Local Information Enhancement Based Traffic Speed Prediction

Yu Lianfei[1], Ma Cheng[1], Qu Zhijian[1*]
[1](School of Computer Science and Technology, Shandong University of Technology, China)

ABSTRACT : *Traffic prediction is a very important research content in the field of intelligent transportation. However, due to the complex spatio-temporal correlation in traffic data, traffic prediction still faces severe challenges. To better capture the spatio-temporal correlation of traffic data, a deep learning-based traffic volume prediction model called Spatio-Temporal Convolutional Transformer (ST-CT) was presented in this paper. The model consists of three main parts, including: the local information enhancement module, the modified graph convolutional neural network (M-GCN), and the gated recursive unit (GRU). The local information enhancement module is composed of the convolutional neural network (CNN), the transposed convolutional neural network and the transformer encoder layer. In the ST-CT, the local information enhancement module is employed to capture the global and local correlation of the traffic data, the M-GCN is employed to capture spatial correlation via learning the complex topology structure of the transport network, the GRU is employed to capture the temporal correlation via learning the time change of traffic volume. We tested the predictive performance of ST-CT on two real datasets Los-loop and SZ-taxi. The test result indicates that the prediction performance of ST-CT is better than the comparison models.*

KEYWORDS -*Traffic prediction, Spatio-temporal correlation, GCN, Transformer*

I.INTRODUCTION

Today modern cities are moving in the direction of smart cities. The rapid rise of the population and the acceleration of urbanization have put a lot of strain on city traffic management. Traffic problems such as traffic congestion and traffic safety are becoming increasingly severe [1]. Accurate traffic prediction can not only help people know the road conditions ahead of time, plan their own travel, and improve travel efficiency; it can also provide administrators with a scientific basis for allocating road resources ahead of time, alleviating traffic congestion, and ensuring public safety [2]. Fortunately, with the development of information technology and transportation industry, more and more sensors are placed and a large number of traffic data are collected through sensors. However, accurate traffic prediction is difficult due to the nonlinearity, spatial-temporal correlation of traffic data and complexity of traffic road network. The spatial correlation of traffic data is mainly reflected in the generally situation, adjacent roads are highly correlated, while distant roads tend to be weakly correlated. As shown in Fig. 1, the spatially adjacent points 2, 5, and 4 were considered more important in predicting the traffic volume of node 1, while distant points such as 3, 6, 7, 8, and 9 were considered weakly correlated. The temporal correlation of traffic data is mainly reflected in the periodicity and trend of traffic data. Fig. 2 (a) depicts changes in traffic over a week and we can see cyclical changes in traffic over a week. The change of traffic volume is closely related to time. As shown in Fig. 2 (b), the daily traffic volume changes with time.

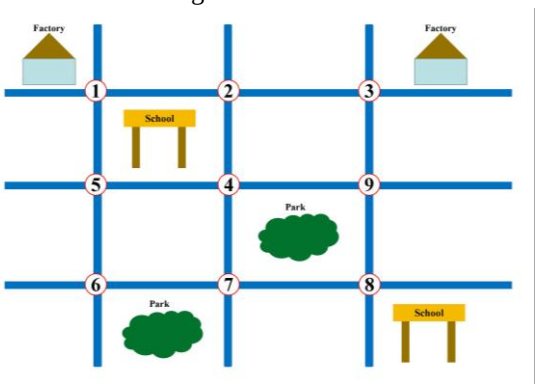

Fig. 1 Spatial correlation

People have conducted extensive research to address the aforementioned issues. In the early days, statistical methods including ARIMA and its variants [3, 4], Kalman filter [5] were popular. However, the traffic data is nonlinear and dynamic, contradicting

these approaches' linear stationary assumption, resulting in poor predicting results. Traditional machine learning methods, such as support vector regression [6, 7], K-nearest Neighbor model [8] and Bayes model [9], can model nonlinearity in traffic data and extract more complex data correlations. However, the ability of these models to predict outcomes is mainly determined by the features of artificial design and it's hard to learn the spatial-temporal correlation of data. With the rapid development of deep learning, deep learning technology has been used to mine spatial-temporal correlation for traffic prediction tasks. Recurrent neural network (RNN) [10] and its variants LSTM [11] or GRU [12] are often used to model temporal correlation. To better capture the spatial correlation of data, some people used CNN [13] and graph convolutional network (GCN) [14] to predict traffic data. At the same time, in order to better obtain the global correlation of the data, many people began to use the attention mechanism to capture the global correlation of the data [15].

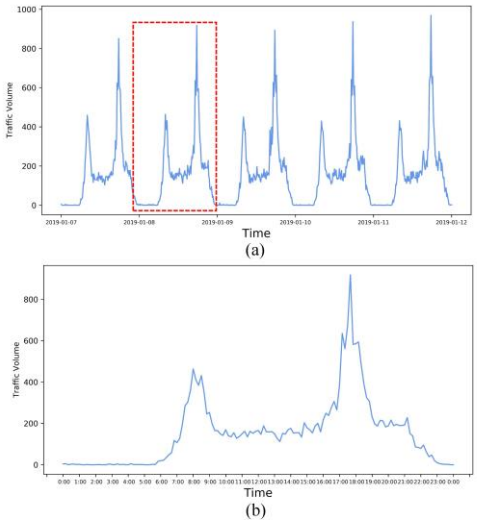

Fig. 2 Temporal correlation

Despite the fact that these methods improved the prediction effect, they had flaws in studying spatio-temporal correlation. These models only used the topological relations of the traffic network to capture spatial correlation, so the captured spatial correlation was incomplete. These models, on the other hand, only considered the data's global correlation and ignored the data's local correlation. In order to solve the above problems, a modified traffic prediction method, Spatio-Temporal Convolutional Transformer (ST-CT) is proposed for traffic prediction tasks. Our contribution is threefold:

1. A modified graph neural network (M-GCN) with a position attention mechanism was designed to solve the problem that traditional graph neural networks only rely on the given topological graphs to capture spatial correlation of data. By using a position attention mechanism, M-GCN can better capture spatial correlation of data by capturing the traffic volume information on adjacent roads.

2. The local information enhancement module is composed of the CNN and the Transformer encoder layer, and was designed to simultaneously capture the global and local correlations of data.

3. We use two real-world traffic datasets to evaluate our approach. The results show that compared with all baseline methods, the prediction error of this method is reduced and the correlation coefficient is improved, which demonstrates the effectiveness of our method.

The rest of the paper is organized as follows: Sect. 2 summarizes the related work of traffic volume prediction. Sect. 3 describes our method in detail. In Sect. 4, we evaluate the predictive performance of ST-CT using real-world traffic data sets. In Sect. 5 is the conclusion of this paper.

II. Related Work

Deep learning has the powerful ability to theoretically approach arbitrary complex functions and can model more complex patterns in various traffic tasks. In addition, due to the improvement of computing capacity (such as GPU) and a large amount of traffic data, deep learning-based technology has been widely applied in various traffic applications and achieved the better prediction results. For example, to predict traffic speed, Ma et al. [16] proposed a CNN-based deep learning method that captured the spatial correlation of data by traffic volumes as images. To gain a better understanding of data's spatio-temporal correlation, a series of models integrating CNN and LSTM [17-20] were constructed to predict short-term traffic flow. Shi et al. [21] proposed a Convolutional LSTM (ConvLSTM) and used it to build a trainable end-to-end model of the short-term rainfall forecasting problem. Lv et al. [22] used CNN to extract the spatial correlation of adjacent roads and used LSTM to extract features from a time series perspective. CNN captured spatial correlation by splitting traffic data into grids one by one. Although these methods achieved good results, they mainly modeled the Euclidean correlation between regions, but many transportation networks are graphically structured in nature, such as road networks and subway networks.

The non-Euclidean correlation is a better fit for describing the road system. The spatial features learned on CNN are not optimal for representing graph-based traffic networks [23]. In other words, these methods are only suitable for data based on grid maps and not for data based on multiple sensors. In terms of spatial correlation, these methods are inadequate.

GCN extends the convolution operation to more general graph structure data, which is more suitable for representing the traffic network structure and extracting the spatial correlation of data. GNN [24] was one of the latest technologies for processing non-Euclidean structured data and was widely used in traffic volume forecast tasks [23, 25, 26]. The spatial correlation between roads was captured using GNN in these methods. However, traditional GNN only uses the traffic network's topological relations to learn spatial information, resulting in the central node uniformly learning the information of adjacent nodes, but the influence of adjacent nodes on the central node is not always equal. As a result, the information of adjacent nodes should not be equally learned by the central node. People have also conducted extensive research in order to address the shortcomings of traditional GNN and better capture spatial correlation. Wu [27] introduced the adaptive adjacency matrix as a constraint to graph convolutional to automatically discover unknown graph structures from data for spatial correlation. Guo [28] learned the optimization map by data-driven method in the training stage, and revealed the potential relationship between sections of traffic data to predict traffic flow. Bai [29] decomposed the shared parameter part of traditional graph convolutional by matrix, so as to obtain node-specific parameters and capture node-specific modes. These methods improved on traditional GNN, which can better learn spatial information and improve prediction accuracy but they ignored the correlation of the data itself. In recent years, attention mechanisms have been widely used in various tasks such as natural language processing, image captioning and speech recognition. The attention mechanism's goal is to select from all input the information that is critical to the task at hand. Wang et al. [30] used a learning position attention mechanism in GCN and used transformer to learn global correlation. However, they did not consider complementary information, only learned the global correlation and ignored the internal relationship of data, so the local correlation was not extracted and lacking global and local mutual relations.

Based on this background, this study proposes a modified deep learning network method, which can extract complex temporal and spatial features from traffic data, and learn the global correlation and local correlation of the data itself.

III.METHODOLOGY

3.1 Problem Definition

In this study, the goal of traffic prediction is to predict traffic information in a future period of time according to historical traffic information on the road. In our approach, traffic information is a general concept, which can be traffic speed, traffic flow or traffic density. Without losing the versatility, taking traffic speed as an example, traffic information is extracted in the experimental section.

$Definition$ 1: (Traffic networks) We use an unweighted graph $G = (V, E)$ to describe the topology of the road network, and treat each road as a node,where V is a set of road nodes, $V = \{v_1, v_2, \cdots v_N\}$, $N$is the number of nodes, and E is the set of edges. The adjacency matrix A is used to represent the connections between roads, $A \in R^{N \times N}$, the adjacency matrix contains only elements 0 and 1. If there is no connection between two roads, the element is 0. If there is a connection between two roads, it is represented by 1.

$Definition$ 2 : (Traffic speed forecasting) Given the traffic network $G = (V, E)$ and the historical traffic information, X_t is used to represent the traffic volume at time t, we aim to build a model f, which can take a sequence of length n as input and predict the traffic information for the next T time steps. As shown in Eq. (1):

$$[X_{t+1}, \cdots X_{t+T}] = f\big(G; (X_{t-n-1}, \cdots X_{t-1}, X_t)\big) \quad (1)$$

3.2 Overview

The model is composed of three parts: the local information enhancement module, the M-GCN, and the GRU. The local information enhancement module is used to simultaneously learn the global correlation and local correlation of data. It consists of the CNN, the transposed convolutional neural network and the transformer encoder layer, which solves the problem that the transformer encoder layer cannot capture local correlation. In this model, we have multiple local information enhancement units arranged in series in order to obtain information at different local locations. M-GCN is used to capture spatial correlation of data. It is not like the traditional

GCN to capture spatial information by learning given topology graphs, but by the location attention mechanism to capture the spatial information of each node. GRU is used to capture time correlation. As shown in Fig. 3, we first use historical time series data of length n as input and then input the time series data into the local information enhancement module to capture the global and local correlation of data. At the same time, in order to avoid the vanishing gradients problem, residual connections [31] are used to connect the outputs. Secondly, the obtained time series with global and local correlation features are input into the M-GCN to capture the spatial correlation of data. Then the time series are input into the GRU to capture the temporal correlation of data. Finally, get the result through the full connection layer.

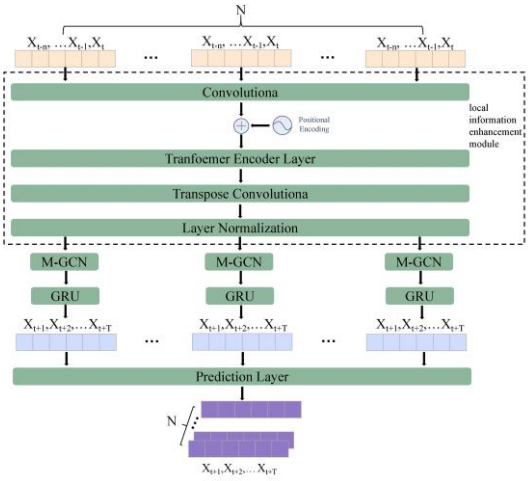

Fig. 3 The proposed model framework

3.3 Global and Local Correlation

Each local information enhancement module consists of a CNN with a convolution kernel size of K, a transposed convolutional neural network with a convolution kernel size of K, and a transformer encoder layer. The transformer encoder layer consists of a multi-head attention layer and a feed forward neural network layer. Our local information enhancement module framework is shown in Fig. 4. First, the data is inputted into the CNN with a convolution kernel of width$K$. The CNN is searched for K neighbors of the input elements, padding is set to 0 in this experiment, making the length of each sequence become shorter $K-1$. The data is processed by CNN goes into the multiple attention layer [32]. The multiple attention layer is based on the dot product attention mechanism. In the multi-attention layer, the elements in sequence position i are related to all elements in the sequence. The inputs of the attention function consist of queries, keys with dimension d_k and values with dimension d_v of all the positions in the sequence. We compute the dot products of a given query with all keys, divide each by $\sqrt{d_k}$ and then apply a softmax function to obtain the attention scores for each position. These attention scores are then used as weights. In practice, the attention for the queries of all positions simultaneously is computed as shown in Eq. (2):

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}V\right) \quad (2)$$

where $Q,K \in R^{T\times d_k}$ and $V \in R^{T\times d_v}$ denote the queries, keys, and values for all the nodes. Specifically, the i-th row of Q denotes the query corresponding to the position i in the sequence. Multi-head attention allows the model to simultaneously attend to information from various representation subspaces at various locations. When using a single attention head, averaging prevents this. Thus, multi-headed attention works better than it. The equation for multiple attention is shown in Eq. (3):

$$MultiHead(Q,K,V) = Concat(head_1,\cdots head_h)W^O \quad (3)$$

$h$is the number of heads. The $head_i$ meanings are shown in Eq. (4):

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \quad (4)$$

d_{model} is our input dimension, $W_i^Q \in R^{d_{model}\times d_k}$, $W_i^K \in R^{d_{model}\times d_k}$, $W_i^V \in R^{d_{model}\times d_k}$ and $W^O \in R^{hd_v\times d_{model}}$.However, the multi-head attention layer ignores relative positions in the sequence because it treats different positions equally in calculating the attentional function. To ensure the multi-head attention layer know the relative position of position i in the entire sequence, the position encoding e_t for each position is adopted. Where e_t is defined as shown in Eq. (5):

$$e_t = \begin{cases} \sin\left(\frac{t}{10000^{\frac{2i}{d_{model}}}}\right), & if\ t = 0,2,4\cdots \\ \cos\left(\frac{t}{10000^{\frac{2i}{d_{model}}}}\right), & otherwise \end{cases} \quad (5)$$

After the multi-head attention layer, the output is passed to the feed forward neural network layer. Then the data is inputted into the transposed convolutional neural network with a convolution kernel of width K. Similarly, the transposed convolutional neural network is also searched for K adjacent elements of the input elements without padding, making the length of each sequence increase by $K-1$. As shown in Fig. 4, after the transposed

convolutional neural network, there is a layer normalization [33]. This makes up a complete local information enhancement module.

In order to be able to capture information from different local units, multiple local information enhancement modules are set up. The size of the convolution kernel for each local information enhancement module is different. With different sizes of convolution kernels, the obtained receptive fields are also different, so that different local information can be captured. In general, the larger the kernel, the larger the field of perception, the more information can be learned and the better the global features can be characterised. However, too large a convolution kernel leads to an increase in parameters, which is not conducive to increasing the depth of the model, and also increases the computational power required. In the case of considering the data dimension, five local information enhancement modules are set and set the convolution kernel size to 9, 7, 5, 3, and 1 respectively in this experiment. At the same time, in order to better enable the model to learn the information from the data and prevent the gradient problem caused by too deep layers, the residual connection is set at the end of the module.

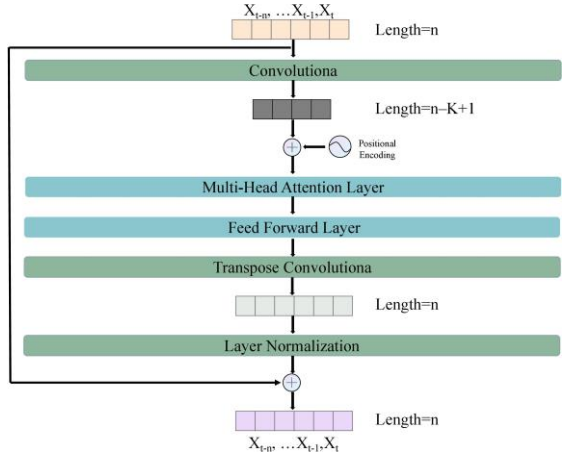

Fig. 4 The local information enhancement module framework

3.4 Modeling the Spatial Correlation

Obtaining complex spatial correlations is a key problem in traffic prediction. In order to capture spatial correlation, we adopt the GCN to transform and disseminate information in the data. Specifically, given input information on $X_{in} \in R^{N \times d_{in}}$ on the network, the output $X_{out} \in R^{N \times d_{out}}$ can be generated as shown in Eq. (6):

$$X_{out} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X_{in}W\right) \quad (6)$$

where σ is a nonlinear activation function and $RELU(\cdot)$ is used in this experiment, W is the parameter for learning, I_N is the N -dimensional identity matrix, $\tilde{A} = A + I_N$ is the refined adjacency matrix and $\tilde{D}$ is the refined degree matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. In Eq. (6), the operation entirely based on the road connection information of topology graph, but in most cases, the spatial correlation is not fully captured.

Meanwhile, in terms of traffic prediction, the closer the two roads are to each other, the more likely the traffic conditions on the two roads will affect each other. So, we try to capture the spatial correlation for each node by learning location representations.

Specifically, for each node v_i, we try to learn the potential location representation p_i by using attention mechanism. Then, the pairwise relationship between any road nodes is modeled as shown in Eq. (7):

$$R[i,j] = \frac{\exp\left(\emptyset\left(Score(p_i, p_j)\right)\right)}{\sum_{k=1}^{N} exp\left(\emptyset\left(Score(p_i, p_k)\right)\right)} \quad (7)$$

the $Score()$ is a relation score function modeled using the dot product, as shown in Eq. (8):

$$Score(p_i, p_j) = p_i^T p_j \quad (8)$$

We can then perform a GCN operation on the newly learned relational matrix as shown in Eq. (9):

$$X_{out} = \sigma\left(\tilde{D}_R^{-\frac{1}{2}}\tilde{R}\tilde{D}_R^{-\frac{1}{2}}X_{in}W^{(l)}\right) \quad (9)$$

where $\tilde{R} = R + I_N$ and $\tilde{D}_R$ is the degree matrix for $\tilde{R}$.

Through the above method, the shortcomings of traditional GCN which is highly dependent on topological graph and the information of adjacent nodes is equally learned by central nodes can be conquered. Meanwhile it can excavate deeper hidden relationships between nodes.

3.5 Modeling the Temporal Correlation

Obtaining temporal correlation is another key problem in traffic forecasting. The most widely used method to capture temporal correlation of data is the RNN. However, due to defects such as gradient disappearance and gradient explosion, the effect of long-term prediction is poor with traditional RNN. The LSTM model and the GRU model are variants of the RNN by using gated mechanism to maintain long-term information. So good results have been obtained in long-term prediction. The basic principles of the LSTM and the GRU are substantially the same. However, due to the complex structure of LSTM, the

training time is longer, while the GRU model has a relatively simple structure, fewer parameters and faster training ability. Therefore, we chose the GRU model to obtain temporal correlation from traffic data.

The GRU not only captures the current traffic information, but also retains the changing trend of historical traffic information and has the ability to better capture temporal correlation. The GRU obtains the traffic information of time t by taking the hidden state of time $t-1$ and the current traffic information as input. As shown in Fig. 5, h_{t-1} indicates the hidden state at $t-1$, r_t is the reset gate, used to control the degree of ignoring the state information of the previous moment; u_t is the update gate, used to control the state information of the last moment into the current state; c_t is the storage content of t time storage; h_t is the output state at time t.

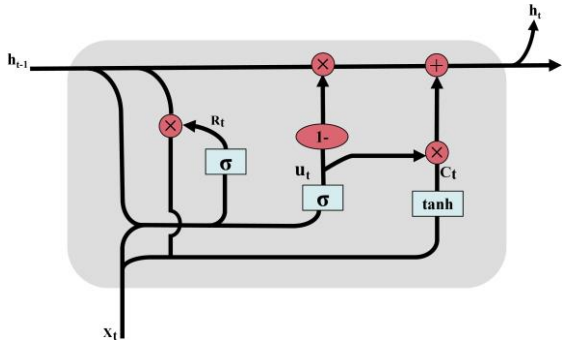

Fig. 5 The architecture of the GRU model

The specific calculation process is as shown in Eq. (10)-(13):

$$u_t = \sigma(W_u[X_{out}, h_{t-1}] + b_u) \quad (10)$$

$$r_t = \sigma(W_r[X_{out}, h_{t-1}] + b_r) \quad (11)$$

$$c_t = \sigma(W_c[X_{out}, (r_t \times h_{t-1})] + b_c) \quad (12)$$

$$h_t = u_t \times h_{t-1} + (1 - u_t) \times c_t \quad (13)$$

X_{out}represents the output of M-GCN and is defined in Eq. (9). W and b represent the weights and biases in the training process.

The ST-CT model is capable of handling complex spatio-temporal data. The local information enhancement module improves the capture of local spatio-temporal information while capturing the global correlation by using a combination of CNN and transformer encoder layer. The M-GCN captures spatial information better by learning location representations. The GRU is used to capture the dynamic changes of traffic volume on the road in order to obtain temporal correlation.

IV. EXPERIMENTS

4.1 Data Description

We evaluated the effect of the model on two real datasets, the SZ-taxi dataset and the Los-loop dataset. The both datasets are related to traffic speed.

(1) SZ-taxi. The dataset includes taxi operation data in Shenzhen from January 1 to January 31, 2015. 156 main roads in Luohu District were selected as the research area. The experimental data mainly consists of two parts. One is the 156 by 156 adjacency matrix, which describes the spatial relationships between roads. Each row represents one path, and the values in the matrix represent the connectivity between the paths. The other is the feature matrix, which describes the speed over time on each road. Each column represents the speed of traffic on the road at a different time. The speed of traffic on each road was aggregated every 15 minutes.

(2) Los-loop. This data set was collected in real time by a loop detector on freeways in Los Angeles County. We selected 207 sensors and their traffic speeds from March 1 to March 7, 2012. The speed of traffic was aggregated every five minutes. The data consists of an adjacency matrix and an feature matrix. The adjacency matrix is calculated according to the distance between sensors in the traffic network.

In the experiment, the input data was normalized to the interval [0,1]. In addition, 80% of the data is used as a training set and the remaining 20% as a test set. We predicted traffic speeds for the next 15 minutes, 30 minutes, 45 minutes, and 60 minutes.

4.2 Evaluation Metrics

We use four metrics to evaluate the prediction performance of model:

(1) Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N}\left(y_i^j - \hat{y}_i^j\right)^2} \quad (14)$$

(2) Mean Absolute Error (MAE):

$$MAE = \frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N}\left|\left(y_i^j - \hat{y}_i^j\right)\right| \quad (15)$$

(3) Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{j=1}^{M}\sum_{i=1}^{N}\left(y_i^j - \hat{y}_i^j\right)^2}{\sum_{j=1}^{M}\sum_{i=1}^{N}\left(y_i^j - \bar{Y}\right)^2} \quad (16)$$

(4) Explained V ariance Score (var):

$$var = 1 - \frac{Var\{Y - \hat{Y}\}}{Var\{Y\}} \quad (17)$$

where y_i^j and $\hat{y}_i^j$ represent the real traffic information and predicted one of the jth time sample in the ith

road. M is the number of time samples; N is the number of roads; Y and $\hat{Y}$ represent the set of y_i^j and $\hat{y}_i^j$ respectively, and $\bar{Y}$ is the average of$Y$.

Specifically, RMSE and MAE are used to measure prediction errors: the smaller the value, the better the prediction. R^2 and var calculate the correlation coefficient, which measures the ability of predictions to represent actual data: the larger the value, the better the prediction.

4.3 Choosing Model Parameters

In this section, the parameters of the model were introduced.

The hyperparameters of the model mainly include learning rate, batch size and the number of local information enhancement module. In the experiment, we manually adjusted and set the learning rate to 0.003, the batch size to 64, the number of local information enhancement modules are 5.

The training data set (accounting for 80% of the total data) is taken as the input in the training process, and the rest of the data is taken as the input in the testing process. The model was trained using the ADAM optimizer.

4.4 Baseline Methods

To verify the validity of this model, it was compared with some traditional and representative methods.

(1) ARIMA [4], which conducts parameter model fitting on observed time series to predict future traffic data.

(2) Support Vector Regression model (SVR) [34], which uses historical data to train the model in order to determine the relationship between input and output and then predicts future traffic data based on the trained model. The kernel we used in this model is a linear kernel.

(3) Graph Convolutional Network model (GCN) [14], the model is trained using historical data and traffic topology maps to learn the spatial information of the data to predict future traffic data.

(4) Gated Recurrent Unit model (GRU) [12], input historical data into the model, and the model captures the temporal correlation of the data to make predictions on the data.

(5) T-GCN [23], combines GCN and GRU to carry out traffic prediction.

(6) NA-DGRU [35], extracts spatial features from the neighborhood space of the road by using the neighborhood aggregation method.

4.5 Experimental Results

Table 1. The prediction results on SZ-taxi and Los-loop datasets

Data	Models	15 min				30 min				45 min				60 min			
		RMSE	MAE	R^2	var	RMSE	MAE	R^2	var	RMSE	MAE	R^2	var	RMSE	MAE	R^2	var
SZ-taxi	ARIMA	7.2406	4.9824	*	0.0035	6.7899	4.6765	*	0.0081	6.7852	4.6734	*	0.0087	6.7708	4.6655	*	0.0111
	SVR	4.1455	2.6233	0.8423	0.8424	4.1628	2.6875	0.8410	0.8413	4.1885	2.7539	0.8391	0.8397	4.2156	2.7751	0.8370	0.8379
	GCN	5.6596	4.2367	0.6654	0.6655	5.6918	4.2647	0.6616	0.6617	5.7142	4.2844	0.6589	0.6590	5.7361	4.3034	0.6554	0.6554
	GRU	3.9994	2.5955	0.8329	0.8329	4.0942	2.6906	0.8429	0.8450	4.1534	2.7743	0.8198	0.8199	4.0747	2.7712	0.8266	0.8267
	T-GCN	**3.9265**	2.7117	0.8541	0.8541	3.9663	2.7410	0.8456	0.8457	3.9859	2.7612	0.8441	0.8441	4.0048	2.7889	0.8422	0.8423
	NA-DGRU	4.0587	2.7387	*	*	4.0683	2.7280	*	*	4.0777	2.7393	*	*	4.0851	2.7487	*	*
	ST-CT	3.9612	**2.5947**	**0.8561**	**0.8562**	**3.9651**	**2.6319**	**0.8558**	**0.8558**	**3.9768**	**2.6200**	**0.8550**	**0.8551**	**3.9965**	**2.6410**	**0.8536**	**0.8536**
Los-loop	ARIMA	10.0439	7.6832	*	*	9.3450	7.6891	*	*	10.0508	7.6924	*	*	10.0538	7.6952	*	*
	SVR	6.0084	3.7285	0.8123	0.8146	6.9588	3.7248	0.7492	0.7523	7.7504	4.1288	0.6899	0.6947	8.4388	4.5036	0.6336	0.5593
	GCN	7.7922	5.3525	0.6843	0.6844	8.3353	5.6118	0.6402	0.6404	8.8036	5.9534	0.5999	0.6001	9.2657	6.2892	0.5583	0.5593
	GRU	5.2182	3.0602	0.8576	0.8577	6.2802	3.6505	0.7957	0.7958	7.0343	4.0915	0.7446	0.7451	7.6621	4.5186	0.6980	0.6984
	T-GCN	5.1264	3.1802	0.8634	0.8634	6.0598	3.7466	0.8098	0.8100	6.7065	4.1158	0.7679	0.7684	7.2677	4.6021	0.7283	0.7290
	NA-DGRU	5.1348	3.0281	*	*	6.1358	3.6692	*	*	6.7604	4.0567	*	*	7.2776	4.4256	*	*
	ST-CT	**4.9997**	**2.7068**	**0.8700**	**0.8705**	**5.9821**	**3.0979**	**0.8146**	**0.8152**	**6.6636**	**3.3897**	**0.7709**	**0.7719**	**7.2038**	**3.6614**	**0.7352**	**0.7359**

(1) High Prediction Precision.

Table 1 shows the performance of the model and other baseline methods on the 15-minute, 30-minute, 45-minute, and 60-minute prediction tasks on the SZ-taxi and Los-loop datasets. * indicates that these values are small and can be ignored, or the evaluation metrics that are not calculated in the baseline. From Table 1 we can see that the neural network-based methods (e.g., ST-CT and GRU) generally have better prediction precision than other baselines, and the prediction results of traditional time series analysis methods and machine learning methods are often unsatisfactory. For example, in the prediction of Los-loop data set, for the 15-minute traffic prediction task, the RMSE errors and MAE errors of the ST-CT model are reduced by approximately 50.22% and 64.76% compared with the ARIMA model. The RMSE errors and MAE errors of the ST-CT model are reduced by approximately 16.78% and 27.40% compared with the SVR model. This is because traditional time series analysis methods have difficulty in learning the non-linearity of traffic data, while machine learning methods also have difficulty in achieving good results with large data volumes. Meanwhile these models cannot capture spatial-temporal correlations. Therefore, these methods have limited capability compared to deep learning methods. Moreover, ARIMA gains by averaging the errors of different sections. Traffic data is dynamic. The data of some sections might greatly fluctuate to increase the final error. So, the ARIMA prediction effect is the worst.

(2) More accurately spatial-temporal forecasting ability.

Table 1 demonstrates that the spatial-temporal correlation-based methods (T-GCN) have better forecasting results than single-temporal or single-spatial (GRU, GCN) based methods. This is because single-temporal or single-spatial (GRU, GCN) methods can either only learn the spatial information of the data or only learn the temporal information of the data. But T-GCN learns the spatial information of the data as well as the temporal information of the data, so the results are better than single-temporal or single-spatial (GRU, GCN) methods. Our model obtains global and local correlation of data on the basis of capturing spatial-temporal correlation, and gets better results. For example, in the prediction of Los-loop data set, for the 15-minute traffic prediction task, the RMSE errors and MAE errors of the ST-CT model are reduced by approximately 38.29% and 49.43% compared with the GCN model. The RMSE errors and MAE errors of the ST-CT model are reduced by approximately 4.18% and 11.55% compared with the GRU model. The RMSE errors and MAE errors of the ST-CT model are reduced by approximately 2.47% and 14.88% compared with the T-GCN model. The RMSE errors and MAE errors of the ST-CT model are reduced by approximately 2.63% and 10.61% compared with the NA-DGRU model. Results based on SZ-taxi are similar to those based on Los-loop. Our proposed framework now achieves the best performance in almost all evaluation metrics on both datasets. In other words, after learning the global and local correlation of the data, our model can better capture the spatial topology characteristics of an urban road network as well as the temporal variation characteristics of the traffic state, while outperforming T-GCN at various prediction levels. Therefore, our model has more accurate space-time prediction ability.

(3) Long-Term Prediction Capability.

In all the prediction range, ST-CT has the best prediction effect. Fig. 6 shows the changes of MAE and *var* in different prediction horizons. It can be seen from Fig. 6 that the change trend of the prediction results of the model is small and has certain stability. It shows that our method is not sensitive to the prediction level. Therefore, our model can be used not only for short-term prediction, but also for long-term prediction.

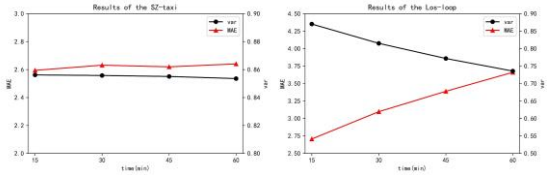

Fig. 6 Long-term prediction ability

The ST-CT model consistently yielded better results regardless of the prediction range. The ST-CT model can capture and analyze the spatial and temporal correlation of road traffic information and capture the global and local correlation of traffic speed information, and predict the changing trend of road traffic information. In addition, the model helps us to determine the start or end of peak periods by predicting the actual speed of traffic. This helps us to alleviate traffic congestion and other traffic problems.

4.6 Ablation Studies

In order to quantitatively verify the validity of the model design, ablation experiments for the local information enhancement module is conducted.

ST-C is the local information enhancement module to remove transformer encoder layer, ST-T is the local information enhancement module to remove the CNN and transpose convolutional neural network, the results are shown in Table 2:

Table 2. Ablation experiment results

T	Metric	SZ-taxi			Los-loop		
		ST-C	ST-T	**ST-CT**	ST-C	ST-T	**ST-CT**
15 min	RM SE	4.05 23	4.03 64	**3.96 12**	5.14 30	5.08 22	**4.99 97**
	MA E	2.70 99	2.63 91	**2.59 47**	2.77 43	2.73 03	**2.70 68**
	R^2	0.84 93	0.85 06	**0.85 61**	0.86 25	0.86 58	**0.87 00**
	var	0.84 86	0.85 06	**0.85 62**	0.86 34	0.86 65	**0.87 05**
30 min	RM SE	4.06 98	4.02 83	**3.96 51**	6.18 54	6.12 58	**5.98 21**
	MA E	2.73 81	2.64 73	**2.63 19**	3.19 84	3.15 19	**3.09 79**
	R^2	0.84 81	0.85 14	**0.85 58**	0.80 19	0.80 56	**0.81 46**
	var	0.84 81	0.85 26	**0.85 58**	0.80 30	0.80 62	**0.81 52**
45 min	RM SE	4.06 07	4.01 03	**3.97 68**	6.96 09	6.79 44	**6.66 36**
	MA E	2.71 62	2.63 01	**2.62 00**	3.57 36	3.43 77	**3.38 97**
	R^2	0.84 88	0.85 26	**0.85 50**	0.75 00	0.76 19	**0.77 09**
	var	0.84 88	0.85 27	**0.85 51**	0.75 10	0.76 32	**0.77 19**
60 min	RM SE	4.06 71	4.04 02	**3.99 65**	7.60 21	7.45 70	**7.20 38**
	MA E	2.72 64	2.64 30	**2.64 10**	3.86 14	3.73 65	**3.66 14**
	R^2	0.84 83	0.85 03	**0.85 36**	0.70 30	0.71 43	**0.73 52**
	var	0.84 84	0.85 04	**0.85 36**	0.70 45	0.71 61	**0.73 59**

The comparison of the ablation experimental results of RMSE, MAE, R^2, var for the dataset at prediction times of 15 min, 30 min, 45 min, and 60 min. According to the experimental results, we can see that after removing the CNN and transposed convolutional neural network in the local information enhancement module, the MAE and RMSE have increased compared with ST-CT, and R^2 and var have also decreased. After removing the transformer encoder layer in the local information enhancement module, the RMSE and MAE have also increased compared with ST-CT, and R^2 and var results decreased. We visualize the effect of the ablation experiment to make it more intuitive, as shown in Figs. 7-10. Based on the results of our ablation experiment, the effectiveness of our local information enhancement module can be demonstrated.

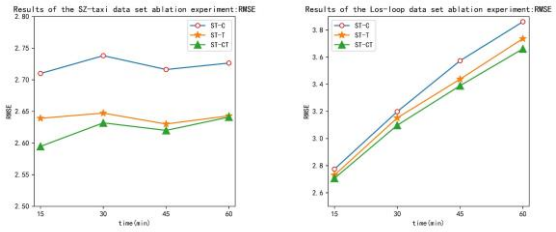

Fig. 7 Results of ablation experiments: RMSE

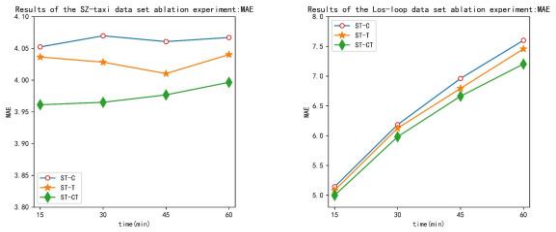

Fig. 8 Results of ablation experiments: MAE

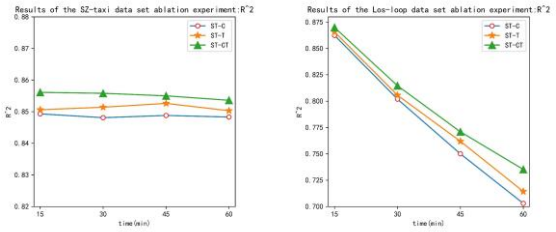

Fig. 9 Results of ablation experiments:R^2

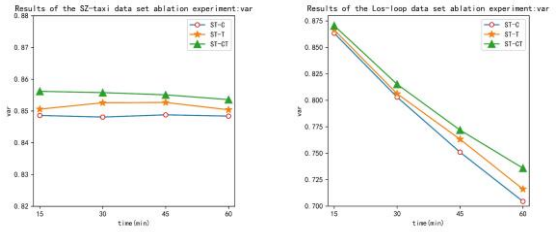

Fig. 10 Results of ablation experiments: var

We also looked into the effect of the number of local information enhancement modules on the model. As shown in Table 3, the experimental results of 4 local information enhancement modules (convolution kernel are 7, 5, 3, 1, respectively) and 6 local information enhancement modules (convolution kernel are 11, 9, 7, 5, 3, 1, respectively) were verified.

Table 3. The influence of the number of local information enhancement modules

T	Metric	SZ-taxi			Los-loop		
		n=4	n=5	n=6	n=4	n=5	n=6
15min	RMSE	3.99	**3.96**	3.98	5.06	**4.99**	5.04
		33	**12**	10	54	**97**	37
	MAE	2.62	**2.59**	2.61	2.72	**2.70**	2.73
		75	**47**	87	43	**68**	43
	R^2	0.85	**0.85**	0.85	0.86	**0.87**	0.86
		38	**61**	47	67	**00**	78
	var	0.85	**0.85**	0.85	0.86	**0.87**	0.86
		38	**62**	47	77	**05**	84
30min	RMSE	3.97	**3.96**	3.97	6.01	5.98	**5.98**
		69	**51**	52	02	21	**04**
	MAE	2.63	**2.63**	2.63	3.10	**3.09**	3.14
		41	**19**	73	24	**79**	22
	R^2	0.85	**0.85**	0.85	0.81	**0.81**	0.81
		50	**58**	51	29	**46**	08
	var	0.85	**0.85**	0.85	0.81	**0.81**	0.811
		50	**58**	53	37	**52**	6
45min	RMSE	3.99	**3.97**	3.97	6.84	**6.66**	6.59
		39	**68**	76	94	**36**	09
	MAE	2.61	**2.62**	2.63	3.50	**3.38**	3.42
		41	**00**	63	43	**97**	82
	R^2	0.85	**0.85**	0.85	0.75	0.77	**0.77**
		37	**50**	47	82	09	**58**
	var	0.85	**0.85**	0.85	0.75	0.77	**0.77**
		37	**51**	51	99	19	**68**
60min	RMSE	4.00	**3.99**	4.00	7.39	**7.20**	7.22
		60	**65**	50	46	**38**	07
	MAE	2.64	**2.64**	2.63	3.73	**3.66**	3.74
		19	**10**	84	88	**14**	09
	R^2	0.85	**0.85**	0.85	0.71	**0.73**	0.73
		29	**36**	29	89	**52**	22
	var	0.85	**0.85**	0.85	0.72	**0.73**	0.73
		29	**36**	29	04	**59**	40

As shown in Table 3, n represents the number of local information enhancement modules. The effect of 4 and 6 local information enhancement modules, respectively, in the SZ-taxi data set is not as good as the initial set of 5. Only three indicators are higher when the number of local information enhancement modules is 6 in the Los-loop data set than when the local information enhancement module is 5, because when the local information enhancement module is 4, less local information is learned than when the local information enhancement module is 5, so the effect is worse. Although more local information is learned when the local information enhancement module is set at 6, as the number of network layers deepens, some information is inevitably lost, resulting in a worse effect than when the local information enhancement module is set at 5. Experiments, on the other hand, show that the deeper the model layer, the better the effect is not always.

V. Conclusion

A deep learning-based modified traffic volume prediction model called ST-CT is proposed in the paper. By using this model, the spatial-temporal correlation and the global and local correlation of the traffic data can be better captured simultaneously. Therefore, the traffic prediction performance of the model is improved. Specifically, the data is inputted into the local information enhancement unit, and the local information enhancement unit captures the global and local information of the data through the combination of CNN with different convolution kernels and attention mechanism. The M-GCN is used to capture the spatial correlation of traffic data by learning the location representations of each node. The GRU is used to capture the temporal correlation by using gated mechanisms. Finally, the ST-CT model is used to solve the problem of spatio-temporal traffic prediction. It is tested on two real traffic datasets, and compared with HA, ARIMA, SVR, GCN, GRU and T-GCN, the ST-CT model achieves better performance at different prediction levels, demonstrating its utility in traffic prediction.

References

[1] D.W. Xu, H.H. Dong, H.J. Li, L.M. Jia, Y.J. Feng, The estimation of road traffic states based on compressive sensing, Transportmetrica B: Transport Dynamics, 3 (2015) 131-152.

[2] S.Y. Chang, H.C. Wu, Y.C. Kao, Tensor Extended Kalman Filter and Its Application to Traffic Prediction, IEEE Transactions on Intelligent Transportation Systems, (2023) 1-17.

[3] M.S. Ahmed, A.R.J.T.R.R. Cook, ANALYSIS OF FREEWAY TRAFFIC TIME-SERIES DATA BY USING BOX-JENKINS TECHNIQUES, 722 (1979).

[4] M.M. Hamed, H.R. Al-Masaeid, Z.M.B.J.J.o.T.E. Said, Short-Term Prediction of Traffic Volume in Urban Arterials, Journal of Transportation Engineering, 121 (1995) 249-254.

[5] I. Okutani, Y.J. Stephanedes, Dynamic prediction of traffic volume through Kalman

filtering theory, Transportation Research Part B: Methodological, 18 (1984) 1-11.

[6] W. Chun-Hsin, H. Jan-Ming, D.T. Lee, Travel-time prediction with support vector regression, IEEE Transactions on Intelligent Transportation Systems, 5 (2004) 276-281.

[7] Z. Yao, C.F. Shao, Y.L. Gao, Research on methods of short-term traffic forecasting based on support vector regression, Journal of Beijing Jiaotong University, (2006) 19-22.

[8] G. Lin, A. Lin, D. Gu, Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient, Information Sciences, 608 (2022) 517-531.

[9] S. Shiliang, Z. Changshui, Y. Guoqiang, A bayesian network approach to traffic flow forecasting, IEEE Transactions on Intelligent Transportation Systems, 7 (2006) 124-132.

[10] J.L.J.M.L. Elman, Distributed Representations, Simple Recurrent Networks, And Grammatical Structure, Machine learning, 7 (1991) 195-225.

[11] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation, 9 (1997) 1735-1780.

[12] K. Cho, B.v. Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, in: Conference on Empirical Methods in Natural Language Processing, 2014.

[13] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, T. Li, Predicting citywide crowd flows using deep spatio-temporal residual networks, Artificial Intelligence, 259 (2018) 147-166.

[14] J. Bruna, W. Zaremba, A. Szlam, Y. Lecun, Spectral Networks and Locally Connected Networks on Graphs, arXiv preprint arXiv:1312.6203, (2013).

[15] P. Fang, J. Zhou, S.K. Roy, P. Ji, L. Petersson, M. Harandi, Attention in Attention Networks for Person Retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (2022) 4626-4641.

[16] 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 (2017) 818.

[17] M. Cao, V.O.K. Li, V.W.S. Chan, A CNN-LSTM Model for Traffic Speed Prediction, in: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1-5.

[18] H. Yu, Z. Wu, S. Wang, Y. Wang, X. Ma, Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks, Sensors, 17 (2017).

[19] Y. Li, S. Chai, Z. Ma, G. Wang, A Hybrid Deep Learning Framework for Long-Term Traffic Flow Prediction, IEEE Access, 9 (2021) 11264-11271.

[20] K. Liu, Y. Zhang, X. Zhang, W. Qiao, P. Dong, Network Traffic Classification Based on LSTM+CNN and Attention Mechanism, in, 2023, pp. 545-556.

[21] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.K. Wong, W.-c. Woo, Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting, Advances in neural information processing systems, 28 (2015).

[22] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, X. Zhou, LC-RNN: A Deep Learning Model for Traffic Speed Prediction, in: International Joint Conference on Artificial Intelligence, 2018, pp. 27.

[23] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction, IEEE Transactions on Intelligent Transportation Systems, 21 (2020) 3848-3858.

[24] M. Gori, G. Monfardini, F. Scarselli, A new model for learning in graph domains, in: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., 2005, pp. 729-734.

[25] H. Peng, H. Wang, B. Du, M.Z.A. Bhuiyan, H. Ma, J. Liu, L. Wang, Z. Yang, L. Du, S. Wang, P.S. Yu, Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting, Information Sciences, 521 (2020) 277-290.

[26] C. Zheng, X. Fan, S. Pan, H. Jin, Z. Peng, Z. Wu, C. Wang, P.S. Yu, Spatio-Temporal Joint Graph Convolutional Networks for Traffic Forecasting, IEEE Transactions on Knowledge and Data Engineering, (2023) 1-14.

[27] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph WaveNet for Deep Spatial-Temporal Graph Modeling, 2019.

[28] K. Guo, Y. Hu, Z. Qian, H. Liu, K. Zhang, Y. Sun, J. Gao, B. Yin, Optimized Graph Convolution Recurrent Neural Network for Traffic Prediction, IEEE Transactions on Intelligent Transportation Systems, 22 (2021)

1138-1149.

[29] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting, 2020.

[30] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, Traffic Flow Prediction via Spatial Temporal Graph Neural Network, 2020.

[31] C. Liu, X. Liu, D.W.K. Ng, J. Yuan, Deep Residual Learning for Channel Estimation in Intelligent Reflecting Surface-Assisted Multi-User Communications, IEEE Transactions on Wireless Communications, 21 (2022) 898-912.

[32] X. Wang, M. Tang, T. Yang, Z. Wang, A novel network with multiple attention mechanisms for aspect-level sentiment analysis, Knowledge-Based Systems, 227 (2021) 107196.

[33] J. Ba, J.R. Kiros, G.E.J.A. Hinton, Layer Normalization, arXiv preprint arXiv:1607.06450, abs/1607.06450 (2016).

[34] A. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing, 14 (2004) 199-222.

[35] T. Xiaoping, C. Zou, Y. Zhang, L. Du, S. Wu, NA-DGRU: A Dual-GRU Traffic Speed Prediction Model Based on Neighborhood Aggregation and Attention Mechanism, Sustainability, 15 (2023) 2927.