Clustering-property Matters: A Cluster-aware Network for Large Scale Multivariate Time Series Forecasting

Yuan Wang
Zezhi Shao
Tao Sun
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
{wangyuan21s,shaozezhi19b,suntao}@ict.ac.cn

Chengqing Yu
Yongjun Xu,
Fei Wang*
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
{yuchengqing22b,xyj,wangfei}@ict.ac.cn

ABSTRACT

Large-scale Multivariate Time Series (MTS) widely exist in various real-world systems, imposing significant demands on model efficiency. A recent work, STID, addressed the high complexity issue of popular Spatial-Temporal Graph Neural Networks (STGNNs). Despite its success, when applied to large-scale MTS data, the number of parameters of STID for modeling spatial dependencies increases substantially, leading to over-parameterization issues and suboptimal performance. These observations motivate us to explore new approaches for modeling spatial dependencies in a parameter-friendly manner. In this paper, we argue that the spatial properties of variables are essentially the superposition of multiple cluster centers. Accordingly, we propose a Cluster-Aware Network (CANet), which effectively captures spatial dependencies by mining the implicit cluster centers of variables. CANet solely optimizes the cluster centers instead of the spatial information of all nodes, thereby significantly reducing the parameter amount. Extensive experiments on two large-scale datasets validate our motivation and demonstrate the superiority of CANet.

CCS CONCEPTS

• **Information systems** → **Data mining**.

KEYWORDS

large-scale, multivariate time series forecasting, cluster centers

ACM Reference Format:
Yuan Wang, Zezhi Shao, Tao Sun, Chengqing Yu, Yongjun Xu, Fei Wang. 2023. Clustering-property Matters: A Cluster-aware Network for Large Scale Multivariate Time Series Forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615253

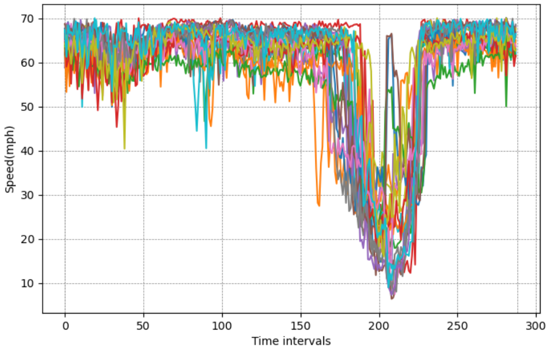

Figure 1: Traffic data (speed) over 24 hours of 20 sensors in the METR-LA dataset, which shares similar patterns.

1 INTRODUCTION

Multivariate time series (MTS) widely exist in the real world, creating massive demand for MTS forecasting techniques [16]. Spatial-Temporal Graph Neural Networks (STGNNs) [2, 7, 8, 14, 15, 20] have recently achieved State-Of-The-Art (SOTA) performance in MTS forecasting. Researchers adeptly model spatial dependencies between variables based on graph structure and graph convolution, significantly improving prediction accuracy. Despite their success, STGNNs usually suffer from efficiency issues and fail to handle large-scale MTS data. Their complexity increases quadratically with the number of variables, which is unacceptable for tens of thousands of variables. To solve the efficiency issues, a recent work, STID [12], identifies the indistinguishability of samples in both spatial and temporal dimensions as a critical bottleneck and introduces spatial identities in the spatial dimension. However, the number of spatial identities is equal to the variables, and they are optimized without specific constraints, leading to a sharp parameter increase and optimization issues in large-scale datasets simultaneously. Specifically, the numbers of spatial identities in large-scale datasets are hundreds more times than in commonly-used datasets [8, 20] and may easily cause over-parameterization issues.

This observation motivates us to explore new approaches for modeling spatial dependencies in a parameter-friendly manner. We argue that spatial identities of variables are essentially the superposition of multiple cluster centers. Accordingly, we can solely optimize limited cluster centers, notably reducing parameters for modeling spatial dependencies.

[*] Corresponding author.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615253

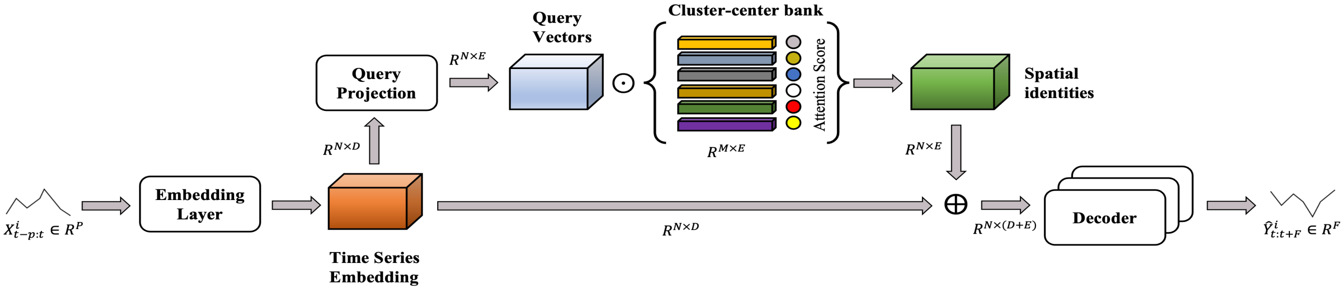

Figure 2: Architecture of our proposed framework. Cuboids in the cluster-center bank represent embeddings of cluster centers, and circles with different colors represent different attention scores.

To illustrate our motivation, we take the intersections in traffic systems as an example: intersections located in similar areas or playing similar roles usually have close traffic distribution. As shown in Figure 1, different time series reveal similar patterns over time, which indicates implicit clusters exist between variables, and the number of them is far less than the number of variables. Therefore, we can obtain variables' spatial identities by exploiting limited cluster centers, resulting in significant alleviation of parameters and optimization issues simultaneously.

In this paper, we propose a Cluster-Aware Network, named CANet, to generate spatial identities in a parameter-friendly manner, which includes a core component, cluster-aware(CA) module. CA module consists of a cluster-center bank to store the cluster centers' embeddings and an attention-based method to obtain the distribution of cluster centers for variables. To elaborate, firstly, the CA module uses time series embeddings to query all the cluster centers' embeddings and implements the weighted summation of them according to the results of queries. Compared with STGNNs and STID, exploiting hidden clusters among variables for modeling spatial dependencies reduces parameters significantly. Extensive experiment results show CANet achieves the best performance on datasets. Ablation study results further prove the effectiveness of the CA in modeling spatial dependencies, and visualization results suggest that our framework captures hidden clusters successfully.

2 PROBLEM DEFINITION

Multivariate Time Series Forecasting. Multivariate time series forecasting aims to simultaneously predict all the variables' future time series. We assume we receive all the time series as $X \in R^{N\times T}$ and predict their future data, where N is the number of variables, and T is the length of the time series. We define the input time series of variable i from past P time steps as $X^i_{t-P:t-1} \in R^P$ and the F future time series as $Y^i_{t:t+F-1} \in R^F$ respectively. We denote our prediction for variable i as $\hat{Y}^i_{t:t+F-1} \in R^F$.

3 MODEL ARCHITECTURE

3.1 Overview

Figure 2 shows that our model consists of a time series embedding layer, a cluster-aware module, and an MLP-based decoder. Firstly, the time series embedding layer maps time series to latent space, obtaining time series embeddings. Then, we use these embeddings as queries to implement weighted summation of cluster centers based on the attention mechanism [1], generating spatial identities of variables. Finally, time series embeddings and spatial identities are concatenated and decoded by the decoder to make a prediction.

3.2 Time Series Embedding layer

We use FC(·) to represent a fully connected (FC) layer for simplicity. The time series embedding layer consists of a FC layer to encode time series, mapping them from $X_{t-P:t-1} \in R^{N\times P}$ to $H_t \in R^{N\times D}$:

$$H_t = FC_{Embedding}(X_{t-P:t}), \tag{1}$$

where D is the hidden dimension of the embedding layer.

3.3 Cluster-aware Module

The cluster-aware module mainly consists of a cluster-center bank $\Theta \in R^{M\times E}$, where M is the number of cluster centers, and E is the dimension of a cluster center's embedding. It implements time series clustering based on classical attention mechanism [1].

Classical attention mechanism requires three vectors Q, K and V, which respectively correspond to Q_t, Θ, Θ in this work. Θ is initialized randomly and $Q_t \in R^{N\times E}$ is projected from H_t:

$$Q_t = QueryProjection(H_t) = FC_{proj}(H_t). \tag{2}$$

For numerical stability, we normalize the embeddings stored in the cluster-center bank by its $L2$ norm:

$$\hat{\Theta} = \frac{\Theta}{\|\Theta\|_2}. \tag{3}$$

Then, time series embeddings are used to query all the cluster centers and obtain weights measuring the relative distance between variables and them:

$$W_t = Softmax((Q_t)^T\hat{\Theta}), \tag{4}$$

where $W_t \in R^{N\times M}$, reflecting the relative distance between N variables and M cluster centers.

Finally, to generate spatial identities of variables, all the cluster centers are superposed according to the calculated weight:

$$S_t = W_t \cdot \hat{\Theta}, \tag{5}$$

where $S_t \in R^{N\times E}$, representing the spatial identities of N variables.

3.4 Time Series Decoder

Before making a prediction, our model attaches time series embeddings H_t and spatial identities S_t together:

$$Z_t^1 = H_t || S_t, \tag{6}$$

where $Z_t^1 \in R^{N\times(D+E)}$, representing spatial-temporal features.

Then, we use the time series decoder consisting of L fully connected layers to implement a prediction. The first $L-1$ layers further transform the spatial-temporal features Z_t^1 and the $l-th$ ($l = 1, 2, 3, ..., L-1$) layer's output Z_t^{l+1} can be denoted as:

$$(Z_t)^{l+1} = FC_2^l(\sigma(FC_1^l((Z_t)^l))) + (Z_t)^l. \tag{7}$$

Finally, $L-th$ layer conducts a prediction based on $(Z_t)^L$ by:

$$\hat{Y}_{t:t+F-1} = Decoder_L((Z_t)^L) = FC((Z_t)^L), \tag{8}$$

where $Z_t^L \in R^{D+E}$ and $\hat{Y}_{t:t+F} \in R^{N\times F}$ is the prediction.

We use Mean Average Error (MAE) to measure the deviation between prediction and ground truth:

$$\mathcal{L}_{MAE} = \frac{1}{NF}\sum_{i=1}^{N}\sum_{j=t}^{t+F-1}|\hat{Y}_j^i - Y_j^i|, \tag{9}$$

where $\hat{Y}_j^i$ and Y_j^i are variable i's prediction and ground truth in time step j, respectively.

Furthermore, to ensure the capacity of the cluster-aware module for differentiating cluster centers, inspired by [4, 11], we introduce two constraints, consistency and contrast loss:

$$\mathcal{L}_{consistency} = \sum_{i=1}^{N}||Q_t^i - \Theta[a]^i||^2, \tag{10}$$

where $Q_t^i \in R^E$ and $\Theta[a]^i \in R^E$ respectively represent the query from variable i and the closest cluster center for variable i.

$$\mathcal{L}_{contrast} = \sum_{i=1}^{N} max(||Q_t^i - \Theta[a]^i||^2 - ||Q_t^i - \Theta[b]^i||^2 + \lambda, 0), \tag{11}$$

where $\Theta[b]^i \in R^E$ represents the second closest cluster-center for variable i and $\lambda \in R$ denotes the margin between positive (the closest cluster center) and negative (the rest) pairs.

To summarize, consistency loss aims to avoid excessive dispersion and simultaneously preserve a certain level of discrimination between cluster centers. And contrary loss seeks to maximize the discrimination between variables as much as possible.

Therefore, our loss function is:

$$\mathcal{L} = \mathcal{L}_{MAE} + \mathcal{L}_{consistency} + \mathcal{L}_{contrast} \tag{12}$$

We use Adam [5] to optimize all fully connected layers and the cluster-center bank by minimizing L via the backpropagation algorithm and stochastic gradient descent.

4 EXPERIMENTS

4.1 Experimental Setup

Datasets. We conduct experiments on two datasets collected from California's highway network [10]. They are sampled from the same sensors, recording traffic speed and flow, respectively. For simplicity, we call them SPEED and FLOW. And we list their vital statistics in Table 1.

Table 1: Statistics of datasets

Dataset	Variates	Length	Sample Rate	Time Span
SPEED	11160	105120	5 min	1 year
FLOW	11160	105120	5 min	1 year

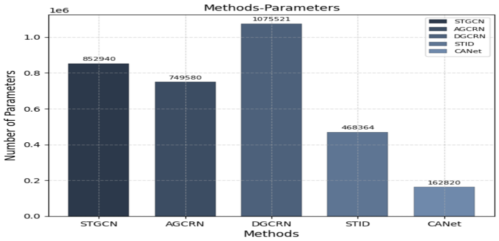

Figure 3: Methods-Parameters.

Baselines. Our baseline consists of HI [3], MLP, STID [12], D-Linear [18], STGCN [17] and Pyraformer [9]. Although there are many novel and powerful SOTA methods, such as crossformer [19], D2STGNN [14], and STEP [13], they suffer from high computational resource consumption and **fail to work on SPEED and FLOW datasets**. Thus we do not choose them as the baseline. Besides, the work initially presented these datasets is also excluded for not evaluating performance according to commonly-used evaluation metrics and high computational resource consumption [10].

Metrics. We evaluate the performance of methods on Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), which are commonly used in the field of MTS forecasting.

Implementation. We conducted experiments with Pytorch 1.10.0 on an NVIDIA RTX 3090 GPU. Due to the different complexity of models, we set different batch_sizes and learning_rates on different baselines. In our model, the number of cluster centers is M, and the model's input and output lengths are P and F.

4.2 Performance Study

We divide original datasets into training sets, test sets, and validation sets according to the ratio of 7:1:2. Our task is to predict the future time series with a length of 12. We compared the performance of different methods on the **1st**, **3rd**, **6th**, **9th**, **12th** time steps, and average (**Avg.**) performance on 1-12 time steps. The best results are highlighted in bold, and the suboptimal results are underlined. As shown in Table 2, our approach performs best on all datasets without complex spatial dependencies modeling module. It achieves performance improvement of at least 3% or 7% than the suboptimal results on the datasets, which is significant improvement in the MTS forecasting area. These results demonstrate the validity of our model.

4.3 Ablation Study

In this subsection, we conduct an ablation study on both datasets to verify the effectiveness of the cluster-aware module by comparing the results before and after removing the cluster-aware module. As shown in Table 3, when removing it, the performance decreases

Table 2: Multivariate time series forecasting on the SPEED and FLOW datasets.

Dataset		SPEED						FLOW					
Method	**Metric**	**@1**	**@3**	**@6**	**@9**	**@12**	**Avg.**	**@1**	**@3**	**@6**	**@9**	**@12**	**Avg.**
HI	MAE	2.84	2.84	2.84	2.84	2.84	2.84	36.59	36.59	36.59	36.59	36.59	36.59
	RMSE	6.28	6.28	6.28	6.28	6.28	6.28	56.99	56.99	56.99	56.99	56.99	56.99
	MAPE	6.12%	6.12%	6.12%	6.12%	6.12%	6.12%	32.51%	32.51%	32.51%	32.51%	32.51%	32.51%
MLP	MAE	0.98	1.53	2.01	2.37	2.67	1.98	15.61	18.64	22.44	26.30	30.61	23.12
	RMSE	1.88	3.23	4.48	5.33	5.96	4.36	25.66	30.73	36.56	42.15	48.43	37.36
	MAPE	1.85%	3.10%	4.30%	5.28%	6.15%	4.31%	14.58%	16.89%	19.77%	23.47%	28.15%	20.78%
D-Linear	MAE	1.02	1.60	2.09	2.45	2.76	2.07	15.53	18.90	23.05	27.45	32.29	23.88
	RMSE	1.94	3.32	4.56	5.41	6.04	4.46	25.96	31.84	38.79	45.65	52.66	39.76
	MAPE	1.92%	3.21%	4.45%	5.43%	6.30%	4.45%	14.13%	16.47%	20.82%	25.50%	34.62%	22.86%
STGCN	MAE	1.13	1.54	1.87	2.08	2.25	1.83	21.15	21.74	22.39	23.51	24.71	22.74
	RMSE	2.23	3.18	4.03	4.52	4.89	3.91	32.88	34.29	35.57	37.36	39.28	36.01
	MAPE	2.31%	3.20%	4.07%	4.65%	5.10%	4.00%	19.92%	20.53%	21.51%	22.93%	24.58%	21.94%
Pyraformer	MAE	2.68	2.70	2.72	2.74	2.77	2.72	28.38	28.49	28.63	28.93	29.12	28.73
	RMSE	5.34	5.37	5.42	5.47	5.52	5.43	51.31	51.62	51.86	52.31	52.51	51.97
	MAPE	6.03%	6.07%	6.13%	6.19%	6.26%	6.14%	31.28%	31.27%	31.48%	31.59%	31.64%	31.52%
STID	MAE	0.97	1.47	1.87	2.11	2.31	1.82	14.39	15.87	17.38	18.67	19.96	17.45
	RMSE	1.82	3.03	4.03	4.61	5.02	3.88	23.95s	26.64	29.13	31.14	33.09	29.15
	MAPE	1.83%	3.01%	4.11%	4.85%	5.42%	4.02%	13.53%	14.91%	16.62%	18.20%	19.88%	16.81%
CANet	MAE	**0.93**	**1.40**	**1.75**	**1.96**	**2.13**	**1.70**	**14.17**	**15.51**	**16.84**	**17.98**	**19.22**	**16.92**
	RMSE	**1.78**	**2.92**	**3.83**	**4.33**	**4.67**	**3.67**	**23.77**	**26.30**	**28.48**	**30.20**	**32.00**	**28.52**
	MAPE	**1.75%**	**2.85%**	**3.83%**	**4.45%**	**4.90%**	**3.72%**	**13.19%**	**14.49%**	**15.99%**	**17.36%**	**18.75%**	**16.12%**

Table 3: Ablation Study Results.

Dataset		SPEED			FLOW		
Method	**Metric**	**@6**	**@12**	**Avg.**	**@6**	**@12**	**Avg.**
CA-removed	MAE	1.95	2.56	1.93	18.95	23.64	19.26
	RMSE	4.29	5.02	4.18	31.87	39.02	32.21
	MAPE	4.26%	6.02%	4.25%	18.06%	24.34%	19.15%
CANet	MAE	**1.75**	**2.13**	**1.70**	**16.84**	**19.22**	**16.92**
	RMSE	**3.83**	**4.67**	**3.67**	**28.48**	**32.00**	**28.52**
	MAPE	**3.83%**	**4.90%**	**3.72%**	**15.99%**	**18.75%**	**16.12%**

by 14% or 20%, which indicates that using our module to generate spatial identities of variables is of great importance to prediction.

4.4 Parameters Study

Figure 3 shows the parameters of several methods [2, 7, 12, 17] when the number of variables is 11160, and indicates that CANet implements predictions in a parameter-friendly manner.

4.5 Visualization

This part visually explains the capacity to capture clusters and model spatial dependencies of the cluster-aware module. For better visualization, we applied a dimensionality reduction algorithm, MDS (Multiple Dimensional Scaling) [6], on spatial identities and cluster centers. We use a model trained on the SPEED dataset and obtain spatial identities by its cluster-aware module. The visualization result is shown in Figure 4.

Figure 4 shows that the spatial identities of variables exhibit a cluster distribution around several cluster centers. The phenomenon is consistent with our proposed example in the introduction: intersections in a traffic system have several implicit clusters, and clusters are far less than variables. Besides, the cluster centers (red dots) and spatial identities (green dots) closely correlate in the space, proving our cluster-aware module's effectiveness in capturing clusters and modeling spatial dependencies.

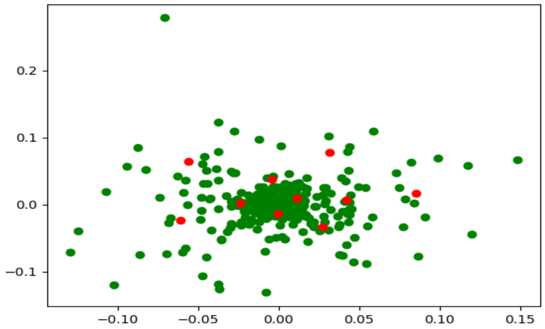

Figure 4: Visualization. Red dots and green dots denote dimensionally reduced cluster centers and spatial identities.

5 CONCLUSIONS

In this paper, we explore new approaches for modeling spatial dependencies in large-scale MTS data. Based on observations of traffic systems, we identify the spatial identities of variables are the superposition of multiple cluster centers. The perspective motivates us to propose a cluster-aware network, named CANet, to maintain and optimize a limited number of cluster centers that represent the implicit clusters of numerous variables. CANet implements weighted summation of cluster centers based on the cluster-aware module, which uses fewer parameters to achieve the best performance in large-scale MTS data. This result shows that we can model spatial dependencies in a parameter-friendly manner by mining the implicit clusters of variables.

6 ACKNOWLEDGEMENTS

This work is supported by the Youth Innovation Promotion Association CAS No.2023112.

REFERENCES

[1] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

[2] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems* 33 (2020), 17804–17815.

[3] Yue Cui, Jiandong Xie, and Kai Zheng. 2021. Historical inertia: A neglected but powerful baseline for long sequence time-series forecasting. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2965–2969.

[4] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. 2019. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1705–1714.

[5] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. (2014).

[6] Joseph B Kruskal. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29, 1 (1964), 1–27.

[7] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. 2023. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data* 17, 1 (2023), 1–21.

[8] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. [n. d.]. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

[9] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. 2021. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*.

[10] Tanwi Mallick, Prasanna Balaprakash, Eric Rask, and Jane Macfarlane. 2020. Graph-partitioning-based diffusion convolutional recurrent neural network for large-scale traffic forecasting. *Transportation Research Record* 2674, 9 (2020), 473–488.

[11] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. 2020. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14372–14381.

[12] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. 2022. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4454–4458.

[13] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. 2022. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1567–1577.

[14] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S Jensen. 2022. Decoupled Dynamic Spatial-Temporal Graph Neural Network for Traffic Forecasting. *Proc. VLDB Endow.* 15, 11 (2022), 2733–2746.

[15] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for deep spatial-temporal graph modeling. In *International Joint Conference on Artificial Intelligence 2019*. Association for the Advancement of Artificial Intelligence (AAAI), 1907–1913.

[16] Yongjun Xu, Xin Liu, Xin Cao, Changping Huang, Enke Liu, Sen Qian, Xingchen Liu, Yanjun Wu, Fengliang Dong, Cheng-Wei Qiu, et al. 2021. Artificial intelligence: A powerful paradigm for scientific research. *The Innovation* 2, 4 (2021), 100179.

[17] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3634–3640.

[18] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2022. Are transformers effective for time series forecasting? *arXiv preprint arXiv:2205.13504* (2022).

[19] Yunhao Zhang and Junchi Yan. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*.

[20] Jiawei Zhu, Qiongjie Wang, Chao Tao, Hanhan Deng, Ling Zhao, and Haifeng Li. 2021. AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting. *IEEE Access* 9 (2021), 35973–35983.