MoME: Mixture of Multi-Domain Experts for Multivariate Long-Term Series Forecasting

1st Xinyu Li[1,2] lxy_fdu2022@163.com
2nd Zhiheng Yang[1,2] 23210240360@m.fudan.edu.cn
3rd Hao Xu[1,2] 21210240376@m.fudan.edu.cn
4th Yunqi Cai[1,2] vegetableqqq@gmail.com
5th Hong Lu[1,2] honglu@fudan.edu.cn
6th Xin Wang[1,2] xinw@fudan.edu.cn
7th Jin Zhao[1,2] jzhao@fudan.edu.cn
8th Fenglin Qi[3] flqi@fudan.edu.cn
9th Jiajie Shen[3] jiajieshen@fudan.edu.cn

[1]School of Computer Science, [2]Shanghai Key Laboratory of Intelligent Information Processing, [3]Informatization Office
Fudan University
Shanghai, China

***Abstract*—Time series forecasting is always important, with multivariate long-term series forecasting being its most challenging task. Here, the existing methods typically learn only in a single domain and focus on optimizing model structures, leading to incomplete information mining and imprecise predictions. To address this, we propose a generalized Mixture of Multi-Domain Experts (MoME) for multivariate long-term series forecasting. Unlike most existing methods, MoME focuses on multi-perspective information mining and fusing. To this end, MoME transforms time series into the frequency and spatial domains to learn their respective representations. MoME regards variates information as embedded features and applies fast Fourier transform to the time dimension. Then it learns embedded features in the frequency domain. In spatial domain learning, MoME applies self-attention mechanism on the variates dimension to efficiently capture dependencies among multiple variates. Finally, MoME fuses the outputs from all domains, reinterprets and integrates information across multiple domains, and predicts future time series. Extensive experiments prove that MoME outperforms state-of-the-art (SOTA) methods. Code is available at: https://github.com/lxy-PhD2022/MoME**

***Index Terms*—Time series forecasting, Multivariate long-term forecasting, Multi-Domain learning, Generalized MoE**

I. Introduction

Time series forecasting is essential in various real-world scenarios. The development of this field began in the 1970s, initially focusing on statistical models [1-9]. However, these methods rely heavily on manually designed feature engineering, which may lead to poor predictions if the design is not suitable. The introduction of machine learning technologies [10-18] marked a shift from heavy reliance on manual feature engineering. The advent of deep learning methods [19-25] has further weakened the reliance on manual feature engineering. The evolution of deep learning methods in time series forecasting field has seen a shift from simple models to complex architectures.

This work is partially supported by the National Natural Science Foundation of China (Project No. 62472101) and Shanghai Municipal Science and Technology Commission (Project No. 22dz1204900). Xin Wang is the corresponding author.

Initially, Recurrent Neural Networks [23] and their variants, such as Long Short-Term Memory Networks [24] and Gated Recurrent Units Networks [25], were widely used, but they suffered from issues like gradient vanishing or exploding and error accumulation. Subsequently, Convolutional Neural Networks [26] were introduced, using stacked convolutional layers to aggregate local features and capture complex patterns, yet they fell short in capturing global features.

To overcome this limitation, Transformer-based model [27-31] was introduced, effectively handling long-term dependencies with its self-attention mechanism, but it disrupted the temporal order, making it challenging to model time series correctly. In response, PatchTST [32] splits time series into patches for linear layer processing, then applies self-attention to patch dimensions. It prevents the temporal order disruption with self-attention applied directly to time series, thus improving prediction precision. To better model complex time series, TimesNet [33] designs period analysis techniques. In addition, FreTS [34], MSGNet [35], and iTransformer [36] further explore accurate forecasting. However, a common issue with these deep learning methods was their failure to fully exploit multi-perspective information, understanding and modeling data only from a single data domain perspective.

To mine information from multiple perspectives and capture comprehensive patterns, we designed the Mixture of Multi-Domain Experts (MoME) to learn and fuse in the time, frequency, and spatial domains, enhancing prediction accuracy. Existing frequency domain learning methods [28, 34] usually transform and learn the variates or time dimension independently, resulting in a lack of global perspective. We regard the variates dimension as an embedded feature and perform fast Fourier transform on the time dimension. In this way, we obtain more accurate global information. We apply a linear layer to learn and extract the embedded features of each frequency component to achieve comprehensive and accurate frequency domain learning.

Additionally, learning in the spatial domain helps understand how features across multiple variates interact and cap-

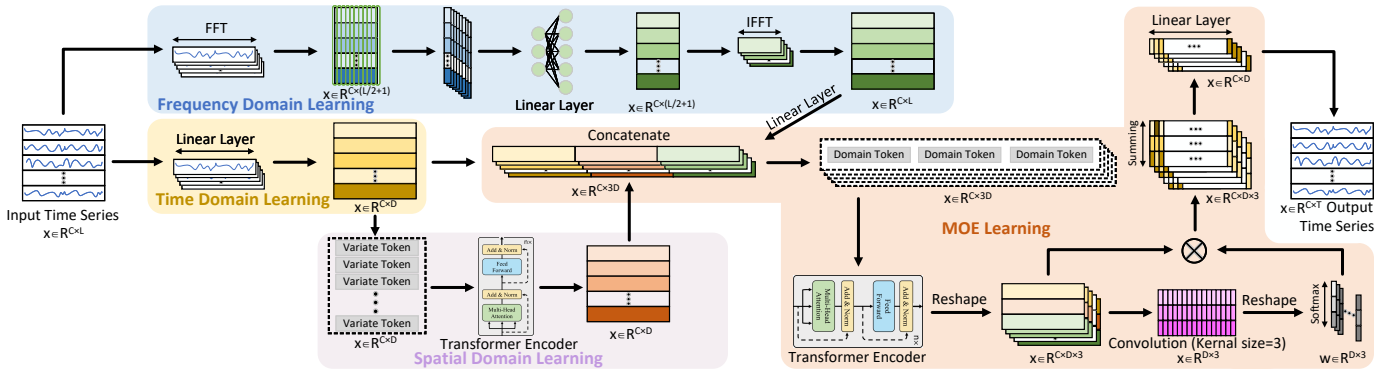


Fig. 1. Overall structure of MoME, which includes frequency, time, spatial, and MoE learning. It first transfers time series into multiple domains and extracts respective features. Then, multiple domains' features are concatenated and fused by the MoE learning. Finally, prediction results are obtained through linear mapping.

tures the dependencies between variates, further enhancing prediction accuracy. We apply self-attention mechanism on the variates dimension to calculate their dependencies, offering stronger modeling capabilities than existing methods that compute on the time dimension. Finally, we designed a MoE mechanism to fuse the learning results of multiple domains. Specifically, we apply Transformer to interact with them and complement each other by learning the relationships between different domains. Then, we perform concatenation and convolution with softmax on the corresponding variates dimensions. The convolution learns the importance of different domains and outputs weights. We assign the weights to these domains correspondingly and sum them up as a whole, fusing rich multiple domain features. After that, we perform linear mapping on it, achieving robust and precise multivariate long-term time series forecasting.

Our main contributions are as follows.

- We find that information mining is very important but often overlooked.
- We further propose a method for mining and fusing from multiple information domains.
- Extensive experiments prove that our method outperforms SOTA methods which ignore information mining.

II. MoME

A. Problem Definition

The inputs are several time series and each series represents the single variate. For each input series $x = (x_1, \ldots, x_L) \in R^{1\times L}$ with single variate and look-back window L, the goal is to forecast T future values $y = (x_{L+1}, \ldots, x_{L+T}) \in R^{1\times T}$.

B. Model Structure

1) Frequency Domain Learning: We first apply instance normalization [37-38] to the input, normalizing each sample point to have zero mean and unit standard deviation. Due to different time data might show different patterns in time or frequency domain, studying both helps the model capture these patterns, making predictions better. Thus, MoME learns from frequency domain to find more information than just time domain, shown in the frequency domain learning of Fig. 1 (the blue part).

It could be seen that MoME first performs a fast Fourier transform (FFT) on the time dimension and converts each time series into the frequency domain. Subsequently, MoME innovatively regards the variates dimension as the embedded feature of the frequency dimension, as shown in the green box. This unique processing method means that at each frequency component, the model could comprehensively observe the features of all variates, providing a richer information foundation for subsequent analysis. For each frequency component in the frequency domain, MoME cleverly applies a linear layer on its embedded feature dimension for in-depth learning. In this way, MoME could accurately capture the periodic characteristics and significant patterns of the time series on the global structure of the entire data.

After learning, the frequency domain information is subjected to an inverse fast Fourier transform (IFFT) and converted back to the time domain again. At this time, the output results of frequency domain learning obviously contain significant periodic characteristics and patterns, providing strong support for multivariate long-term series prediction.

Formally, the process of frequency domain learning could be described as follows. (1) $X_{fft} = FFT(X_{inp})$, where X_{inp}, $FFT(\cdot)$, and X_{fft} respectively represent the input time series, fast Fourier transform employed in dimension L, and the representation of the input time series in the frequency domain. $X_{inp} \in R^{C\times L}$ and $X_{fft} \in R^{C\times(L/2+1)}$, where C is the variates number and L is the series length. (2) $A_{fea} = Linear(X_{fre})$, where $Linear(\cdot)$ and A_{fea} represents the linear layer employed in dimension C and extracted features in the frequency domain, respectively. $A_{fea} \in R^{C\times(L/2+1)}$. (3) $X_{fre} = IFFT(A_{fea})$, where $IFFT(\cdot)$ and X_{fre} represent the inverse fast Fourier transform employed in dimension L and the representation of the extracted features in the time domain, respectively. $X_{fre} \in R^{C\times L}$. Then, due to the requirement of MoE learning, the time dimension of X_{fre} is projected from L to D through a linear layer.

2) Time Domain Learning: The time domain is the most basic form of time series representation, and its learning is also essential. Since the effectiveness of linear layers in time series prediction has been deeply proven [39], MoME applies a linear layer to capture the temporal relationships of sequences. Another role of the linear layer is to embed the input sequence into variates tokens with a dimension of D, which could be directly used as input tokens for Transformer during spatial domain learning.

3) Spatial Domain Learning: Learning in the spatial domain refers to understanding the relationships between variates. This aids in prediction by modeling the dependencies between variates. Since there is no sequential relationship between variates, self-attention naturally suits learning their interrelations. MoME introduces the Transformer encoder, performing self-attention calculations on the variates tokens "$Tokens$" obtained in time domain learning to model dependencies between variates.

Formally, the process could be described as follows. (1) $Q_h = (\boldsymbol{Tokens})^T \mathbf{W}_h^Q$, (2) $K_h = (\boldsymbol{Tokens})^T \mathbf{W}_h^K$, (3) $V_h = (\boldsymbol{Tokens})^T \mathbf{W}_h^V$. Where Q_h, K_h, and V_h represents the query, key, and value matric, respectively. W_h^Q, W_h^K, and W_h^V is the weight in multi-head attention space, where h is head number and h = 1,...,H. (4) $\mathrm{A_{tte}} = \mathrm{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right) V_h$, where A_{tte} represents the attention score. Then, it is fed into the BatchNorm layers and feed forward network with residual connections, obtaining the output of spatial domain learning. By doing this, each variates is treated as a token, and the relationships between them are calculated by the self-attention mechanism. Thus, MoME could learn the dependencies between multiple variates, enabling spatial domain learning.

4) MoE Learning: After learning in each domain is completed, we combine them through a novel MoE mechanism by weighted fusion to aggregate the features of each domain. As shown in the orange part of Fig. 1, we first apply Transformer to learn the relationships between various domains, which can effectively interact and complement information of features in different domains. To this end, we concatenate the outputs of all domain learning along the time dimension and then process each variable dimension independently in the form of channel independence. Each domain is regarded as a token, and their correlations are calculated through self-attention.

Then, we concatenate each domain in the channel dimension and then apply convolution to learn the importance of each domain and assign it a weight. Merely roughly calculating the overall weight of each domain is insufficient to delicately learn the strength relationship between different time steps among domains, which in turn affects domain feature fusion and prediction accuracy. Therefore, we embed the time dimension into the input channel of 1D convolution to calculate the weights of all time steps and extract sufficient semantic information. At this time, the 1D convolution kernel could be represented by $kernal \in R^{D\times 3}$, where D is the number of input and output channels, 3 is the convolution kernel size. We perform the convolution on the dimension of 3, take the time dimension D as the number of input and output channels of convolution, and broadcast the convolution to the variable dimension C.

The output of convolution is the importance of each domain, which is compressed into a weight coefficient from 0 to 1 after passing through softmax. Formally, the convolution process could be described as follows. $W = Softmax(Conv(X_{int}))$, where $Conv(\cdot)$, $Softmax(\cdot)$, X_{int}, and W represent the 1D convolution employed in dimension 3, softmax employed in dimension 3, multi-domain learning result after interaction through Transformer, and learned weight, respectively. We multiply the weight by the output of multi-domain learning correspondingly, and then sum all domains in each variable dimension to obtain the fused multi-domain learning result. Finally, we apply a linear layer to it for prediction and obtain multivariate prediction results.

C. Loss Function

$\mathcal{L} = \mathbb{E}_{\boldsymbol{x}} \frac{1}{C} \sum_{c=1}^{C} \left\| \boldsymbol{y}_{L+1:L+T}^{(c)} - \boldsymbol{x}_{L+1:L+T}^{(c)} \right\|_2^2$ is our training loss function. It calculates mean squared error (MSE) loss to constrain the predictions to be close to the ground truth. Where $x_{L+1:L+T}^{(c)}$, $y_{L+1:L+T}^{(c)}$, c, and C represents future T time steps $(x_{L+1}, \ldots, x_{L+T})$ in training set, predicted T time steps, c-th variate, and the number of variates, respectively. $c \in [1, C]$.

III. Experiments

A. Experimental Setup

1) Datasets: Following baselines [28-35, 39], we use 9 authoritative public datasets [30-31, 40-42] listed in Table 1 for evaluation.

2) Baselines: Our baselines are numerous high-impact SOTA methods of multivariate long-term forecasting, including Transformer-based methods [28-32, 35-36], Linear-based methods [34, 39], and CNN-based methods [33].

3) Implementation Details: Following baselines, the look-back window sizes are 36 for ILI and 96 for other datasets. MoME is implemented using PyTorch on three 3090 GPUs. Please review the complete hyper-parameters and implementation details in our published codes.

B. Quantitative Experiment

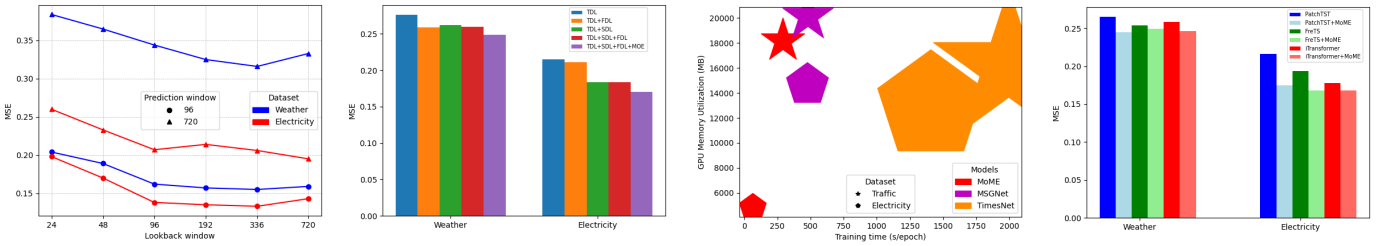
Following baselines, prediction window $T \in \{96, 192, 336, 720\}$. We calculate the average MSE and mean absolute error (MAE) and show the results in Table 2. It could be seen that in both the large scale datasets Traffic (862 variables) and Electricity (321 variables) and the small scale dataset ETT (7 variables), MoME is overall ahead of the existing single-domain learning SOTA methods. This shows that multi-domain learning could effectively mine information from multiple perspectives and capture comprehensive patterns. SOTA with only spatial domain learning, such as iTransformer, although it performs excellently on largest-scale Traffic dataset, it performs poorly on small scale datasets due to the lack of ability to extract other features such as frequency domain. Therefore, multi-domain learning and MoE could make the model's performance more balanced and powerful.

TABLE I
STATISTICS OF BENCHMARK DATASETS.

Datasets	Weather [40]	Traffic [41]	Electricity [30]	Exchange-Rate [42]	ILI [30]	ETTh1 [31]	ETTh2 [31]	ETTm1 [31]	ETTm2 [31]
Variates	21	862	321	8	7	7	7	7	7
Timesteps	52696	17544	26304	7,588	966	17420	17420	69680	69680
Granularity	10min	1hour	1hour	1day	1week	1hour	1hour	5min	5min

TABLE II
MULTIVARIATE LONG-TERM FORECASTING RESULTS OF MOME AND SOTA METHODS. THE BEST RESULTS ARE IN BOLD.

Models	MoME		iTransformer[36]		MSGNet[35]		FreTS[34]		PatchTST[32]		TimesNet[33]		DLinear[39]		FEDformer[28]		Pyraformer[29]		Autoformer[30]		Informer[31]	
Venue	-		ICLR 24		AAAI 24		NeurIPS 23		ICLR 23		ICLR 23		AAAI 23		ICML 22		ICLR 22		NeurIPS 21		AAAI 21	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	**0.246**	**0.274**	0.258	0.278	0.249	0.278	0.254	0.305	0.265	0.286	0.259	0.287	0.265	0.317	0.309	0.360	0.946	0.717	0.338	0.382	0.634	0.548
Traffic	0.442	**0.275**	**0.428**	0.282	0.661	0.384	0.518	0.321	0.529	0.341	0.620	0.336	0.625	0.383	0.610	0.376	0.878	0.469	0.628	0.379	0.764	0.416
Electricity	**0.168**	**0.261**	0.178	0.270	0.194	0.300	0.194	0.286	0.216	0.318	0.193	0.295	0.212	0.300	0.214	0.327	0.379	0.445	0.227	0.338	0.311	0.397
Exchange	**0.349**	**0.378**	0.360	0.403	0.399	0.430	0.486	0.464	0.352	0.397	0.416	0.443	0.354	0.414	0.519	0.500	1.913	1.159	0.613	0.539	1.550	0.998
ILI	**1.575**	**0.793**	1.856	0.873	3.789	1.353	2.349	1.032	1.633	0.801	2.139	0.931	2.616	1.090	2.847	1.144	7.635	2.050	3.006	1.161	5.137	1.544
ETTh1	**0.440**	**0.434**	0.454	0.448	0.452	0.452	0.475	0.465	0.516	0.484	0.458	0.450	0.456	0.452	0.440	0.460	0.827	0.703	0.496	0.487	1.040	0.795
ETTh2	**0.366**	**0.395**	0.383	0.407	0.396	0.417	0.506	0.490	0.391	0.412	0.414	0.427	0.559	0.515	0.437	0.449	0.826	0.703	0.450	0.459	4.431	1.729
ETTm1	**0.398**	**0.403**	0.407	0.410	**0.398**	0.411	0.437	0.452	0.406	0.408	0.400	0.406	0.403	0.407	0.448	0.452	0.691	0.607	0.588	0.517	0.961	0.734
ETTm2	**0.285**	**0.329**	0.288	0.332	0.288	0.330	0.320	0.366	0.290	0.334	0.291	0.333	0.341	0.401	0.305	0.349	1.502	0.869	0.327	0.371	1.410	0.810



(a) MSE variation as lookback window increases. (b) Ablation study results of MoME. (c) Spatio-temporal overhead and larger markers indicate more parameters. (d) Evaluation results of generalization performance.

Fig. 2. (a), (b), (c), and (d) showcases the experimental results of increasing the lookback window, ablation study, overhead, and generalization performance, respectively.

Additionally, we measure the MSE under various look-back window, with the results shown in Fig. 2(a). The result shows that the error generally decreases with the increase of look-back window and reaches the optimum at 336 or 720. This indicates that MoME could efficiently mine richer multi-domain features from longer input sequences to achieve more accurate predictions. It also proves that MoME could achieve excellent long-term prediction ability through longer input sequences.

C. Ablation Study

We conduct an ablation study as shown in Fig. 2(b), which calculates the average MSE of four prediction windows. Among them, 'TDL', 'FDL', 'SDL', and 'MOE' represents time domain, frequency domain, spatial domain, and MoE learning respectively. As more and more domain learning is added, the error shows a step-wise decrease on the whole, indicating that each domain and MoE learning designed by us is effective. Note that the orange and green parts are verified separately and are not related. Therefore, they will not show a progressive decline.

D. Resource Overhead Analysis

Fig. 2(c) shows the comparison of spatio-temporal costs between MoME and SOTA methods, where larger markers represent models with more parameters. On these two largest-scale datasets, almost all three indicators of MoME are superior to the compared methods, proving its high efficiency and ideal overhead.

E. Evaluation of Generalization Performance

To evaluate the generalization performance of MoME, we add multiple domains and MoE learning to the SOTA methods that only learn a single domain. As shown in Fig. 2(d), the blue, green, and red parts respectively represent the SOTA methods for learning the time domain, frequency domain, and spatial domain. After completing the missing domain learning, the prediction error of each SOTA method has been significantly reduced. This proves that the multi-domain and MoE learning of MoME have strong generalization ability and could help existing methods significantly improve prediction accuracy.

IV. CONCLUSION

To achieve complete information mining and accurate prediction, we propose the Mixture of Multi-Domain Experts (MoME) for multivariate long-term series forecasting. MoME emphasizes multi-perspective information mining and fusion through converting time series into the frequency and spatial domains to learn their respective representations. Extensive experiments on 9 benchmarks verify the superiority of MoME in multivariate long-term prediction tasks.

REFERENCES

[1] Shumway R H, Stoffer D S, Shumway R H, et al. Arima models [J]. Time series analysis and its applications: with R examples, 2017: 75-163.

[2] Vroman P, Happiette M, Rabenasolo B. Fuzzy adaptation of the holt–winter model for textile sales-forecasting[J]. Journal of the Textile Institute, 1998, 89(1): 78-89.

[3] https://github.com/facebook/prophet[Z].

[4] Yule G U. Vii. on a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers[J]. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 1927, 226(636-646): 267-298.

[5] Walker G T. On periodicity in series of related terms[J]. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 1931, 131 (818): 518-532.

[6] Engte R F. Autoregressive conditional heteroscedasticity with estimates of the variance of uk inflations[J]. Econometrica, 1982, 59: 987-1007.

[7] Bollerslev T. Generalized autoregressive conditional heteroskedasticity[J]. Journal of econometrics, 1986, 31(3): 307-327.

[8] Ord J K, Koehler A B, Snyder R D. Estimation and prediction for a class of dynamic nonlinear statistical models[J]. Journal of the American Statistical Association, 1997, 92(440): 1621-1629.

[9] Mélard G, Pasteels J M. Automatic arima modeling including interventions, using time series expert software[J]. International Journal of Forecasting, 2000, 16(4): 497-508.

[10] Cortes C, Vapnik V. Support-vector networks[J]. Machine learning, 1995, 20: 273-297.

[11] Elsayed S, Thyssens D, Rashed A, et al. Do we really need deep learning models for time series forecasting?[A]. 2021.

[12] Zahari A, Jaafar J. A novel approach of hidden markov model for time series forecasting[C]//Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication. 2015: 1-5.

[13] Hong W C. Hybrid evolutionary algorithms in a svr-based electric load forecasting model[J]. International Journal of Electrical Power & Energy Systems, 2009, 31(7-8): 409-417.

[14] Prokhorenkova L, Gusev G, Vorobev A, et al. Catboost: unbiased boosting with categorical features[J]. Advances in neural information processing systems, 2018, 31.

[15] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree[J]. Advances in neural information processing systems, 2017, 30.

[16] Chen T, Guestrin C. Xgboost: A scalable tree boosting system [C]//Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016: 785-794.

[17] Hyndman R J, Khandakar Y. Automatic time series forecasting: the forecast package for r[J]. Journal of statistical software, 2008, 27: 1-22.

[18] Hyndman R, Koehler A B, Ord J K, et al. Forecasting with exponential smoothing: the state space approach[M]. Springer Science & Business Media, 2008.

[19] Oreshkin B N, Carpov D, Chapados N, et al. N-beats: Neural basis expansion analysis for interpretable time series forecasting [A]. 2019.

[20] Olivares K G, Challu C, Marcjasz G, et al. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with nbeatsx[J]. International Journal of Forecasting, 2023, 39(2): 884-900.

[21] Oreshkin B N, Amini A, Coyle L, et al. Fc-gaga: Fully connected gated graph architecture for spatio-temporal traffic forecasting[C]//Proceedings of the AAAI conference on artificial intelligence: Vol. 35. 2021: 9233-9241.

[22] Salinas D, Flunkert V, Gasthaus J, et al. Deepar: Probabilistic forecasting with autoregressive recurrent networks[J]. International journal of forecasting, 2020, 36(3): 1181-1191.

[23] Sherstinsky A. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network[J]. Physica D: Nonlinear Phenomena, 2020, 404: 132306.

[24] Yu Y, Si X, Hu C, et al. A review of recurrent neural networks: Lstm cells and network architectures[J]. Neural computation, 2019, 31(7): 1235-1270.

[25] Dey R, Salem F M. Gate-variants of gated recurrent unit (gru) neural networks[C]//2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). IEEE, 2017: 1597-1600.

[26] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[27] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[28] Zhou T, Ma Z, Wen Q, et al. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting[C]// International conference on machine learning. PMLR, 2022: 27268-27286.

[29] Liu S, Yu H, Liao C, et al. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting[C]//International conference on learning representations. 2021.

[30] Wu H, Xu J, Wang J, et al. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting [J]. Advances in neural information processing systems, 2021, 34: 22419-22430.

[31] Zhou H, Zhang S, Peng J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]// Proceedings of the AAAI conference on artificial intelligence: Vol. 35. 2021: 11106-11115.

[32] Nie Y, Nguyen N H, Sinthong P, et al. A time series is worth 64 words: Long-term forecasting with transformers[A]. 2022.

[33] Wu H, Hu T, Liu Y, et al. Timesnet: Temporal 2d-variation modeling for general time series analysis[A]. 2022.

[34] Yi K, Zhang Q, Fan W, et al. Frequency-domain mlps are more effective learners in time series forecasting[J]. Advances in Neural Information Processing Systems, 2024, 36.

[35] Cai W, Liang Y, Liu X, et al. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting[C]// Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 38. 2024: 11141-11149.

[36] Liu Y, Hu T, Zhang H, et al. itransformer: Inverted transformers are effective for time series forecasting[A]. 2023.

[37] Kim T, Kim J, Tae Y, et al. Reversible instance normalization for accurate time-series forecasting against distribution shift[C]// International Conference on Learning Representations. 2021.

[38] Fan W, Wang P, Wang D, et al. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting[C]// Proceedings of the AAAI conference on artificial intelligence: Vol. 37. 2023: 7522-7529.

[39] Zeng A, Chen M, Zhang L, et al. Are transformers effective for time series forecasting?[C]//Proceedings of the AAAI conference on artificial intelligence: Vol. 37. 2023: 11121-11128.

[40] Max-Planck-Institut for Biogeochemistry. Weather dataset [EB/OL]. https://www.bgc-jena.mpg.de/wetter/.

[41] Traffic dataset[EB/OL]. http://pems.dot.ca.gov/.

[42] Lai G, Chang W C, Yang Y, et al. Modeling long-and short-term temporal patterns with deep neural networks[C]// The 41st international ACM SIGIR conference on research & development in information retrieval. 2018: 95-104.