2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

DAE-FAN: Simple and Efficient Mining of Periodic Patterns for Traffic Prediction

Sheng Li[1] and Shuojiang Xu[1 *]

[1] Guilin University of Electronic Technology, Guilin 541004, China
[*] Corresponding author: shuojiang.xu@guet.edu.cn

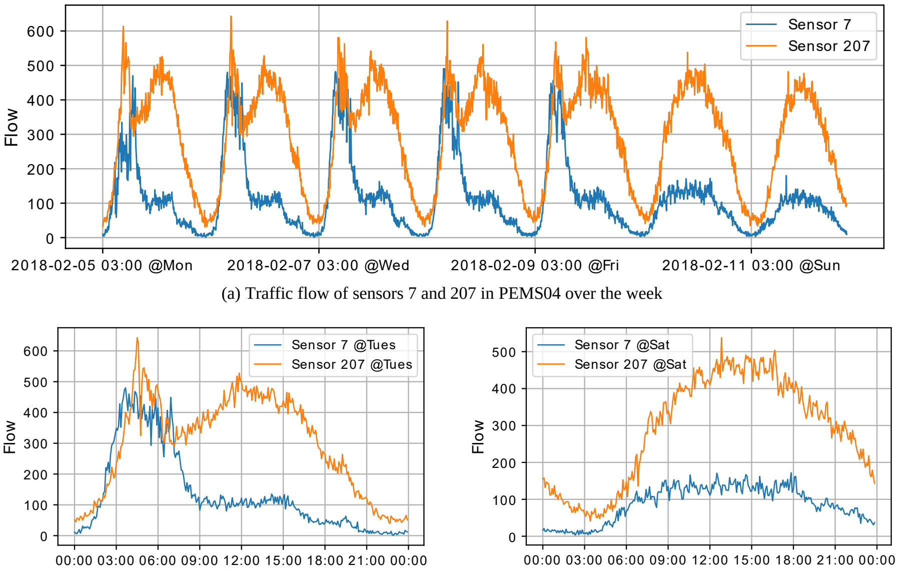
Abstract. As a fundamental task of intelligent transportation systems, traffic prediction aims to forecast traffic time series in road networks based on historical observation data to support upper-level applications. Although deep learning models have performed well in this field in recent years, their architectures have become increasingly complex and less efficient and lack sufficient modelling of the intrinsic features of traffic data over time. In this paper, we focus on the special periodicity of traffic data and propose a novel model called DAE-FAN, which designs a dual adaptive embedding mechanism consisting of feature, periodicity, and spatial embedding. Employing self-supervised learning, the combination of these embedding matrices can autonomously represent the periodic changes in traffic features and spatial patterns. In addition, we introduce the Fourier principle to construct the Fourier neural network to enhance the capability of modelling periodicity. Extensive experiments on four large public traffic datasets demonstrate the superior performance and efficiency of DAE-FAN with its simpler structure compared to current traffic prediction models, providing a promising direction for efficiently solving traffic prediction challenges.

Keywords: Traffic Prediction, Adaptive Embedding, Fourier Principle.

1 Introduction

Traffic prediction aims to predict traffic time series in road networks based on historical observations. As a foundational task of intelligent transportation systems, traffic prediction supports a large number of upper-layer applications in transportation scenarios, such as congestion warning, route planning, and location services. In recent years, deep learning models have become powerful tools for traffic prediction, among which spatio-temporal graph neural networks (STGNNs) and Transformer-based models have dominated because of their outstanding performance. Researchers have invested significant effort in developing various features for these models, such as graph convolution operations, dynamic graph structure learning mechanisms, and attention modules. However, traffic model architectures are continuously becoming complex and inefficient, with diminishing performance gains. Moreover, most of these approaches focus on modelling the relationships between individual nodes in the STGNN [1], ignoring the significance of the intrinsic features in the traffic data that change over time.

To illustrate our focus, Fig. 1(a) visualises the traffic flow of two sensors in a widely used traffic prediction benchmarking dataset (PEMS04) over one week. Clearly, the traffic data exhibit a unique daily periodic pattern. As shown in Fig. 1(b), sensors 7 and 207 display distinct peaks and valleys during weekdays, while in Fig. 1(c), the traffic flow during the non-weekdays is relatively smooth. These unique daily periodic patterns are arranged within a week, constituting the weekly periodic pattern. In addition, because of the spatial heterogeneity of the sensors, each sensor has unique feature patterns within periods, and these periods overlap and interact. Yet, most existing models still lack sufficient modelling in this regard [2].



(a) Traffic flow of sensors 7 and 207 in PEMS04 over the week

(b) Significant trend during the weekday

(c) Smooth trend during the non-weekday

Fig. 1. Examples of traffic data with special periodicity

In light of this, we focus on the special periodic characteristics of traffic data. Rather than develop more complex model architectures, our research aims to explore more efficient data representation and introduce appropriate methods into simple network models to mitigate this problem.

Specifically, we design a novel dual adaptive embedding (DAE) mechanism. Unlike conventional traffic prediction models relying on predefined graph structure data, our approach uses self-supervised learning to automatically extract discriminative embeddings from traffic feature sequences and additional temporal context. The resulting feature, spatial, and periodicity embeddings are integrated into a pair of embeddings that represent periodic changes in traffic features and spatial patterns, respectively. This integration allows the model's encoder to utilise rich contextual information, eliminating the need for extra representation extraction steps during training. In addition, to improve the coding ability for modelling periodicity, we introduce the Fourier principle into the model network and form our Fourier analysis network (FAN). Experimental results on four large publicly available traffic datasets show that our proposed DAE-FAN achieves competitive prediction performance with previous state-of-the-art (SOTA) models in traffic prediction, with a simpler structure and higher computational efficiency. Moreover, the DAE-FAN model can support real-time traffic prediction even with limited computational resources.

2 Preliminaries

2.1 Problem Statement

Traffic prediction aims to predict the traffic data $X_{t+1:t+T'} \in \mathbb{R}^{N\times D_{out}}$ for the next T' time steps by training a model $F(\cdot)$ with parameters θ, based on the historical sequence $X_{t-T+1:t} \in \mathbb{R}^{N\times D_{in}}$ of the past T time steps. This process can be expressed as follows:

$$[X_{t-T+1},\ldots,X_t] \xrightarrow[\theta]{F(\cdot)} [X_{t+1},\ldots,X_{t+T'}], \tag{1}$$

where t is the current time step, N is the number of sensors in the traffic network, D_{in} is the dimension of the input features, and D_{out} is the dimension of the output features. In this example, the dimension of the input features and that of the output features are 1, representing traffic speed or traffic flow.

2.2 Fourier Analysis

At the core of the Fourier function is the decomposition of any function into linear combinations of sine and cosine functions of different frequencies, including non-periodic functions via period expansions, revealing underlying periodic patterns within complex functions [3]. Mathematically, a function $f(x)$ can be expressed through the Fourier series:

$$f(x) = a_0 + \sum_{n=1}^{\infty}\left[a_n \cos\left(\frac{2\pi nx}{T}\right) + b_n \sin\left(\frac{2\pi nx}{T}\right)\right], \tag{2}$$

where T is the period of the function; n is the frequency component; and a_0, a_n, and b_n are determined by the integral of the function over one period:

$$a_0 = \frac{1}{T}\int_0^T f(x)dx\ , \tag{3}$$

$$a_n = \frac{1}{T}\int_0^T f(x)\cos\left(\frac{2\pi nx}{T}\right)dx\ , \tag{4}$$

$$b_n = \frac{1}{T}\int_0^T f(x)\sin\left(\frac{2\pi nx}{T}\right)dx\ . \tag{5}$$

In this paper, we embed Fourier decomposition into the structure of the network because different combinations of frequencies can cover a variety of complex period patterns, allowing the network to learn complex interactions based on frequency features [4]. Specifically, a function $f(x): R^{dx} \to R^{dy}$ is fitted to the network, where dx and dy denote the dimensions of x and y, respectively, and the Fourier series expansion of the function is as follows:

$$\begin{aligned}
f(x) &\triangleq a_0 + \sum_{n=1}^{\infty}\left[a_n \cos\left(\frac{2\pi nx}{T}\right) + b_n \sin\left(\frac{2\pi nx}{T}\right)\right] \\
&\overset{\text{(I)}}{=} a_0 + \sum_{n=1}^{N}\left(w_n^c \cos\left(w_n^{\text{in}} x\right) + w_n^s \sin\left(w_n^{\text{in}} x\right)\right) \\
&\overset{\text{(II)}}{=} B + [w_1^c, w_2^c, \cdots, w_n^c]\cos([w_1^{\text{in}} \,||\, w_2^{\text{in}} \,||\cdots||\, w_n^{\text{in}}]x) \\
&\quad + [w_1^s, w_2^s, \cdots, w_n^s]\sin([w_1^{\text{in}} \,||\, w_2^{\text{in}} \,||\cdots||\, w_n^{\text{in}}]x) \\
&= B + W_c \cos(W_{\text{in}} x) + W_s \sin(W_{\text{in}} x) \\
&\overset{\text{(III)}}{=} B + W_{\text{out}}[\cos(W_{\text{in}} x) \,||\, \sin(W_{\text{in}} x)],
\end{aligned} \tag{6}$$

where $B \in \mathbb{R}^{dy}, W_{\text{in}} \in \mathbb{R}^{N \times dx}$, $W_{\text{out}} \in \mathbb{R}^{dy \times 2N}$ are all learnable parameters. $[\cdot \,||\, \cdot]$ and $[\cdot, \cdot]$ denote the concatenation along the first and second dimensions, respectively. In Equation (6), (I) follows the computation of a_n and b_n in Equations (4) and (5), and (II) and (III) follow the equivalence of matrix operations.

3 Methodology

Fig. 2 illustrates the structure of the proposed DAE-FAN model. The following subsections describe each module in detail.

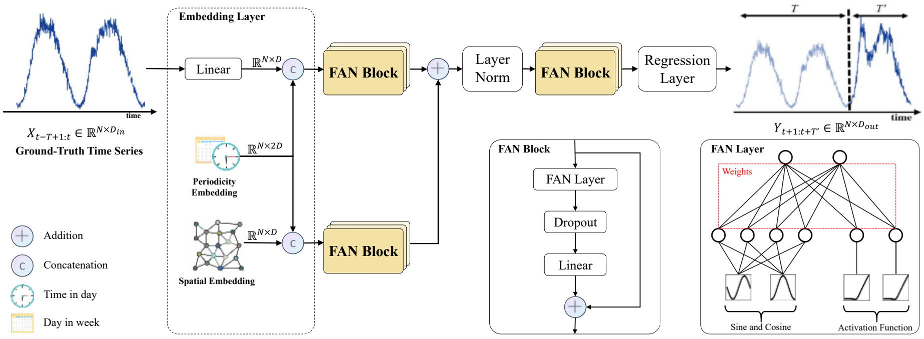


Fig. 2. The overview of the proposed DAE-FAN.

3.1 Embedding Layer

Periodicity embedding. To capture the information about the period of the traffic data, we apply additional time signals at time step t : time-in-day X_t^{Tid} and day-in-week X_t^{Diw}, which are useful for the model to capture the temporal context information embedded in the traffic data.

Additionally, we apply two learnable embedding dictionaries: the time-in-day embedding dictionary $Q^{\mathrm{Tid}} \in \mathbb{R}^{N_d \times D}$ and the day-in-week embedding dictionary $Q^{\mathrm{Diw}} \in \mathbb{R}^{N_w \times D}$. These two embedding dictionaries use the inputs X_t^{Tid} and X_t^{Diw} as indices to retrieve the corresponding time-scale embeddings, i.e. the daily embedding $E_t^{\mathrm{Tid}} \in \mathbb{R}^{N \times D}$ and the weekly embedding $E_t^{\mathrm{Diw}} \in \mathbb{R}^{N \times D}$. Through this approach, the two embedding dictionaries adaptively learn the internal representations of daily and weekly periodicities within the network, effectively differentiating the traffic pattern differences between weekends and weekdays.

In our example, $N_d = 288$ denotes the number of time slots in a day, $N_w = 7$ denotes the number of days in a week, N is the number of nodes, and D is the hidden dimension. For the convenience of representation, we integrate the above two scales of embeddings (E_t^{Tid} and E_t^{Diw}) through the concatenation operation $\mathrm{Concat}(\cdot)$ and represent it as the periodic embedding $E_t^{\mathrm{P}} \in \mathbb{R}^{N \times 2D}$:

$$E_t^{\mathrm{P}} = \mathrm{Concat}(E_t^{\mathrm{Tid}}, E_t^{\mathrm{Diw}}) . \tag{7}$$

Feature embedding. For the input feature sequence $X_{t-T+1:t} \in \mathbb{R}^{N \times D_{in}}$, we apply a linear layer $\mathrm{Linear}(\cdot)$ for encoding to retain the intrinsic information in the original data:

$$E_t^F = \mathrm{Linear}(X_{t-T+1:t}) . \tag{8}$$

where $E_t^F \in \mathbb{R}^{N \times D}$ is the feature embedding. By combining the feature embedding with the periodicity embedding, we obtain the combined representation $E_t \in \mathbb{R}^{N \times 3D}$ of traffic features over the period:

$$E_t = \mathrm{Concat}(E_t^F, E_t^P) . \tag{9}$$

Dual Embedding. Traffic flow data are characterised by the sequential flow of events and the cyclic nature of time, i.e., events at a node in a traffic road network usually affect its direct neighbouring nodes, resulting in the similarity of periods. Therefore, establishing the correlation between sequences is beneficial for the model to understand the internal relationships of the periodic changes in different traffic data. In this regard, instead of constructing a predefined or dynamic adjacency matrix as a graph for spatial relationship modelling, we create a shared adaptive spatial embedding $E^S \in \mathbb{R}^{N \times D}$, which is initialised with the Xavier uniform initialisation and then treated as a model parameter.

We concatenate the spatial embedding with the periodicity embedding to obtain the combined representation of space over the period, denoted as $E_t^{\mathrm{Dual}} \in \mathbb{R}^{N\times 3D}$, and we refer to it as the dual embedding:

$$E_t^{\mathrm{Dual}} = \mathrm{Concat}(E^S, E_t^{\mathrm{P}}) . \tag{10}$$

3.2 FAN Block and Regression layer

The Fourier analysis layer is based on the Fourier series formula in Equation (2). To appropriately express it as part of a neural network, we need to ensure that the features of its intermediate layers can be used for subsequent modelling. Moreover, as the depth of the model network increases, the Fourier analysis layer should be able to handle or represent more complex Fourier coefficients [5], enhancing the ability to model periodicity. Based on the above principles, we decouple Equation (6) as follows:

$$f(x) = f_{out} \circ f_{in} \circ x , \tag{11}$$

where

$$f_{in}(x) = [\cos(W_{\mathrm{in}} x) \,||\, \sin(W_{\mathrm{in}} x)] , \tag{12}$$

$$f_{out}(x) = B + W_{\mathrm{out}} x . \tag{13}$$

In the neural network, f_{in} and f_{out} are not applied sequentially but simultaneously. Therefore, the FAN layer $\phi(\cdot)$ can be expressed as follows:

$$\phi(x) \triangleq [\cos(W_p x) \,||\, \sin(W_p x) \,||\, \sigma(B_{\bar{p}} + W_{\bar{p}} x)] , \tag{14}$$

where $W_p \in \mathbb{R}^{d_x \times d_p}$, $W_{\bar{p}} \in \mathbb{R}^{d_x \times d_{\bar{p}}}$, and $B_{\bar{p}} \in \mathbb{R}^{d_{\bar{p}}}$ are learnable parameters; d_p and $d_{\bar{p}}$ are hyperparameters, representing the first dimensions of W_p and $W_{\bar{p}}$, respectively; $\phi(x) \in \mathbb{R}^{2d_p + d_{\bar{p}}}$ is the output; and σ denotes the activation function, which is set as the ReLU activation function in this example.

Based on the FAN layer, we construct FAN blocks with residual connections to encode information. For the feature embedding E_t, the l th FAN block can be expressed as follows:

$$(E_t)^{l+1} = \mathrm{Linear}^l(\mathrm{Dropout}(\phi^l(E_t)^l)) + (E_t)^l . \tag{15}$$

We also use the FAN block to encode the dual embedding E_t^{Dual}. In this way, we obtain the encoded hidden representations $(E_t)^L$ and $(E_t^{\mathrm{Dual}})^L$. Subsequently, we fuse the two embeddings into the layernorm layer:

$$H_t = \mathrm{LayerNorm}((E_t)^L + (E_t^{\mathrm{Dual}})^L) . \tag{16}$$

Similarly, it then goes through the M layer FAN block again for global coding to obtain $(H_t)^M \in \mathbb{R}^{N\times 3D}$, which is fed into the regression layer for prediction:

$$Y_{t+1:t+T'} = \mathrm{Linear}((H_t)^M) , \tag{17}$$

where $Y_{t+1:t+T'} \in \mathbb{R}^{N\times D_{out}}$ is the prediction.

4 Experimental Study

4.1 Experimental Setup

Datasets. We evaluated the performance of the DAE-FAN model on four major traffic prediction benchmarking datasets – PEMS-BAY, PEMS04, PEMS07, and PEMS08 – widely used in the traffic prediction research community. PEMS-BAY provides traffic speed information, and PEMS04, PEMS07, and PEMS08 provide traffic flow information on the road network. Table 1 shows the details of these datasets.

Table 1. Summary of traffic datasets.

Dataset	No. of Sensors	No. of Records	Interval (min)	Time Range	Location
PEMS-BAY	325	52,116	5	01/2017–05/2017	San Francisco Bay Area
PEMS04	307	16,992	5	01/2018–02/2018	California District 4
PEMS07	883	28,224	5	05/2017–08/2017	California District 7
PEMS08	170	17,856	5	07/2016–08/2016	California District 8

Metrics. Following previous work, we adopted three widely used metrics to evaluate the performance of traffic prediction methods: mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i| , \tag{18}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100\% , \tag{19}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2} , \tag{20}$$

where $\hat{y}_i$ and y_i represent the predicted value and ground truth of traffic information at node i , respectively, and n denotes the total number of nodes. MAE calculates the average absolute error between predictions and ground truth. RMSE weights larger errors by squaring them before averaging. MAPE normalises errors by ground truth values as percentages. All metrics follow the principle that lower values indicate better performance.

Implementation. We referred to previous work to divide the dataset and used a ratio of 7:1:2 for training, validation, and test sets for the PEMS-BAY dataset and 6:2:2 for the other datasets. We set the input length and prediction length to 1 h, i.e., $T = T' = 12$, the number of layers in the Fourier analysis block was 3, the number of epochs was 100, and Adam was used as the optimiser with a learning rate of 0.001, which was gradually reduced. For the PEMS-BAY, PEMS04 and PEMS08 datasets, we set the embedding dimensions E_F , E_{Tod} , E_{Dow} , and E_S in DAE-FAN to 36, and for the PEMS07 dataset, which has more nodes, we set these embedding dimensions to 72. The proposed model was implemented with Pytorch 2.1.1 on an NVIDIA RTX 4080Ti GPU. All experiments were conducted on the BasicTS platform for fair and consistent comparisons [6].

Baselines. In this study, we compare our proposed method with several widely used baselines in the field. We use HI [7] as a traditional benchmark, which reflects standard industry practices. We also consider STGNNs such as spatio-temporal graph convolutional networks (STGCN, [8]), DCRNN [9], GWNet [10], DGCRN [11], MTGNN [12], and AGCRN [13], in which GWNet, DGCRN, MTGNN, and AGCRN construct spatial dependencies in transportation road networks through adaptive learning of adjacency matrix. In addition, we selected special methods, such as STNorm [14], which understand the spatio-temporal relationship by decomposing the high- and low-frequency components of traffic data. In addition, STID [15] mitigates the indistinguishability of the samples in spatial and temporal dimensions by adding spatial and periodic embeddings to the inputs. We also selected some Transformer-based models focusing on traffic prediction, such as GMAN [16], STAEformer [17], and D2STGNN [18]. In terms of data representation, GMAN incorporates spatial and temporal embeddings, STAEformer modifies the spatial embedding to a spatio-temporal embedding, and D2STGNN decomposes the diffusion signal in the estimate gate using two temporal embeddings, a source node embedding, and a target node embedding. These embedding matrices are learnable parameters.

4.2 Performance study

In this section, we evaluate the performance of the proposed DAE-FAN. Table 2 shows the model performance on the PEMS-BAY, PEMS04, PEMS07, and PEMS08 datasets, where the best results are in bold, and the second-best results are underlined. Overall, models that adaptively learn spatial dependencies using spatial embedding techniques generally outperform models that construct spatial dependencies based on adjacency matrices. At the architectural level, Transformer-based models outperform STGNNs, and this advantage stems from the stronger ability to capture complex spatio-temporal relationships by the self-attention mechanism.

Specifically, for the PEMS-BAY dataset, DAE-FAN outperforms other models in all metrics. Compared with the second-best model, it improves by 1.92%, 0.86%, and 0.28% in terms of MAE, MAPE, and RMSE, respectively. For the PEMS04, PEMS07 and PEMS08 datasets, DAE-FAN shows comparable performance to STAEformer in the MSE and MAPE and achieves an overall advantage in the metrics, with improvements of 1.88%, 0.79%, and 0.26%, respectively, over the second-best model. This suggests that DAE-FAN performs well in terms of reduced extreme errors and better overall prediction stability. Overall, DAE-FAN achieves competitive results in terms of performance compared to other SOTA models; moreover, our proposed method is simpler and more efficient.

Table 2. Comparative performance analysis of models on PEMS datasets.

Model	PEMS-BAY			PEMS04			PEMS07			PEMS08		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
HI	6.83	16.83	14.81	38.32	28.17	56.83	45.74	21.86	67.98	32.86	20.49	47.94
STGCN	1.69	3.81	3.80	19.67	13.44	31.36	22.17	9.58	35.58	16.35	10.68	25.63
DCRNN	1.59	3.59	3.68	19.55	13.31	31.10	21.16	9.07	34.11	15.24	9.76	24.25
GWNet	1.58	3.50	3.65	19.01	13.40	30.43	20.29	8.67	33.21	14.64	9.28	23.60
DGCRN	1.60	3.59	3.74	19.19	14.15	30.57	20.43	8.81	33.47	14.97	10.13	23.90
MTGNN	1.58	3.54	3.67	19.31	13.46	31.12	20.54	9.30	34.07	15.45	10.02	24.43
STNorm	1.60	3.57	3.71	19.09	13.25	31.75	20.43	8.73	33.93	15.46	10.39	25.23
STAEformer	1.57	3.53	3.58	**18.21**	**12.18**	30.38	19.26	**7.94**	33.25	**13.40**	**8.74**	23.33
D2STGNN	1.56	3.49	3.59	18.64	13.11	30.00	19.56	8.31	32.77	14.16	9.21	23.44
AGCRN	1.61	3.66	3.70	19.29	13.16	31.00	20.65	8.76	34.35	15.95	10.35	25.50
GMAN	1.61	3.62	3.57	19.34	13.71	30.98	20.35	8.65	33.39	14.32	9.50	23.89
STID	1.58	3.54	3.60	18.42	12.85	29.90	19.65	8.28	32.78	14.17	9.26	23.31
DAE-FAN	**1.53**	**3.46**	**3.56**	18.27	12.42	**29.81**	**19.21**	8.02	**32.51**	14.05	9.14	**23.27**

4.3 Efficiency Study

In this section, we compare the efficiency of DAE-FAN with other learning methods based on all datasets. For a more intuitive and effective comparison, we compare the average training time required for each epoch of these models. The results are shown in Table 3. Transformer-based models such as STAEformer, D2STGNN, and GMAN require more computation time compared to other models because of their computational complexity, which usually grows quadratically with the length of the input time series, highlighting the impact of the attention mechanism on the computational requirements for training. Second, because of their combined sequence model (e.g., RNN) and graph convolutional architecture, STGNNs result in significant time consumed to implement their iterative approach for prediction and graph-to-graph message passing. DAE-FAN, although second only to STID, which has a pure multilayer perceptron architecture in terms of training speed, still achieves a significant reduction in training time compared to the other models, demonstrating an efficient computational performance.

Table 3. Training time (s) for each epoch of the models on four datasets.

Method	PEMS-BAY	PEMS04	PEMS07	PEMS08
STGCN	24.58	6.73	30.02	4.50
DCRNN	83.63	21.92	154.79	13.70
GWNet	40.92	11.12	68.04	7.30
DGCRN	112.95	24.93	263.19	70.41
MTGNN	16.34	4.72	28.43	3.05
STNorm	19.24	5.26	31.81	3.07
STAEformer	152.90	40.50	288.65	30.03
D2STGNN	169.02	44.74	369.58	43.62
AGCRN	36.67	10.17	55.44	6.99
GMAN	125.78	37.65	238.91	39.63
STID	**3.16**	**1.30**	**3.26**	**0.87**
DAE-FAN	8.48	1.92	12.38	2.07

This analysis emphasises the advantages of DAE-FAN in terms of prediction accuracy and computational efficiency. DAE-FAN achieves comparable prediction accuracy to the traffic prediction SOTA model with a concise architecture and low computational resource consumption. It exhibits a balance between model complexity and prediction efficiency and provides a solution that takes into account both the technological frontiers and engineering practicability in the field of traffic prediction and spatio-temporal data modelling.

4.4 Ablation Study

To validate the effectiveness of our proposed components, we set up three variants of DAE-FAN: w/o E^{Dual} removes the dual embedding E^{Dual}; w/o E^{P} removes the periodicity embedding E^{P}; w/o FAN replaces the FAN in the model with a fully connected layer. We conducted experiments on the PEMS04 and PEMS07 datasets, and the results are shown in Fig. 3. The w/o E^{Dual} performance degradation is significant, demonstrating that our proposed embedding effectively captures the spatio-temporal dependence of the traffic scene. The w/o E^{P} error rises, indicating the importance of capturing periodic patterns in traffic sequences for traffic prediction. In addition, removing FAN from the model degrades the performance, especially in PEMS07 (which has more nodes), suggesting that FAN enhances the model for a more comprehensive understanding of the traffic data.

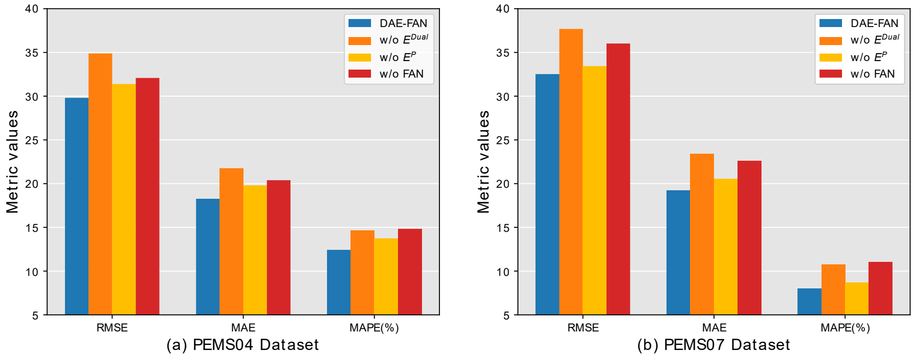
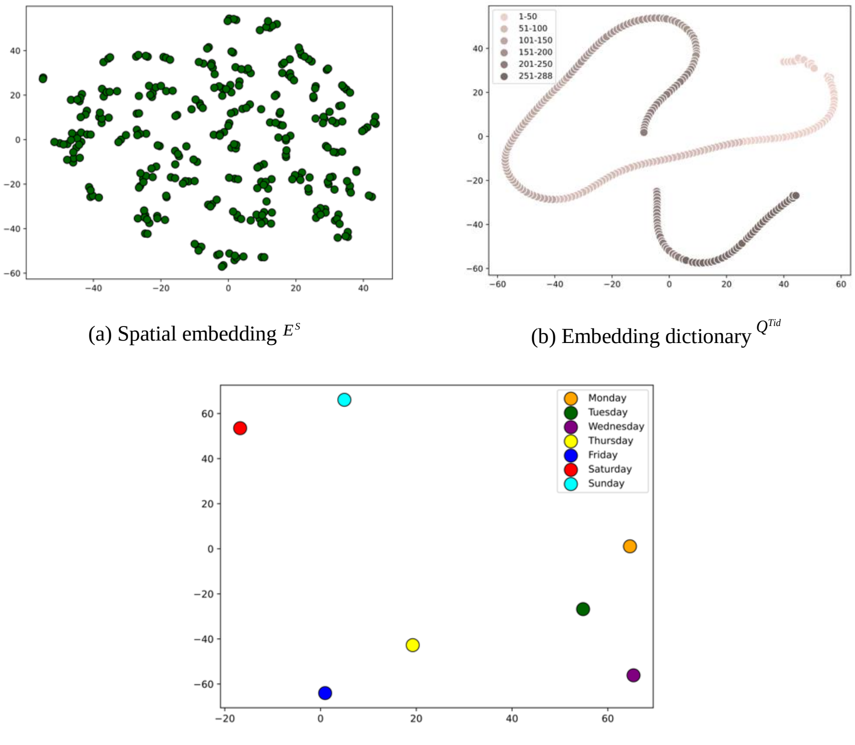


Fig. 3. Model component analysis of DAE-FAN.

4.5 Visualization

To visualise and understand our proposed embedding representation, in this section, we use the t-SNE (t-distributed stochastic neighbour embedding) technique [19] to visualise the model on the PEMS04 dataset and learn the adaptive embedding matrix $E^{S} \in \mathbb{R}^{N \times D}$, the embedding dictionary $Q^{Tid} \in \mathbb{R}^{N_d \times D}$, and $Q^{Diw} \in \mathbb{R}^{N_w \times D}$. The results are shown in Fig. 4.



(a) Spatial embedding E^S

(b) Embedding dictionary Q^{Tid}

(c) Embedding dictionary Q^{Diw}

Fig. 4. Visualization of learned embedding.

First, Fig. 4(a) indicates that different node embeddings learned by the model naturally form clusters, consistent with the knowledge that neighbouring node features tend to be similar in traffic road networks. Second, Fig. 4(b) visualises the embeddings of 288 timestamps per day, and the adjacent timestamp features are highly correlated, consistent with the continuity of temporal data. Finally, Fig. 4(c) indicates that significant separation occurs in the feature embeddings of weekdays and non-weekdays, which validates the effectiveness of the model's ability to distinguish multiple periodic patterns in traffic data.

5 Conclusion

In this study, we focus on the intrinsic periodicity of traffic data and propose the novel DAE-FAN. DAE-FAN can efficiently learn the feature patterns at each node within multiple periods using the Fourier principle to enhance periodic modelling. Extensive experiments were conducted on four open-source traffic datasets, demonstrating the dual advantages of DAE-FAN over other benchmark methods in performance and efficiency. We believe DAE-FAN is promising for solving traffic prediction challenges rather than being constrained to designing complex models. In future work, we will extend the method to other scenarios with periodic characteristics, as well as evaluate the performance on non-periodic datasets.

Acknowledgements.
This project was funded by the Guangxi Science and Technology Planning Project, Science and Technology Department of Guangxi Zhuang Autonomous Region (grant number AD23026153). This work was also funded by the National Natural Science Foundation of China (grant number: 62462016) and the Science and Technology Department of Guangxi Zhuang Autonomous Region (grant number: AD23026096).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ye, B.-L., Zhang, M., Li, L., Liu, C., Wu, W.: A survey of traffic flow prediction methods based on long short-term memory networks. IEEE Intell. Transp. Syst. Mag. **16**(5), 87–112 (2024).
2. Liu, Z., Hartwig, T., Ueda, M.: Neural networks fail to learn periodic functions and how to fix it. In: Advances in Neural Information Processing Systems, pp. 1583–1594 (2020).
3. Duoandikoetxea, J.: Fourier analysis. 1st ed. American Mathematical Society, (2024).
4. Uteuliyeva, M., et al.: Fourier neural networks: A comparative study. Intell. Data Anal. **24**(5), 1107–1120 (2020).
5. Dong, Y., et al.: FAN: Fourier analysis networks. arXiv e-prints, arXiv:2410.02675 (2024).
6. Shao, Z., Wang, F., Xu, Y., et al.: Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. IEEE Trans. Knowl. Data Eng. **37**(1), 291–305 (2024).
7. Cui, Y., Xie, J., Zheng, K.: 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, pp. 2965–2969 (2021).
8. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 3634–3640 (2018).
9. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: 6th International Conference on Learning Representations (2018).
10. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, pp. 1907–1913 (2019).
11. Li, F., et al.: Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. ACM Trans. Knowl. Discov. Data. **17**(1), 1–21 (2023).
12. Wu, Z., et al.: Connecting the dots: Multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2020).

13. Bai, L., Yao, L., Li, C., Wang, X., Wang, C.: Adaptive graph convolutional recurrent network for traffic forecasting. In: Advances in Neural Information Processing Systems, pp. 17804–17815 (2020).
14. Deng, J., Chen, X., Jiang, R., Song, X., Tsang, I. W.: ST-Norm: Spatial and temporal normalization for multi-variate time series forecasting. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 269–278 (2021).
15. Shao, Z., Zhang, Z., Wang, F., Wei, W., Xu, Y.: 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, pp. 4454–4458 (2022).
16. Zheng, C., Fan, X., Wang, C., Qi, J.: GMAN: A graph multi-attention network for traffic prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1234–1241 (2020).
17. Liu, H., et al.: Spatio-temporal adaptive embedding makes vanilla transformer SOTA for traffic forecasting. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 4125–4129 (2023).
18. Shao, Z., et al.: Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. In: Proceedings of the VLDB Endowment. **15**(11), pp. 2733–2746 (2022).
19. Van Der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(2605), 2579-2605 (2008).