement in respiratory symptoms, etc.

4. Number of deaths: The number of deaths due to clinically relevant disease among suspected or confirmed cases.

Data on urban transport hubness [59] are derived from a principal component analysis of the energy level of high-speed railway stations within cities, the number of highways passing through the national level, the number of cities directly accessible by road in 3 h, and the number of logistics stations. Intercity transportation infrastructure is the basic guarantee for urban transportation, and its construction efforts will undoubtedly affect the restoration of future transportation vitality.

POIs can accurately depict the distribution characteristics of city function points. We acquire POIs through the open API of Gaode Map, and there are 482,175 POIs, which are divided into 23 categories: food service, shopping service, sports and leisure service, accommodation service, scenic spot, science, education and culture service, company enterprise, etc. Weather conditions are classified into 14 categories, including heavy snow, heavy rain, thunderstorms, sunny, etc. We normalize the temperature data to the range $[-1, 0]$ with Min-Max normalization. Holiday data are labeled with weekdays and weekends.

4.1.2. Implementation details

In TCM and MetaTCM, five residual units are stacked with dilatation rates of 1, 2, 4, 8, and 16 to capture temporal features from different receptive fields, and the kernel size is 3×3 , padding is $2 \times$ dilatation rate, and dropout is 0.2. The contextual learner is the Conv2d with a kernel of 3×5 , the meta-features learner is a 32-dimensional FCN, and the meta learner consists of FCNs with dimensions of 16, 2, and z , respectively, where z is the target output dimension. K of Chebyshev polynomial in GCN is 3. Finally, a 3-dimensional fully connected layer maps the spatio-temporal features to obtain the final predictions.

We use the historical TRI at 12 time steps to predict the TRI at the last 3 time steps. The results are the average of 5 training times. The batch size is set to 5 and the initial learning rate adopted for training is 0.001. All models are trained by the Adam optimizer. In this experimental training process, we choose the commonly employed loss function L2, which enables the model to converge quickly, giving the gradient an appropriate penalty weight and making a more accurate gradient update direction. Our framework is implemented by the library of PyTorch 1.9.0. All methods run on a 6-core computer with a GPU of NVIDIA GeForce RTX 2080Ti and a CPU of Intel Xeon W-2133 3.6 GHz.

4.1.3. Baseline methods

We compare DeepMeta-TRI with the following state-of-the-art approaches.

- LSTM [29]: Long Short-Term Memory network is a special recurrent neural network that learns long-term dependent information.
- GRU [30]: Gated Recurrent Unit replaces forget gate and input gate in LSTM with update gate, which consumes less computational resources and improves computational efficiency.
- GCN [22]: Graph Convolution Network is a generalization of the convolutional neural network on the graph domain, which can perform end-to-end learning of both node feature information and structure information.
- STDN [35]: Spatio-Temporal Dynamic Network consists of CNN and LSTM. A traffic gating mechanism is introduced and a periodic movement attention mechanism is designed to handle periodic temporal movements.
- T-GCN [38]: Temporal Graph Convolutional Network which consists of two parts: graph convolutional network and gated recursive units.
- A3TGCN [39]: Attention Temporal Graph Convolutional Network adds the attention mechanism to GCN and GRU to adjust the importance of different time points.
- STGCN [41]: Spatial-Temporal Graph Convolutional Network integrates ChebNet and gated sequential convolution.
- ASTGCN [42]: Attention-based Spatial-Temporal Graph Convolutional Networks is designed with spatio-temporal attention mechanisms, integrating three components to model three temporal properties separately.

4.1.4. Evaluation metrics

In the experiments, three popular metrics are applied to evaluate the predictive performance of all methods, which are defined as:

$$\begin{cases} MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| \\ RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - y_i)^2} \\ MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{f_i - y_i}{y_i} \right| \end{cases} \quad (12)$$

where y_i is the true value, and f_i is the corresponding predicted value. Specifically, Mean Absolute Error (MAE) describes the difference between the predicted and true values. Root Mean Squared Error (RMSE) measures the degree of dispersion of a set of numbers itself and gives a strong indication of the accuracy of the model. Mean Absolute Percentage Error (MAPE) not only considers the error between the predicted and true values but also the ratio between the error and the true value.

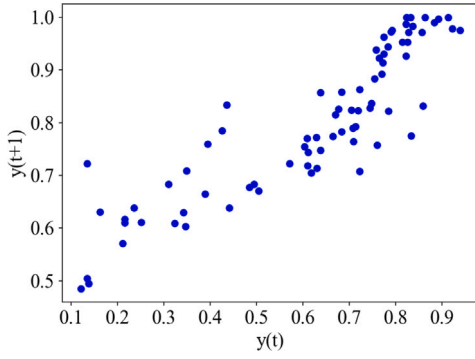


Fig. 8. Lag scatter.

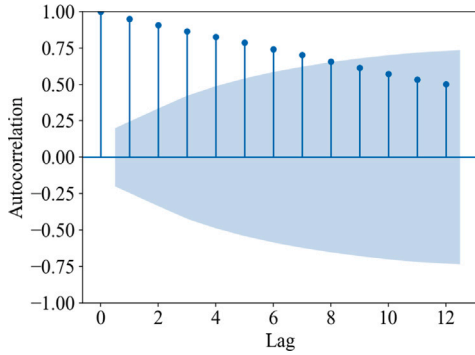


Fig. 9. Autocorrelation of order 1–12.

4.2. Performance analysis

4.2.1. Autocorrelation analysis

Autocorrelation generally exists between time series, but to verify that our work is not entirely based on the autocorrelation of the data, we make a lag scatter plot and autocorrelation figure of the experimental data. Then, the experimental results derived from the autocorrelation model are compared with the true values to illustrate the unpredictability of the autocorrelation method in this work.

As shown in Fig. 8, the lag scatter plot of the time-series data at moments t and $t + 1$ can visualize the degree of autocorrelation of the data. We can see that the distribution of scatter points from the lower left to the upper right is aggregated, which indicates that the data are positively correlated and have a strong autocorrelation. We use the Pearson correlation coefficient to calculate the correlation between the data, as shown in Fig. 9. Lag on the X-axis means the lag coefficient, and the lag value of 12 in the figure denotes the correlation between series values separated by 12 time intervals. Values in $[-1, 0)$ on the Y-axis indicate negative correlation, values in $(0, 1]$ show a positive correlation, and 0 represents no correlation. The shaded area is the 95% confidence interval, and points falling outside or near the 95% confidence interval demonstrate significant non-zero, i.e., the presence of autocorrelation.

We utilize a univariate autoregressive model [60] as an autocorrelation model, and the predicted results are shown in Fig. 10. The X-axis denotes the time step. The Y-axis represents TRI. We can clearly see that the autocorrelation model is not capable of enhancing the forecasting performance, because they only account for the statistical characteristics of the input data and cannot handle complex spatio-temporal data. Although the predicted results fluctuate in the short term, the trend gradually disappears with time.

After validation and analysis, we can conclude that although there is a certain degree of autocorrelation in the data, the autocorrelation model is not applicable to the mining and calculation of the features in this work.

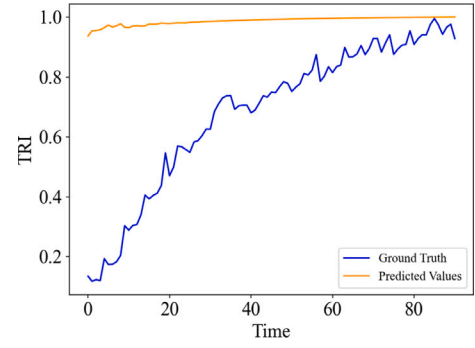


Fig. 10. Predicted results of the autocorrelation model.

Table 1

Comparison of predictive performance.

Models	MAE (1d/2d/3d)	RMSE (1d/2d/3d)	MAPE % (1d/2d/3d)
LSTM [29]	4.87/4.71/4.63	7.21/6.88/6.68	9.81/8.80/8.06
GRU [30]	4.76/4.70/4.61	7.14/6.85/6.62	9.73/8.69/7.98
GCN [22]	4.62/4.55/4.43	6.86/6.66/6.46	9.38/8.51/7.76
STDN [35]	4.53/4.50/4.42	6.84/6.67/6.49	9.21/8.47/7.75
T-GCN [38]	4.51/4.54/4.32	6.80/6.77/6.40	9.21/8.51/7.58
A3TGCN [39]	4.48/4.47/4.42	6.65/6.57/6.44	8.49/7.95/7.77
ASTGCN [42]	3.72/3.68/3.66	5.41/5.36/5.34	7.61/7.62/7.46
STGCN [41]	2.75/2.75/2.73	3.57/3.57/3.54	4.52/4.41/4.28
DeepMeta-TRI	2.68/2.61/2.62	3.44/3.35/3.39	4.37/4.16/4.09

4.2.2. Overall comparison

This section discusses the overall predictive performance of the model as well as the predictive performance w.r.t. the same location over continuous-time and different locations at the same time.

Table 1 compares the predictive performance of our method with eight baselines at three different time steps, and the results are presented as averages. Fig. 11 shows the predictive performance of DeepMeta-TRI compared with 4 types of baseline methods. The X-axis represents the ground truth and the Y-axis shows the predicted value. The closer the scatter point is to the red line, the more accurate the prediction is. It can be seen that DeepMeta-TRI obtains superior results.

On one hand, LSTM and GRU are able to capture temporal correlation effectively. However, in this case, spatial correlation is ignored, resulting in lower accuracy for these prediction tasks. Although GRU is optimized on the basis of LSTM by reducing a gate function, the experimental results of LSTM and GRU are poor overall. On the other hand, GCN can capture spatial features of topological graph structures effectively, but it focuses only on spatial relationships and cannot adequately learn features of time series. From the perspective of space and time, compared with GCN, LSTM, and GRU, which focus only on one of the spatio-temporal relations, STDN that considers both of them improves the predictive performance, and its three metrics obtain 4.47%, 3.05%, and 3.75% improvement relative to GRU, respectively. However, the CNN in STDN does not fully learn spatial features. T-GCN, which also incorporates spatio-temporal relations, uses GCN in processing spatial information, and its three metrics improve to 0.66%, 0.15%, and 0.47% relative to STDN. The A3TGCN with the addition of the attention mechanism highlights the importance of different time points, which boosts the prediction accuracy to some extent. STGCN and ASTGCN are two powerful baselines. Compared with A3TGCN, the three metrics of STGCN get an average improvement of 38.4%, 45.6%, and 45.4%, respectively. They consist entirely of convolution and do not rely on recurrent neural networks, therefore, they have a significant improvement in predictive performance. In comparison with STGCN, DeepMeta-TRI has an average improvement of 4.01%, 4.77%, and 4.54% in the three metrics, respectively. And since the

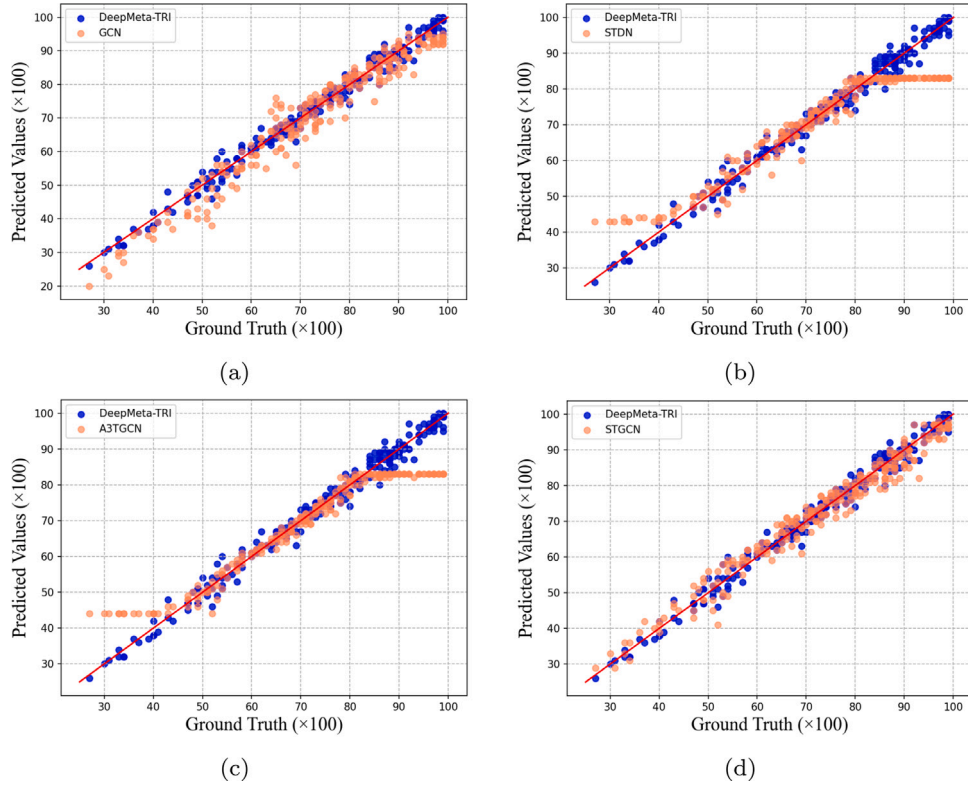


Fig. 11. Visualization of overall forecasting performance.

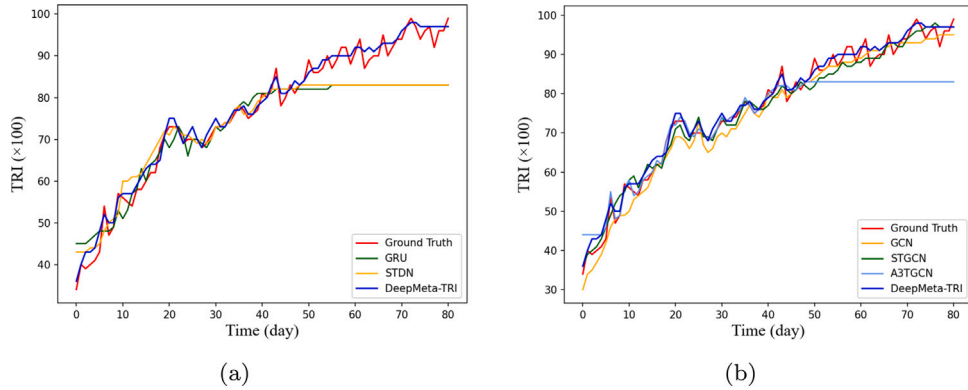


Fig. 12. Visualization of detailed performance w.r.t. one node in continuous time.

number of filters of the temporal convolution module in DeepMeta-TRI depends on the number of layers (only one filter per layer) rather than the input length, the memory requirement of DeepMeta-TRI is lower than that of the RNN-based model, where the GPU memory requirements of DeepMeta-TRI, STDN and A3TGCN are 3657.2MB, 4319.4MB, and 4215.8MB. DeepMeta-TRI shows stable and superior predictive performance in different time ranges. The main reason is that DeepMeta-TRI introduces external auxiliary factors such as COVID-19 epidemics based on the consideration of spatio-temporal correlations, which facilitates the model to perceive and learn from a global perspective. The relationship between external auxiliary factors and dynamic TRI is also integrated, and the performance of the network is enhanced by the meta learner, which realizes the learning of complex and diverse dynamic spatio-temporal correlations and trends of TRI.

The experimental results are visualized from two perspectives of the same node at continuous-time and different nodes at the same time to further analyze the underfitting problem.

It can be concluded from the metrics in Table 1 that the prediction accuracy of some methods is ideal, and they can achieve a certain degree of fitting for the more aggregated data in the middle part seen in Fig. 12. However, RNN-based GRU, STDN, and A3TGCN have poor fitting results at the beginning and end, and no trend is even observed. In Figs. 12 and 13, although the accuracy of GCN in Table 1 is not as high as that of A3TGCN and STDN, convolution-based GCN and STGCN alleviate the above problems. The above method also suffers from the problem of unsatisfactory fitting of local peaks when the data suddenly changes significantly. In contrast, DeepMeta-TRI has a stronger perception and fit for the overall trend, local peaks, and values at the beginning and end.

In summary, by comparing the metrics in Table 1 and the visualizations in Figs. 12 and 13, we can draw that DeepMeta-TRI has significant competitiveness and superiority. First, compared to RNN-based methods, TCM in DeepMeta-TRI is able to process long-term sequences in a non-recursive manner, which facilitates parallel computation and mitigates gradient explosion. Another difference from RNN is

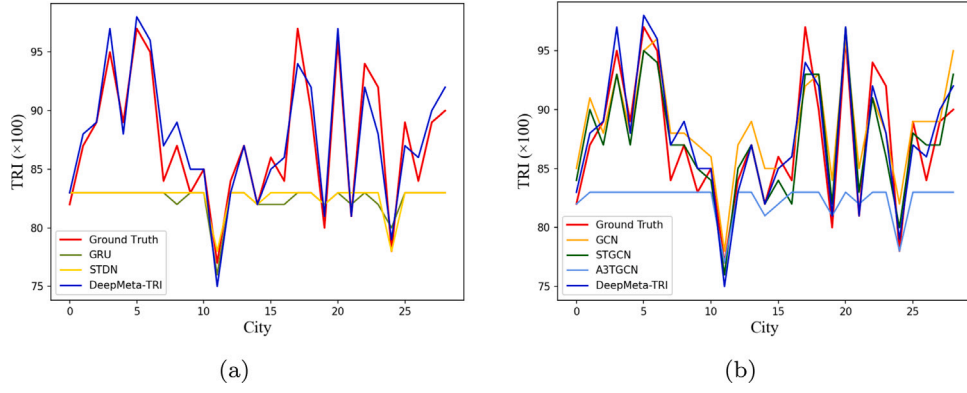


Fig. 13. Visualization of detailed performance w.r.t. all nodes at the same time.

Table 2

Performance comparison of ablation experiments.

Models	Running time (s/epoch)	MAE	RMSE	MAPE %
No MGF in MetaGCN	0.6317	2.96	3.82	4.86
No COVID-19 data	0.6440	2.97	3.78	4.84
No ML in MetaGCN	0.6052	2.77	3.59	4.56
No MGF in MetaTCM	0.6403	2.75	3.54	4.35
No hubness	0.6389	2.75	3.56	4.40
No pp COVID-19 data	0.6716	2.72	3.51	4.34
No ML in MetaTCM	0.6188	2.68	3.43	4.22
DeepMeta-TRI	0.6449	2.63	3.39	4.20

that the gradient is not in the temporal direction but the network depth direction and is more stable due to the use of residual connections. As the network deepens, its receptive field grows exponentially without introducing additional parameters. The feature extraction capability of the low-parameter convolution provides more detailed long-term dependence information. Then, to strengthen the dynamic synchronization features of urban nodes, we adopt GCN instead of CNN to establish the spatial relative relationships among nodes in the graph structure. Further, the effective integration of external auxiliary information such as the COVID-19 pandemic, traffic hubness, etc. enhances DeepMeta-TRI's perception of semantic and temporal global perspectives. And subsequently captures the intrinsic relationship between external auxiliary information and dynamic TRI in a multi-scale post-fusion manner. Finally, the introduction of meta-learning based on parameter weight generation augments the information extraction capability of TCM and GCN. Thus, DeepMeta-TRI is more effective and stable in TRI prediction.

4.2.3. Ablation experiments

To illustrate the effectiveness of incorporating the COVID-19 pandemic data and traffic hubness, the meta gating fusion module, and the meta learner, we conduct ablation experiments and show the results in Table 2.

- *No MGF in MetaGCN*: the TRI context of the meta gating fusion module (MGF) in the meta graph convolution network module is removed, i.e., no fusion is applied, and only the processed external static auxiliary information is fed to the meta learner.
- *No MGF in MetaTCM*: the TRI context of the meta gating fusion module (MGF) in the meta temporal convolution module is removed, i.e., no fusion is applied, and only the processed external dynamic auxiliary information is input to the meta learner.
- *No ML in MetaGCN*: remove the meta learner (ML) of the meta graph convolution network module.
- *No ML in MetaTCM*: remove the meta learner (ML) of the meta temporal convolution module.

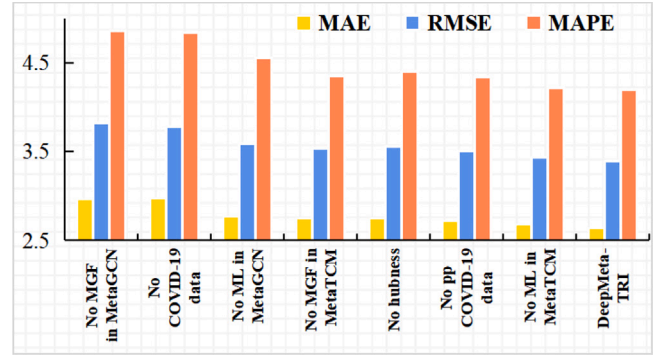


Fig. 14. Performance of ablation experiments.

- *No hubness*: remove traffic hubness from the input data of the meta temporal convolution module.
- *No COVID-19 data*: remove four types of COVID-19 pandemic data from the input data of the meta temporal convolution module.
- *No pp COVID-19 data*: remove the pre-processing of the COVID-19 pandemic data of the meta temporal convolution module.

From Table 2 and Fig. 14, we can visualize that removing the MGF of MetaGCN has the greatest impact on the prediction effect of the model, mainly due to learning the meta-features of external auxiliary information and considering the dynamic TRI while capturing the intrinsic relationship between spatio-temporal correlation and dynamic TRI. Compared with DeepMeta-TRI, the metrics are reduced by 11.1%, 11.2%, and 13.5%, respectively, verifying the necessity of fusing external auxiliary information with dynamic TRI in a multi-scale post-fusion manner. Secondly, the proposed model *No COVID-19 data* illustrates that COVID-19 pandemic data is also an essential part and its influence is stronger than the traffic hubness and pre-processing before the COVID-19 data, because of the significant dynamic cross-correlation between COVID-19 pandemic data and TRI. Comparison of DeepMeta-TRI with *No ML in MetaGCN* or *No ML in MetaTCM* demonstrates the effectiveness of meta-learning based on weight generation for the prediction of TRI. Although the metrics of removing the meta learner in MetaTCM are not greatly reduced compared to DeepMeta-TRI, DeepMeta-TRI still has 1.86%, 1.16%, and 0.47% improvement in metrics. Despite the integration of multiple modules in DeepMeta-TRI, its running time is not significantly increased. As can be seen in Fig. 15, the changing speed of each loss varies significantly with epoch, and the loss of DeepMeta-TRI, which integrates multiple modules, decreases faster and changes smoother in the later stages.

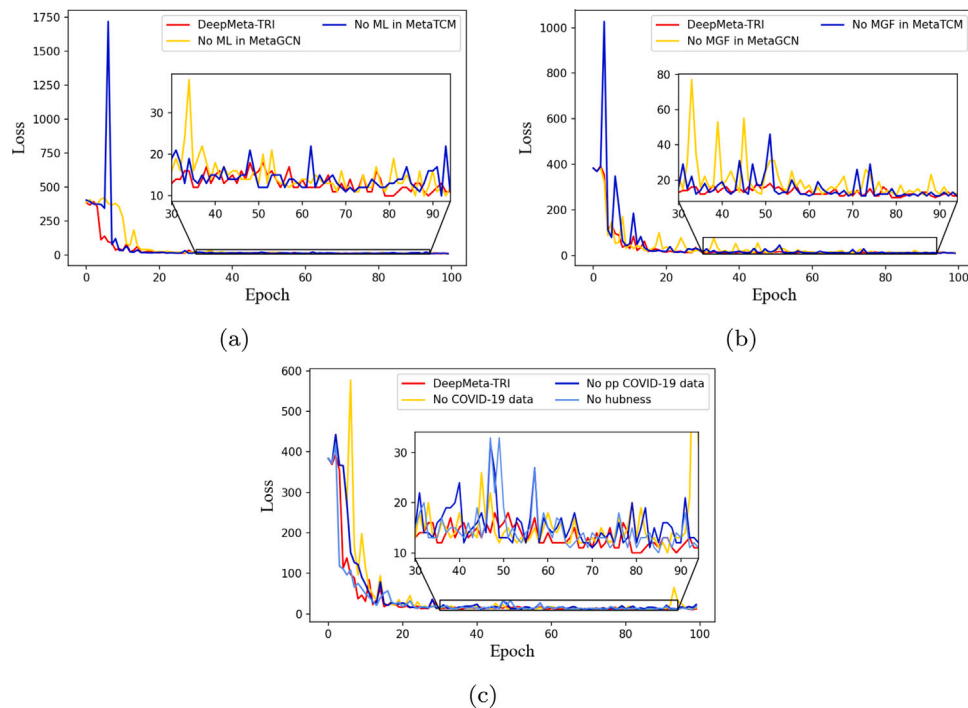


Fig. 15. Loss curves of the ablation experiment.

5. Conclusion

Traffic is an indispensable guarantee for urban development. The prediction of TRI during the COVID-19 pandemic can be deployed and applied to transportation-related platforms to provide informative references for the analysis and prediction of public safety disasters. We propose a deep spatio-temporal meta-learning framework for traffic revitalization index prediction (DeepMeta-TRI). DeepMeta-TRI consists of a temporal convolution module, a meta-graph convolution network module (MetaGCN), and a meta-temporal convolution module (MetaTCM). Both MetaGCN and MetaTCM include the meta learner and meta gating fusion module, which integrates the external auxiliary information, such as COVID-19 data and traffic hubness. We conduct extensive experiments to evaluate DeepMeta-TRI on real-world traffic revitalization index, and the experimental results demonstrate that our method exhibits highly competitive performance both in metrics and degree of the fitting. In the future, we will further explore the factors affecting the traffic revitalization index, investigate its learning and fusion approach, and extend our framework to a broader range of traffic prediction tasks.

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.

Data availability

I have shared the link to the data in the manuscript.

Acknowledgments

This research was supported in part by National Natural Science Foundation of China under Grant No. 62106117, and Shandong Provincial Natural Science Foundation, China under Grant No. ZR2021QF084.

References

- [1] S. Parr, B. Wolshon, J. Renne, P. Murray-Tuite, K. Kim, Traffic impacts of the COVID-19 pandemic: Statewide analysis of social separation and activity restriction, *Nat. Hazards Rev.* 21 (3) (2020) 04020025.
- [2] https://zjhjy.mot.gov.cn/yaowendt/jiaotongyw/202002/t20200219_3426419.html.
- [3] Y. Zhang, L. Yang, X. Wang, Analysis and calculating of comprehensive urban vitality index by multi-source temporal-spatial big data and EW-TOPSIS, in: 2021 IEEE International Conference on Data Science and Computer Application, 2021, pp. 196–201.
- [4] G. Nian, B. Peng, D.J. Sun, W. Ma, B. Peng, T. Huang, Impact of COVID-19 on urban mobility during post-epidemic period in megacities: From the perspectives of taxi travel and social vitality, *Sustainability* 12 (19) (2020) 7954.
- [5] B. Goenaga, N. Matini, D. Karanam, B.S. Underwood, Disruption and recovery: Initial assessment of covid-19 traffic impacts in North Carolina and Virginia, *J. Transp. Eng., Part A: Syst.* 147 (4) (2021) 06021001.
- [6] A. Bassolas, R. Gallotti, F. Lamanna, M. Lenormand, J.J. Ramasco, Scaling in the recovery of urban transportation systems from massive events, *Sci. Rep.* 10 (1) (2020) 1–13.
- [7] A.M. Nagy, V. Simon, Improving traffic prediction using congestion propagation patterns in smart cities, *Adv. Eng. Inform.* 50 (2021) 101343.
- [8] F. Guo, Y. Wang, Y. Qian, Computer vision-based approach for smart traffic condition assessment at the railroad grade crossing, *Adv. Eng. Inform.* 51 (2022) 101456.
- [9] Z. Lv, J. Li, C. Dong, Z. Xu, DeepSTF: A deep spatial-temporal forecast model of taxi flow, *Comput. J.* (2021).
- [10] Y. Wang, J. Li, A. Zhao, Z. Lv, G. Lu, Temporal attention-based graph convolution network for taxi demand prediction in functional areas, in: *International Conference on Wireless Algorithms, Systems, and Applications*, Springer, Cham, 2021, pp. 203–214.
- [11] S. Rath, A. Tripathy, A.R. Tripathy, Prediction of new active cases of coronavirus disease (COVID-19) pandemic using multiple linear regression model, *Diabetes Metab. Syndr.: Clin. Res. Rev.* 14 (5) (2020) 1467–1474.
- [12] M.S. Al-Musaylh, R.C. Deo, J.F. Adamowski, Y. Li, Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia, *Adv. Eng. Inform.* 35 (2018) 1–16.
- [13] F.I. Rahman, Short term traffic flow prediction using machine learning-KNN, SVM and ANN with weather information, *Int. J. Traffic Transp. Eng.* 10 (3) (2020).
- [14] F. Li, Z. Liu, T. Li, H. Ju, H. Wang, H. Zhou, Privacy-aware PKI model with strong forward security, *Int. J. Intell. Syst.* (2020).
- [15] C. Cui, F. Li, T. Li, J. Yu, R. Ge, H. Liu, Research on direct anonymous attestation mechanism in enterprise information management, *Enterprise Inf. Syst.* 15 (4) (2021) 513–529.

- [16] X. Liu, Y. Xia, H. Yu, J. Dong, M. Jian, T.D. Pham, Region based parallel hierarchy convolutional neural network for automatic facial nerve paralysis evaluation, *IEEE Trans. Neural Syst. Rehabil. Eng.* 28 (10) (2020) 2325–2332.
- [17] Y. Ming, X. Meng, C. Fan, H. Yu, Deep learning for monocular depth estimation: A review, *Neurocomputing* (2021).
- [18] L. Tran, M.Y. Mun, M. Lim, J. Yamato, N. Huh, C. Shahabi, DeepTRANS: A deep learning system for public bus travel time estimation using traffic forecasting, *Proc. VLDB Endow.* 13 (12) (2020) 2957–2960.
- [19] Z. Fang, L. Pan, L. Chen, Y. Du, Y. Gao, MDTP: A multi-source deep traffic prediction framework over spatio-temporal trajectory data, *Proc. VLDB Endow.* 14 (8) (2021) 1289–1297.
- [20] T. Nguyen, G. Nguyen, B.M. Nguyen, Eo-CNN: An enhanced CNN model trained by equilibrium optimization for traffic transportation prediction, *Procedia Comput. Sci.* 176 (2020) 800–809.
- [21] H. Zheng, F. Lin, X. Feng, Y. Chen, A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction, *IEEE Trans. Intell. Transp. Syst.* (2020).
- [22] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
- [23] M. Huisman, J.N. Van Rijn, A. Plaata, A survey of deep meta-learning, *Artif. Intell. Rev.* 54 (6) (2021) 4483–4541.
- [24] N.K. Chikkakrishna, C. Hardik, K. Deepika, N. Sparsha, Short-term traffic prediction using sarima and FbPROPHET, in: 2019 IEEE 16th India Council International Conference, IEEE, 2019, pp. 1–4.
- [25] D.W. Xu, Y.D. Wang, L.M. Jia, Y. Qin, H.H. Dong, Real-time road traffic state prediction based on ARIMA and Kalman filter, *Front. Inf. Technol. Electron. Eng.* 18 (2) (2017) 287–302.
- [26] A. Zhao, J. Dong, J. Li, L. Qi, H. Zhou, Associated spatio-temporal capsule network for gait recognition, *IEEE Trans. Multimedia* (2021).
- [27] A. Zhao, J. Li, J. Dong, L. Qi, Q. Zhang, N. Li, H. Zhou, Multimodal gait recognition for neurodegenerative diseases, *IEEE Trans. Cybern.* (2021).
- [28] Z. Lv, J. Li, H. Li, Z. Xu, Y. Wang, Blind travel prediction based on obstacle avoidance in indoor scene, *Wireless Commun. Mob. Comput.* 2021 (2021).
- [29] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
- [30] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.
- [31] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for city-wide crowd flows prediction, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
- [32] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction, *Sensors* 17 (4) (2017) 818.
- [33] J. Ke, H. Zheng, H. Yang, X.M. Chen, Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach, *Transp. Res. Part C: Emerg. Technol.* 85 (2017) 591–608.
- [34] L. Liu, J. Zhen, G. Li, G. Zhan, Z. He, B. Du, L. Lin, Dynamic spatial–temporal representation learning for traffic flow prediction, *IEEE Trans. Intell. Transp. Syst.* (2020).
- [35] H. Yao, X. Tang, H. Wei, G. Zheng, Z. Li, Revisiting spatial–temporal similarity: A deep learning framework for traffic prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 5668–5675.
- [36] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, S. Ji, Temporal multi-graph convolutional network for traffic flow prediction, *IEEE Trans. Intell. Transp. Syst.* (2020).
- [37] J. Zhu, X. Han, H. Deng, C. Tao, L. Zhao, L. Tao, H. Li, Kst-gcn: A knowledge-driven spatial–temporal graph convolutional network for traffic forecasting, 2020, arXiv preprint arXiv:2011.14992.
- [38] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, H. Li, T-gcn: A temporal graph convolutional network for traffic prediction, *IEEE Trans. Intell. Transp. Syst.* 21 (9) (2019) 3848–3858.
- [39] J. Bai, J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, H. Li, A3T-GCN: Attention temporal graph convolutional network for traffic forecasting, *ISPRS Int. J. Geo-Inf.* 10 (7) (2021) 485.
- [40] Z. Lv, J. Li, C. Dong, H. Li, Z. Xu, Deep learning in the COVID-19 epidemic: A deep model for urban traffic revitalization index, *Data Knowl. Eng.* 135 (2021) 101912.
- [41] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, 2017, arXiv preprint arXiv:1709.04875.
- [42] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial–temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 922–929.
- [43] M. Ren, W. Zeng, B. Yang, R. Urtasun, Learning to reweight examples for robust deep learning, in: International Conference on Machine Learning, PMLR, 2018, pp. 4334–4343.
- [44] L. Bertinetto, J.F. Henriques, J. Valmadre, P. Torr, A. Vedaldi, Learning feed-forward one-shot learners, in: Advances in Neural Information Processing Systems, 2016, pp. 523–531.
- [45] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, D. Meng, Meta-weight-net: Learning an explicit mapping for sample weighting, 2019, arXiv preprint arXiv:1902.07379.
- [46] Z. Hu, B. Tan, R. Salakhutdinov, T. Mitchell, E.P. Xing, Learning data manipulation for augmentation and weighting, 2019, arXiv preprint arXiv:1910.12795.
- [47] D. Ha, A. Dai, Q.V. Le, Hypernetworks, 2016, arXiv preprint arXiv:1609.09106.
- [48] J. Chen, X. Qiu, P. Liu, X. Huang, Meta multi-task learning for sequence modeling, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.
- [49] Y. Guo, N.M. Cheung, Attentive weights generation for few shot learning via information maximization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13499–13508.
- [50] K. Huang, M. Zitnik, Graph meta learning via local subgraphs, *Adv. Neural Inf. Proc. Syst.* 33 (2020).
- [51] C. Zhang, M. Ren, R. Urtasun, Graph hypernetworks for neural architecture search, 2018, arXiv preprint arXiv:1810.05749.
- [52] V. Garcia, J. Bruna, Few-shot learning with graph neural networks, 2017, arXiv preprint arXiv:1711.04043.
- [53] A. Sankar, X. Zhang, K.C.C. Chang, Meta-gnn: metagraph neural network for semi-supervised learning in attributed heterogeneous information networks, in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2019, pp. 137–144.
- [54] G. Zhang, D. Liu, Causal convolutional gated recurrent unit network with multiple decomposition methods for short-term wind speed forecasting, *Energy Convers. Manag.* 226 (2020) 113500.
- [55] R. Liu, W. Cai, G. Li, X. Ning, Y. Jiang, Hybrid dilated convolution guided feature filtering and enhancement strategy for hyperspectral image classification, *IEEE Geosci. Remote Sens. Lett.* (2021).
- [56] L. Gong, X. Liu, L. Wu, Y. Liu, Inferring trip purposes and uncovering travel patterns from taxi trajectory data, *Cartogr. Geograph. Inf. Sci.* 43 (2) (2016) 103–114.
- [57] <https://gaia.didichuxing.com>.
- [58] <http://www.nhc.gov.cn/>.
- [59] <https://www.datayicai.com/>.
- [60] K.F. Sellers, S.J. Peng, A. Arab, A flexible univariate autoregressive time-series model for dispersed count data, *J. Time Ser. Anal.* 41 (3) (2020) 436–453.