2023 4th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC)

Traffic Spatial-Temporal Transformer for Traffic Prediction

Zhiyang He
School of Cyberspace Security
Dongguan University of Technology
Dongguan, China
hezhiyang668@163.com

* Ye Ding
School of Cyberspace Security
Dongguan University of Technology
Dongguan, China
dingye@dgut.edu.cn

***Abstract*—Accurate traffic prediction can help administrators better plan and manage urban traffic, alleviating the traffic pressure. Local spatial-temporal dependency is the strongest and most direct dependency within traffic data. However, recent researches in traffic prediction using stacked or coupled fusion methods to combine temporal and spatial learning networks have not fully captured local spatial-temporal dependency in traffic data. This paper introduces a recurrent neural network structure that captures local spatial-temporal dependency by considering the spatial relationship of each time with its current, past, and future time simultaneously. Additionally, a period enhanced attention mechanism is introduced to capture long-term temporal dependency. Finally, the two modules are combined to construct a Traffic Spatial-Temporal Transformer for traffic prediction. Experimental results demonstrate that the proposed transformer outperforms baselines in terms of prediction accuracy.**

Keywords- traffic prediction; spatial-temporal dependency

I. INTRODUCTION

With the development of intelligent sensors and urban computing, Intelligent Transportation Systems (ITS) can collect and analyze voluminous traffic data generated by the busy traffic to extract valuable information. As in [1], ITS can help administrators make informed decisions for rational urban traffic planning and effective management to alleviate traffic pressure. Traffic prediction is essential in ITS, forecasting future traffic conditions from historical data, which involves temporal and non-Euclidean spatial dependencies.

Taking the two dependencies into consideration, the traffic state of a road in the road network at a time is related to its own and the connected roads' state at the current, past, and future time, which is called local spatial-temporal dependency, as shown in Fig. 1. Local spatial-temporal dependency is the most direct and strongest dependency present in traffic data, and it is also the most critical dependency that needs to be captured for traffic prediction.

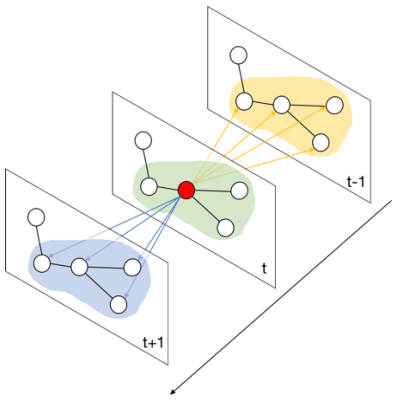

Figure 1. The local spatial-temporal dependency of traffic data.

Recent researches on traffic prediction mainly focused on the network structures that fuse temporal learning network (TLN) and spatial learning network (SLN). According to [2], the fusion methods in these network structures can be divided into two categories: stacked fusion and coupled fusion. Stacked fusion connects TLN and SLN serially or in parallel, capturing temporal and spatial dependencies separately, such as TGCN [3], STGCN [4], ASTGCN [5] and GWN [6]. This fusion method leads to each network neglecting other dependency while capturing their respective dependency and fails to capture the local spatial-temporal dependency. The coupled fusion method is usually embedding GNN-based SLN into RNN-based TLN, capturing spatial dependency in each iteration, such as DCRNN [7] and DGCRN [8]. This method can capture the spatial-temporal dependency between the present and past time, but without the future time. BiLSTM [9] can address this issue by processing bidirectionally but with high costs. S-LSTM [10], a recurrent network structure with a message-passing mechanism that treats the entire time series as a hidden state with global state, providers lower cost and more accurate temporal feature extraction compared to BiLSTM.

To achieve more accurate traffic prediction, we enhanced S-LSTM with a dynamic graph generation process during each iteration and utilized it for graph convolution during hidden state updates, which allows us to capture local spatial-temporal dependency in traffic data effectively. Additionally, we designed an enhanced attention mechanism based on period to capture long-term temporal dependency. This mechanism decomposes time series into the frequency domain to obtain periods and strengthens the attention between each time point and its related periodic time points based on frequency. Our contributions are summarized as follows:

- We propose a recurrent network structure named LSTN to simultaneously capture the spatial and temporal dependencies of each road at each time along with the neighbor roads at current, past and future time.
- We propose an enhanced attention named PEA, which enhanced the attention of each time to its periodic correlated time.

2023 4th International Symposium on Computer Engineering and Intelligent Communications (ISCEIC) | 979-8-3503-0632-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISCEIC59030.2023.10271152

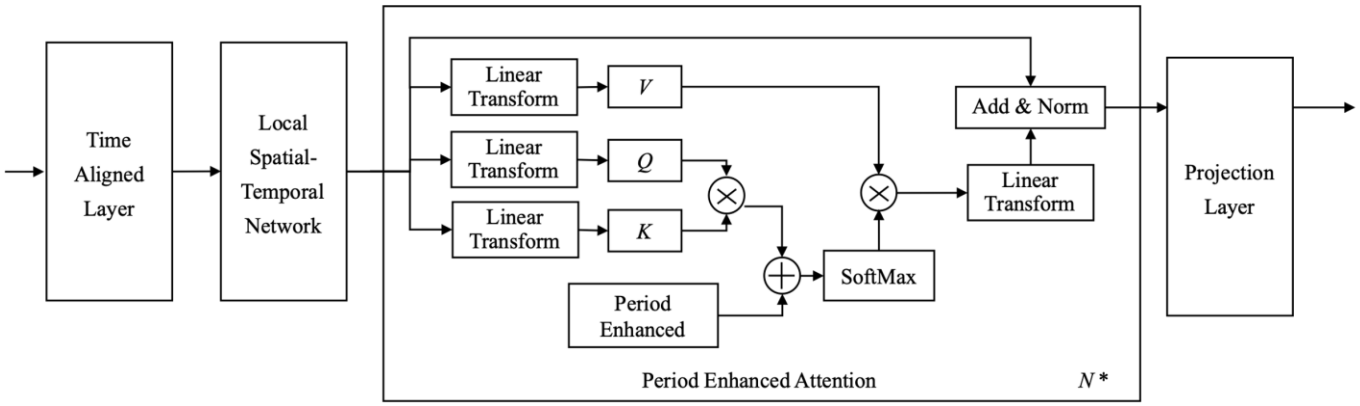

Figure 2. The overall architecture of TTST

- We propose a transformer called TSTT that combines LSTN and PEA for traffic prediction. We conducted experiments based on two real-world traffic datasets, and the results show that TTST outperforms baselines.

II. Related Work

Early researches treated traffic prediction as a time series forecasting problem, only focusing on capturing the temporal dependency in traffic data. In past decade, due to the development of Convolution Neural Network (CNN) in area of computer vision, some researches such as ConvLSTM [11] and PredRNN [12], have applied CNN to catch the Euclidean spatial dependency. In recent years, with the development of Graph Neural Network, especially GCN [13], many researches have increasingly integrated TLN with GCN to effectively capture the spatial-temporal dependency in traffic data.

TGCN [1] stacks GCN and GRU to separately capture time dependency and spatial dependency. STGCN [2] combines one-dimensional convolution and gate mechanism for time dependency, and stacks it with GCN to capture spatial dependency. ASTGCN [3] extracts adjacent, daily, and weekly time segments, computes temporal and spatial attention for each of these three groups of data, and then combines GCN with one-dimensional convolution to capture the spatial-temporal dependency. GWN [4] proposes an adaptive adjacency matrix learning approach that combines GCN to capture spatial dependency while utilizing gated TCN to capture temporal dependency. DCRNN [5] introduces a bidirectional random walk graph diffusion convolutional recurrent network and employs seq2seq architecture for modeling spatial-temporal dependency. DGCRN [6] combines GCN with RNN and uses a hyper-network to generate an dynamic graph before each recurrent of RNN to capture dynamic changes of traffic data. Based on the generated dynamic graph and the original adjacent matrix, GCN can capture more spatial information.

III. Preliminaries

A. Traffic data

We define the traffic data as $D = (V, A, X_t)$. Here, V is a set with $N = |V|$ vertices, where each $v \in V$ represents a traffic sensor or road segment in the traffic network; A is the adjacent matrix, where each $a_{ij} \in A$ represents a connection from v_i to v_j in the traffic network when it equals 1; $X_t = (x_t^1, x_t^2, ..., x_t^n)$ is the traffic state, where x_t^i represents the state collecting from v_i at time t.

B. Problem formulation

The traffic prediction problem can be formulated as follows, given the historical data $X = (X_{t-h}, X_{t-h-1}, ..., X_t)$ of the past h time steps, predict the traffic state $Y = (Y_{t+1}, Y_{t+2}, ..., Y_{t+f})$ of the f future time steps.

IV. Methods

A. Overview

Fig. 2 shows the overall architecture of Traffic Spatial-Temporal Transformer (TSTT) including two main parts: Local Spatial-Temporal Network (LSTN) and Period Enhanced Attention (PEA). TSTT first applies a linear transformation layer to align the time dimension of the input from h to $h+f$, which can better learn the temporal variation of cross past and future. Then the aligned input is fed into LSTN to capture local spatial-temporal dependency. Next, the output of LSTN is further processed through PEA to capture long-term time dependency. Finally, PEA's output is projected from the time dimension of $h+f$ to f using a linear transformation layer, and the projected output serves as the prediction result of TTST.

B. LSTN

We propose LSTN by incorporating dynamic graph generation and GCN into S-LSTM. Fig. 3 shows the structure of LSTN. The hidden state H^k in k-th recurrence include the sub-hidden time state h_i^k at each time i and the global spatial state z^k, as in (1). Following S-LSTM, we initialize h_i^0 and z^0 in H^0 with the same parameter h^0. The transition from H^{k-1} to H^k involves the generation of dynamic graph g^k and the updates of h_i^k and z^k.

$$H^k = \{h_1^k, \ldots, h_t^k, z^k\} \tag{1}$$

Building upon the S-LSTM's process of extracting local temporal dependency based on each time state and its temporal context, LSTN integrates graph convolution into this process to capture local spatial-temporal dependency. GCN is typically based on the adjacency matrix in traffic data, which only records direct connections between roads. However, there are rich relationships between indirectly connected roads. To comprehensively consider the relationships between different

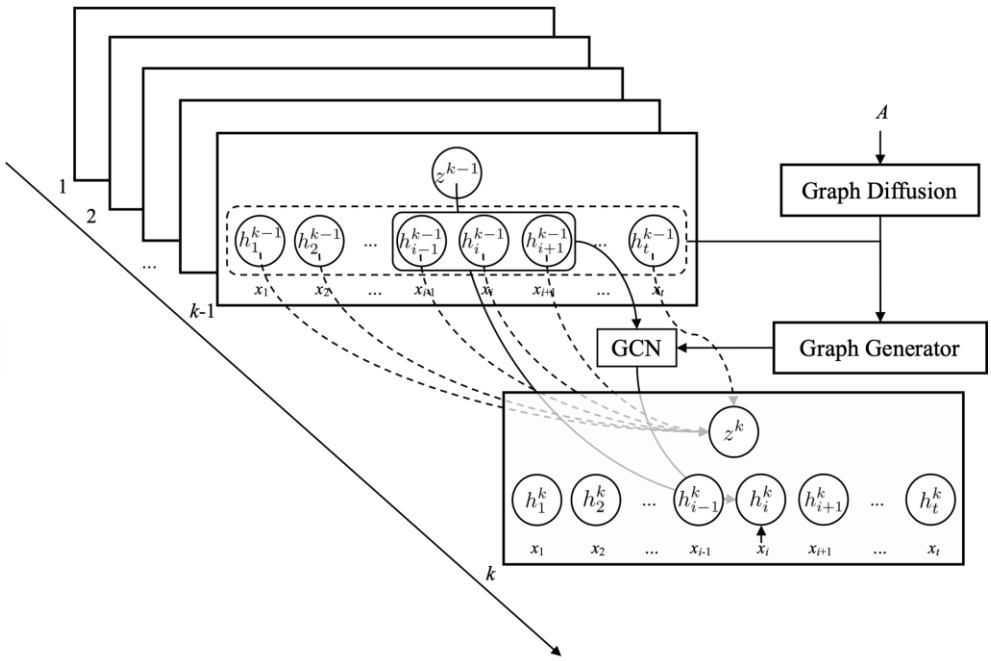

Figure 3. The structure of LSTN

roads, we perform graph diffusion process on the adjacency matrix A before the start of LSTN's recurrence to expand the connection between roads. The graph diffusion process is as shown in (2), where θ_a is a parameter, a is the diffusion step, D is the diagonal matrix of A and A' is the expanded adjacency matrix.

$$A' = \sum_{a=0}^{n} \theta_a (AD^{-1})^a \tag{2}$$

The generation of dynamic graph z^k in each recurrence of LSTN is to help GCN better capture the spatial dependency within the hidden states. The generation process first transposes $\{h_1^{k-1}, \ldots, h_t^{k-1}\} \in \mathbb{R}^{t*N}$ to $\{v_1^{k-1}, \ldots, v_N^{k-1}\} \in \mathbb{R}^{N*t}$ and then uses the method of computing attention scores in graph attention mechanism to calculate the correlation matrix of roads. Finally, A' is used to mask the correlation matrix to obtain the dynamic matrix g^k, as in (3).

$$g^k = \text{mask}(\text{AttScore}(v_1^{k-1}, \ldots, v_N^{k-1}), A') \tag{3}$$

Similar to S-LSTM, the calculation for updating h_i^k is based on x_i^k, h_{i-1}^{k-1}, h_i^{k-1}, h_{i+1}^{k-1} and z^{k-1}, as in the following equation group:

$$\mu_i^k = [h_{i-1}^{k-1}, h_i^{k-1}, h_{i+1}^{k-1}]$$
$$\hat{e}_i^k = \text{sigmod}(W_e \mu_i^k g^k + U_e x_i + V_e z^{k-1} + b_e)$$
$$l_i^k = \text{sigmod}(W_l \mu_i^k g^k + U_l x_i + V_l z^{k-1} + b_l)$$
$$r_i^k = \text{sigmod}(W_r \mu_i^k g^k + U_r x_i + V_r z^{k-1} + b_r)$$
$$f_i^k = \text{sigmod}(W_f \mu_i^k g^k + U_f x_i + V_f z^{k-1} + b_f)$$
$$\hat{s}_i^k = \text{sigmod}(W_s \mu_i^k g^k + U_s x_i + V_s z^{k-1} + b_s)$$
$$o_i^k = \text{sigmod}(W_o \mu_i^k g^k + U_o x_i + V_o z^{k-1} + b_o)$$
$$u_i^k = \tanh(W_u \mu_i^k g^k + U_u x_i + V_u z^{k-1} + b_u)$$
$$i_i^k, l_i^k, r_i^k, f_i^k, s_i^k = \text{softmax}(\hat{e}_i^k, l_i^k, r_i^k, f_i^k, \hat{s}_i^k)$$
$$c_i^k = l_i^k \odot c_{i-1}^{k-1} + f_i^k \odot c_i^{k-1} + r_i^k \odot c_{i+1}^{k-1} + s_i^k \odot c_z^{k-1} + e_i^k \odot u_i^k$$
$$h_i^k = o_i^k * \tanh(c_i^k) \tag{4}$$

where μ_i^k represents the concatenation of hidden states within a time context window and c_i^k denotes the cell state. i_i^k, l_i^k, r_i^k, f_i^k and s_i^k are the setting gates to control the information flow from the input x_i, the past time cell c_{i-1}^{k-1}, the future time cell c_{i+1}^{k-1}, c_i^{k-1}, and the global spatial cell c_z^{k-1} to c_i^k. o_i^k is an output gate from the cell state c_i^k to the hidden state h_i^k. W, U, V and b are the gate-related parameters of model.

Additionally, we have further improved the computation of global spatial state in S-LSTM by drawing inspiration from the method of channel attention calculation in SENet[14]. We performed average pooling on each channel of the sub-hidden time state followed by 1x1 one-dimensional convolution to obtain the global spatial feature of each road. The process of updating z^k is as in the following equation group:

$$\bar{h} = \text{Conv1d}(\text{AvgPool}(h_1^{k-1}, \ldots, h_t^{k-1}))$$
$$f_z^k = \text{sigmod}(W_z z^{k-1} + U_z \bar{h} + b_z)$$
$$f_i^k = \text{sigmod}(W_f z^{k-1} + U_f h_i^{k-1} + b_f)$$
$$o_i^k = \text{sigmod}(W_o z^{k-1} + U_o \bar{h} + b_o)$$
$$f_1^k, \ldots, f_t^k, f_z^k = \text{softmax}(f_1^k, \ldots, f_t^k, f_z^k)$$
$$c_z^k = f_z^h \odot c_z^{k-1} + \sum_{i=1}^{t} f_i^k \odot c_i^{k-1}$$
$$z^k = o^k \odot \tanh(c_z^k) \tag{5}$$

where $\bar{h}$ is the global spatial feature, $f_1^k, \ldots, f_t^k$ and f_z^k are gates controlling the information flow from $c_1^{k-1}, \ldots, c_t^{k-1}$ and c_z^{k-1}. o^k is an output gate from c_z^{k-1} to z^k.

C. PEA

Self-Attention can capture long-term temporal dependency by establishing weight relationships among different elements

in a time series. To further capture the long-term time dependency in traffic data and account for its periodicity, we propose PEA by enhancing the attention scores for related periodic time points within Self-Attention.

PEA consists of two parts: the period enhanced matrix generation and the self-attention. Fig. 4 shows the process of period enhanced matrix generation part, it starts by using Fast Fourier Transform (FFT) to obtain frequency intensity of the data, computing the average frequency intensity for each channel and selecting the top-d highest frequency intensity, as shown in (6), where f_d is the d-th highest frequency intensity.

$$\{f_1, \ldots, f_d\} = \text{Topd}(\text{Avg}(\text{FFT}(X))) \quad (6)$$

Then the periods are computed based on the historical data length T, as in (7), and the enhanced coefficients are obtained by applying LeakyReLU operation to frequency intensity, as in

TABLE I. THE RESULTS OF EXPERIMENT

Data	Model	3 steps (15 mins)			6 steps (30 mins)			12 steps (60 mins)		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
PeMS-Bay	STGCN	2.962	1.352	2.94%	3.942	1.728	3.94%	4.905	2.11	5.12%
	ASTGCN	3.217	1.396	3.08%	4.286	1.814	4.52%	4.984	2.213	5.15%
	GWN	2.812	1.349	2.81%	3.795	1.692	3.86%	4.618	2.025	4.79%
	DCRNN	2.854	1.372	2.87%	3.892	1.765	4.12%	4.811	2.143	5.09%
	DGCRN	2.69	1.28	2.66%	3.63	1.65	3.55%	4.42	1.89	4.43%
	TSTT	**2.651**	**1.275**	**2.64%**	**3.592**	**1.623**	**3.53%**	**4.218**	**1.756**	**4.39%**
METR-LA	STGCN	5.236	3.052	7.12%	6.212	3.476	8.67%	8.142	4.014	10.52%
	ASTGCN	5.512	3.127	7.21%	6.275	3.631	8.63%	8.256	4.132	10.66%
	GWN	5.374	2.935	7.02%	6.174	3.205	8.39%	7.933	3.768	10.34%
	DCRNN	5.059	2.962	7.08%	6.193	3.355	8.48%	8.122	3.855	10.39%
	DGCRN	5.01	2.62	6.63%	6.05	2.99	8.02%	7.19	3.44	9.73%
	TSTT	**4.965**	**2.596**	**6.57%**	**5.832**	**2.895**	**7.93%**	**7.094**	**3.245**	**9.45%**

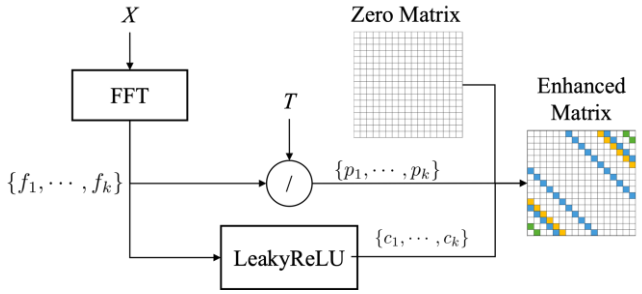

Figure 4. The process of period enhanced matrix generation.

(8), where p_d and c_d represents the period and enhanced coefficient for f_d.

$$\{p_1, \ldots, p_d\} = \{T/f_1, \ldots, T/f_d\} \quad (7)$$

$$\{c_1, \ldots, c_d\} = \text{LeakyReLU}(\{f_1, \ldots, f_d\}) \quad (8)$$

The enhanced matrix is generated by updating a zero matrix M based on the period and frequency intensity. The update process for each $m_{ij} \in M$ is as shown in (9). The detail of self-attention is as shown in (10) and (11), where W_q, W_k and W_v are the parameters of linear transformations, d_k is the dimension of key.

$$m_{ij} = \{c_k : (i - j) \bmod p_k == 0\} \quad (9)$$

$$Q, K, V = XW_q, XW_k, XW_v \quad (10)$$

$$\text{Attention}(Q, K, V) = \text{softmax}((QK^T + M) / (d_k)^{-1/2}) \quad (11)$$

V. Experiment

A. Setup

We employed two real-world datasets for conducting the experiment: METR-LA[7] and PeMS-Bay[15], and adopt the five following baselines to compare with our model: STGCN[4], ASTGCN[5], GWN[6], DCRNN[7], and DGCRN[8]. The following three metrics are used to evaluate the models:

- Mean Absolute Error (MAE):

$$\text{MAE} = (\sum_{i=1}^{m} |\hat{y}_i - y_i|) / m$$

- Root Mean Squared Error (RMSE):

$$\text{RMSE} = ((\sum_{i=1}^{m} (\hat{y}_i - y_i)^2) / m)^{-1/2}$$

- Mean Absolute Percent Error (MAPE):

$$\text{MAPE} = 100\% * (\sum_{i=1}^{m} |\hat{y}_i - y_i| / y_i) / m$$

where m is the number of samples in the test dataset, $\hat{y}_i$ and y_i are the predicted and true values of sample i.

B. Result

Table 1 shows the experimental results. The bold results indicate the best performance, and the underlined results represent the second-best. Overall, TTST performs better than other models in the three evaluated metrics. To be more specific, the advantage of TTST is more evident in long-term

predictions (12 steps) compared to short-term predictions (3 steps and 6 steps).

VI. Conclusions

In this paper, we fully consider the spatial-temporal dependency in traffic data and propose a transformer called TTST to achieve traffic prediction. TTST includes a recurrent neural network combined with GCN to capture the local spatial-temporal dependency, and a period enhanced attention to capture the long-range temporal dependency. We have tested TTST on two real-world datasets for traffic prediction task and the results showed TTST outperforms baselines with a greater advantage in long-term prediction compared to short-term prediction.

Acknowledgment

This work is supported by National Natural Science Foundation of China (U19A2067, 61976051).

References

[1] M. Shaygan, C. Meese, W. Li, X.G. Zhao, and M. Nejad, "Traffic prediction using artificial intelligence: review of recent advances and emerging opportunities," Transportation Research Part C: Emerging Technologies, vol. 145, p. 103921, 2022.

[2] G. Jin, Y. Liang, Y. Fang, J. Huang, J. Zhang, and Y. Zheng, "Spatio-temporal graph neural networks for predictive learning in urban computing: A survey," arXiv preprint arXiv:2303.14483, 2023.

[3] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 9, pp. 3848-3858, 2019.

[4] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," arXiv preprint arXiv:1709.04875, 2017.

[5] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 922-929, 2019.

[6] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph Wavenet for Deep Spatial-Temporal Graph Modeling," arXiv preprint arXiv:1906.00121, 2019.

[7] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting," arXiv preprint arXiv:1707.01926, 2017.

[8] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, and Y. Li, "Dynamic Graph Convolutional Recurrent Network for Traffic Prediction: Benchmark and Solution," ACM Transactions on Knowledge Discovery from Data, vol. 17, no. 1, pp. 1-21, 2023.

[9] M. Schuster and K.K. Paliwal, "Bidirectional Recurrent Neural Networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.

[10] Y. Zhang, Q. Liu, and L. Song, "Sentence-State LSTM for Text Representation," arXiv preprint arXiv:1805.02474, 2018.

[11] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in Advances in Neural Information Processing Systems (NIPS), vol. 28, 2015.

[12] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "PredRNN: Recurrent Neural Networks for Predictive Learning Using Spatiotemporal LSTMs," in Proceedings of the Neural Information Processing Systems (NIPS), vol. 30, 2017.

[13] T.N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," arXiv preprint arXiv:1609.02907, 2016.

[14] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132-7141, 2018.

[15] R. Jiang, D. Yin, Z. Wang, Y. Wang, J. Deng, H. Liu, Z. Cai, J. Deng, X. Song, and R. Shibasaki, "DL-Traff: Survey and Benchmark of Deep Learning Models for Urban Traffic Prediction," in Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 4515-4525, October 2021.