MGTDGraph: Multi-granularity Graph Attention Networks for Multivariate Long-Term Time Series Forecasting

Shumin Tan and Yuexian Zou(✉)

ADSPLAB, School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, China
2201212851@stu.pku.edu.cn

Abstract. Conventional time series forecasting approaches often face challenges in modeling intricate multivariate correlations and fail to jointly capture local temporal dynamics as well as long-range dependencies. To overcome these issues, we propose a novel forecasting framework that leverages multi-granularity feature extraction based on graph neural network (GNN). Our approach integrates information at the node, edge, and subgraph levels to construct a comprehensive representation that encompasses both fine-grained and global structures in multivariate time series data. A Graph Attention Network (GAT) is employed to adaptively assign importance weights between nodes and their neighbors, enabling the model to effectively capture complex spatial–temporal interactions. Extensive experiments conducted on four benchmark datasets across multiple prediction horizons demonstrate that our method consistently outperforms existing baselines in predictive accuracy. Beyond accuracy improvements, the model's ability to represent structural intricacies of time series data enhances its applicability to a wide range of forecasting scenarios across diverse domains.

Keywords: Time Series Forecasting · Graph neutral network · Multi-Granularity Feature · Long-term Time Series Forecasting

1 Introduction

Time series forecasting plays a pivotal role in facilitating decision-making, risk management and operational optimization across diverse domains. With the rapid growth of interconnected infrastructures like smart cities, smart grids, and the IoT, the volume and complexity of time series data have increased dramatically, which highlights the need for more advanced approaches capable of capturing rich multivariate dependencies [1].

Time series data, characterized by temporal dependencies, involves observations ordered over time [2]. Forecasting relies on modeling these dependencies to predict future values based on historical data. Traditional methods lack scalability and applicability, while deep learning methods face challenges such as poor interpretability, overfitting, and high computational costs [3]. Extracting and leveraging the intricate structural features of time series data remains a key challenge [4].

D.-S. Huang et al. (Eds.): ICIC 2025, LNAI 15863, pp. 39–49, 2025.
https://doi.org/10.1007/978-981-95-0009-3_4

Transformer-based models like Informer [5] and Reformer [6] have shown significant advantages in capturing global features of time series. However, they encounter challenges in capturing local features, particularly high-frequency information within time series. The key challenges in multivariate long-term forecasting tasks can be summarized as follows:

- **Limitations in modeling nonlinear relationships in multivariate long sequences**. Traditional methods struggle to capture the complex nonlinear relationships among variables in multivariate time series. These methods also face challenges in handling long-term forecasting tasks, where issues such as high computational complexity and reduced accuracy often arise.
- **Lack of unified representation of global and local features**. Many existing approaches face limitations in simultaneously capturing local dynamic features and global dependencies in time series. Additionally, some methods are insufficient for addressing diverse characteristics in time series, such as periodicity and trends.

To overcome these issues, we propose MGTDGraph (Multi-Granularity Time-Dim Graph), which adopts a multi-granularity feature extraction strategy to comprehensively capture both local temporal dynamics and long-range dependencies in time series data. By performing detailed feature extraction across different granularity levels of subsequences, MGTDGraph enables in-depth analysis and understanding from local to global perspectives. Our contributions are as follows:

- **A novel time series forecasting method based on multi-granularity feature extraction.** This paper introduces a method that extracts time series features at three levels (node-level, edge-level, and subgraph-level) and integrates these multi-granularity features to comprehensively represent the local dynamics and global dependencies of time series data. By fusing multi- granularity features, the model provides richer hierarchical information for each node, enabling it to better capture complex patterns and multi-scale dependencies in graphs.
- **Dynamic modeling of nonlinear multivariate relationships.** By incorporating GAT, this paper applies self-attention mechanisms to time series forecasting, dynamically assigning importance weights to neighboring nodes. This effectively captures the nonlinear dependencies among variables in multivariate time series. GAT adaptively assigns weights to different neighbors through self-attention, overcoming the limitations of fixed weights in traditional convolution methods. By integrating multi-granularity features and attention mechanisms, the model incorporates both global and local information into node representations, enhancing the performance of graph tasks.
- The experimental results demonstrate that the proposed method significantly outperforms mainstream methods, showcasing its strong adaptability in modeling complex scenario.

The remainder of this paper is organized as follows: related work Sect. 2, methodology Sect. 3, experiments Sect. 4, and conclusions Sect. 5.

2 Related Work

Time series forecasting has been widely explored using traditional statistical models, machine learning approaches, deep learning architectures, and more recently, graph-based neural networks.

2.1 Time Series Forecasting Based on Statistical and Machine Learning Approaches

Statistical methods forecast time series by modeling temporal evolution through mathematical parameters. These methods, such as AR [8], ARIMA [9] and VAR [10] are computationally efficient and interpretable, but rely on strong linearity assumptions and struggle to model complex multivariate and nonlinear time series.

Machine learning models such as ICA-SVR [11], RHWFTS-ICA [12], and WLGP [13] leverage feature decomposition and hybrid modeling to capture nonlinear dependencies in time series data. While these approaches improve over purely statistical models, they often rely on manual design choices and exhibit limited robustness when facing highly non-stationary or structurally complex sequences.

2.2 Time Series Forecasting Models Based on Deep Learning

Deep learning models have demonstrated strong capabilities in modeling complex temporal dependencies, particularly through architectures like RNNs and LSTMs. Early works such as ICA-RNN [14], N-BEATS-RNN [15], EMD-LSTM [16], and DLSTM [17] explored hybrid designs combining signal decomposition, recurrent structures, and evolutionary optimization. More recently, Transformer-based models have gained prominence in this area. Autoformer [18] introduces a decomposition-based autocorrelation mechanism to capture periodicity in long sequences, while FEDformer [19] enhances this by integrating Fourier analysis to model frequency-domain representations. Despite their strengths, these models often emphasize global dependencies, which can hinder the learning of local temporal dynamics and lead to information loss when aggressively reducing complexity.

2.3 Time Series Forecasting Models Based on Graph Neural Networks

To address spatial dependencies in multivariate time series, such as traffic flow prediction, recent work has turned to Graph Neural Networks (GNNs), which effectively encode relational structures among variables. DCRNN [20] employs diffusion convolutions within a recurrent framework to model spatiotemporal dependencies. ASTGCN [21] extends this by introducing attention mechanisms across multiple temporal scales, including recent, periodic, and long-term dependencies. GMAN [22] and related models [23] adopt spatiotemporal attention blocks and adaptive adjacency learning to capture dynamic spatial structures. PDFormer [24], a recent advancement, incorporates propagation delay modeling and spatial self-attention to better represent both short-range and long-range dependencies. These methods effectively capture spatial correlations, but often lack mechanisms for integrating multi-granularity temporal representations.

3 Method

We propose a graph-based time series forecasting framework that extracts multi-granularity features at the node, edge, and subgraph levels to model both local and global temporal dependencies. The overall structure of our proposed method is illustrated in Fig. 1.

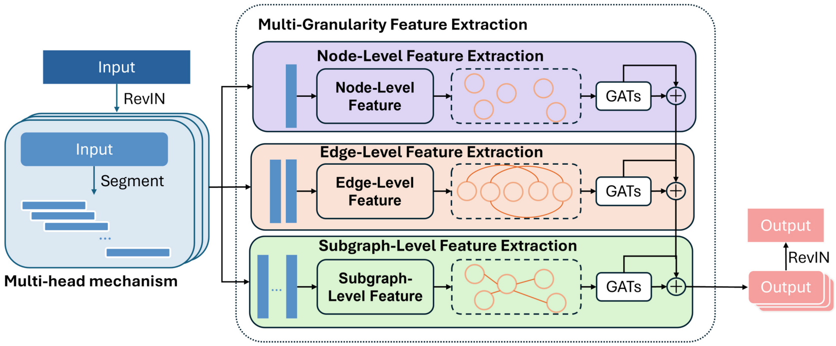

Fig. 1. Overview of MGTDGraph.

3.1 Problem Definition

Let n be the length of the input time series. Denote it by

$$\boldsymbol{\chi} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \tag{1}$$

where each $\boldsymbol{x_i} \in R^d$ is a d-dimentional vector of variables at time index i. Our goal is to predict n' future steps:

$$\lambda = \left\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{\boldsymbol{n}'}\right\}, \boldsymbol{y_i} \in \mathbb{R}^d \tag{2}$$

such that $\boldsymbol{\lambda} = \mathcal{G}(\boldsymbol{\chi})$, with $\mathcal{G}(\cdot)$ being our learnable mapping.

3.2 Multi-granularity Feature Extraction

To extract sequence features from node-level, edge-level, and subgraph-level, we apply multi-granularity feature extraction.

In real-world datasets, the overall data distribution changes over time. Therefore, a Reversible Instance Normalization (RevIN) [28] module is used to dynamically normalize the input sequences. Let μ and σ be the mean and standard deviation of each block or entire sequence. For each time step x_i:

$$x_i' = \frac{x_i - \mu}{\sigma}, \boldsymbol{x'}_i \leftarrow \eta \boldsymbol{x'}_i + \zeta \tag{3}$$

where η and ζ are learnable scalars.

After applying RevIN, we map the normalized input to an embedding space. The transformation is achieved through a simple linear projection layer:

$$H = \sigma(\boldsymbol{W}\boldsymbol{\chi} + \boldsymbol{b}), \tag{4}$$

where $W \in R^{d \times d_{model}}$ is a trainable weight matrix, b is a bias term, and σ(·) is the activation function. To simplify notation, we do not introduce a seperate symbol for the normalized sequence; thus, the "input" here is implicitly assumed to be the RevIN-transformed data.

Node-Level Feature Extraction: Node-level features focus on individual node attributes and the relationships between each node and its neighboring nodes. Let *hi* be the embedding for node *i*. We update it by aggregating information from its neighbors $\mathscr{N}$(i):

$$\boldsymbol{h_i^n} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \frac{1}{\sqrt{|\mathcal{N}(i)|}} \boldsymbol{W h_j} + \boldsymbol{W h_i}\right) \tag{5}$$

where σ(·) is the ReLU activation.

Edge-Level Feature Extraction: By focusing on the attributes of edges or the relationships between the connected nodes, edge-level feature extraction combines the features of two nodes, represented as:

$$\boldsymbol{h_{e^{ij}}} = \boldsymbol{h_i} \cdot \boldsymbol{h_j} \tag{6}$$

where h_i and h_j are the embeddings of node i and node j, respectively.

Subgraph-Level Feature Extraction: The sequence is divided into overlapping segments of approximately equal size, with each segment partially overlapping the next to improve model generalization and mitigate overfitting. Let L_b be the block size and O be the overlap size. We create

$$\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B_M} = Split(\boldsymbol{H}, L_b, O), \tag{7}$$

where each $B_m \in R^{L_b \times d_{model}}$, and M is the total number of blocks.

This process focuses on local structures or subgraph features by aggregating the features of all nodes and edges within a subgraph to generate its feature representation. Within each block B_m, we can directly pool across time steps to form an initial subgraph embedding:

$$\boldsymbol{h_m^{(g)}} = AvgPool(\boldsymbol{B_m}) \tag{8}$$

yielding $h_m^{(g)} \in R^{d_{model}}$.

3.3 Graph Attention and Multi-head Fusion

The fused feature vectors are then used as input node features for GAT.

After establishing node-level and edge-level features, we fuse them along with subgraph-level embeddings using a graph attention mechanism. Let $h_i, h_{e^{ij}}, h_m^{(g)}$ be node-level, edge-level, and subgraph-level features, respectively. We compute a unified representation for node i:

$$\boldsymbol{u_i} = \sum_{j \in \mathcal{N}(i)} \left(\boldsymbol{\alpha_{ij h_i}} + \boldsymbol{\beta_{ij h_{e^{ij}}}} + \boldsymbol{\gamma_{ij h^{(g)}_{m(i)}}} \right), \tag{9}$$

where $m(i)$ indicates which subgraph node i belong to, and $\alpha_{ij}, \beta_{ij}, \gamma_{ij}$ are attention weights.

By fusing features, the multi-layer information of each node (including node-level, edge-level, and subgraph-level features) can be obtained. The attention weight between node i and its neighboring node j is expressed as:

$$\boldsymbol{\alpha_{ij}} = \frac{exp\left(LeakyReLU\left(\boldsymbol{\alpha^T}\left[\boldsymbol{W}\,\boldsymbol{h_i} \parallel \boldsymbol{W}\,\boldsymbol{h_j}\right]\right)\right)}{\sum_{k \in \mathcal{N}(i)} exp\left(LeakyReLU\left(\boldsymbol{\alpha^T}\left[\boldsymbol{W}\,\boldsymbol{h_i} \parallel \boldsymbol{W}\,\boldsymbol{h_k}\right]\right)\right)} \tag{10}$$

where $\parallel$ denotes concatenation. B_{ij} and γ_{ij} are expressed in a similar way.

3.4 Multi-Head Mechanism

To capture different aspects of adjacency, we apply multiple attention heads $l \in \{1, 2, \ldots, heads\}$ in parallel and then concatenate their outputs:

$$\boldsymbol{u_i^{(final)}} = \parallel \left\{ \boldsymbol{u_i^{(l)}} \right\}_{l=1}^{heads} \tag{11}$$

where $\parallel$ means concatenation across heads. Residual connections and feedforward layers are appended for deeper modeling.

After the multi-head graph attention layers, we map $u_i^{(final)}$ to the output dimention with a linear projection. Let h_{out} be the final representation in $R^{n' \times d}$. We apply the inverse of RevIN to recover the original scale:

$$\boldsymbol{h_{out}^{(rev)}} = \frac{\boldsymbol{h_{out}} - \zeta}{\eta} \cdot \sigma + \mu \tag{12}$$

yielding the predicted sequence

$$P = \left\{ \boldsymbol{p}_1, \ldots, \boldsymbol{p_{n'}} \right\}, \boldsymbol{p_i} \in R^d \tag{13}$$

We define the L2 loss between P and ground truth Y as:

$$L = \sum_{i=1}^{n'} \left| \left| \boldsymbol{p_i} - \boldsymbol{y_i} \right| \right|^2 \tag{14}$$

4 Experiments

4.1 Dataset

We evaluate our model on four widely used public datasets: **ETT (Electricity Transformer Temperature)** [5], **ILI (Influenza-Like Illness)**, **Electricity** [29], and **Weather**. These datasets cover diverse domains and sampling resolutions, ranging from hourly and sub-hourly industrial sensor data to weekly influenza statistics and high-frequency meteorological records.

4.2 Baseline

We compare MGTDGraph with several representative baselines, including Transformer-based models (Informer [5], Autoformer [18], FEDformer [19], Robformer [27], PRRegNet [25], TimesNet [26]) and a recent graph-based method (ESG [24]). These baselines reflect SOTA designs in long-term forecasting, frequency-domain modeling, and graph learning. Model implementations and training settings follow the original papers for fair comparison.

4.3 Implementation Details

We employed a sliding window approach to generate data samples. The stride is set to 1 while the window size is equal to the sum of the input length L_x and prediction lengthL_p. The input length was fixed at $L_x = 96$ for all datasets. For ETTm1, Weather, and Electricity datasets,$L_p \in \{96, 192, 336, 720\}$, while for the ILI dataset$L_p \in \{24, 36, 48, 60\}$. The first segment L_x was used as input, and the second segment (L_p) served as the ground truth. Prediction errors were evaluated using MSE and MAE.

The model is optimized using the ADAM algorithm, initialized with a learning rate of 0.001. Training is performed with a batch size of 64, and the attention mechanism comprises 8 heads, each with a hidden size of 64. Experiments were repeated three times, with average results reported.

4.4 Main Results

The effectiveness of MGTDGraph and baselines was evaluated on four $L_P$values for all datasets.

MSE and MAE: As shown in Table 1, MGTDGraph consistently achieves the lowest MSE and MAE across most datasets and prediction horizons, especially on ETTm1, Weather, and Electricity. While Robformer and PRRegNet perform competitively in certain cases, particularly PRRegNet on Weather and Electricity, MGTDGraph remains the most robust overall. TimesNet shows strong performance on the Illness dataset at shorter horizons, suggesting some task specialization. The results also indicate that all models suffer increased error as the prediction horizon extends, highlighting the inherent difficulty of long-term forecasting.

Table 1. Input multivariate time series prediction results ($L_x = 96$)

Method		MGTDGraph **(Ours)**		Informer (2021)		Autoformer (2021)		FEDformer (2022)		ESG (2022)		TimesNet (2023)		Robformer (2024)		PRRegNet (2024)	
Metric		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
ETTm1	96	**0.317**	**0.351**	0.571	0.672	0.475	0.505	0.379	0.419	0.406	0.430	0.338	0.375	0.379	0.398	0.351	0.374
	192	**0.357**	**0.349**	0.669	0.795	0.496	0.553	0.426	0.441	0.458	0.480	0.374	0.387	0.397	0.435	0.369	0.379
	336	**0.397**	**0.398**	0.871	1.212	0.537	0.621	0.445	0.459	0.534	0.567	0.410	0.411	0.413	0.459	0.398	0.400
	720	**0.432**	**0.439**	0.823	1.166	0.561	0.671	0.543	0.490	0.545	0.571	0.450	0.478	0.448	0.468	0.435	0.457
Weathe	96	0.169	0.203	0.384	0.300	0.312	0.231	0.217	0.296	0.170	0.238	0.278	0.214	0.231	0.296	**0.148**	**0.196**
	192	**0.217**	**0.267**	0.544	0.598	0.343	0.278	0.276	0.336	0.226	0.295	0.256	0.286	0.268	0.301	0.218	**0.267**
	336	**0.274**	**0.317**	0.523	0.578	0.378	0.335	0.339	0.380	0.286	0.339	0.292	0.326	0.354	0.364	0.275	0.321
	720	**0.332**	**0.378**	0.741	1.059	0.436	0.429	0.403	0.428	0.384	0.404	0.335	0.381	0.381	0.401	0.333	0.379
Elec-	96	**0.191**	**0.287**	0.368	0.274	0.312	0.197	0.193	0.308	-	-	0.395	0.459	0.278	0.394	0.201	0.323
	192	**0.200**	**0.231**	0.386	0.296	0.321	0.208	0.201	0.315	-	-	0.208	0.267	0.264	0.301	0.221	0.279
	336	**0.201**	**0.278**	0.394	0.300	0.328	0.213	0.214	0.329	-	-	0.302	0.320	0.294	0.335	0.235	0.328
	720	**0.231**	**0.293**	0.439	0.373	0.352	0.245	0.246	0.355	-	-	0.331	0.371	0.301	0.301	0.238	0.298
Illness	24	**0.931**	**1.348**	1.360	4.388	1.345	3.825	2.203	0.963	1.286	1.583	1.433	0.945	1.245	1.399	0.936	1.389
	36	**0.962**	**1.684**	1.391	4.651	1.216	3.319	2.272	0.976	1.457	1.610	1.820	1.094	1.036	1.847	0.969	1.783
	48	**0.972**	**2.068**	1.419	4.581	1.122	2.854	2.209	0.981	1.545	1.616	2.187	1.232	1.159	2.340	0.980	2.215
	60	**0.973**	**1.634**	1.432	4.583	1.232	3.227	2.545	1.061	1.651	1.685	1.655	1.583	1.210	1.841	0.993	1.674

Number of Heads Analysis: As shown in Table 2, using 8 heads consistently yields the best performance across all datasets, particularly on ETTm1. Both too few and too many heads lead to higher errors, reflecting a trade-off between representational capacity and overfitting risk. Errors also increase with longer prediction horizons, aligning with the general difficulty of long-term forecasting.

Efficiency Analysis: As shown in Fig. 2, MGTDGraph and TimesNet consistently demonstrate the best efficiency across varying prediction lengths, with lower memory usage and moderate computation time. In contrast, Informer and Autoformer incur steep computational costs as the prediction horizon increases, while Robformer and PRRegNet show high overhead, limiting their suitability for real-time applications. Overall, MGTDGraph achieves a strong balance between predictive performance and resource efficiency.

Table 2. Results of the number of bulls model on different data sets

Number of head		ETTm1				Weather			
		96	192	336	720	96	192	336	720
1	MSE	0.135	0.256	**0.248**	0.197	0.208	0.301	0.215	0.191
	MAE	0.230	0.297	0.310	0.167	0.231	0.298	0.197	0.197
4	MSE	0.174	0.216	0.316	0.172	0.241	0.249	0.194	0.135
	MAE	0.215	0.254	0.308	0.184	0.248	0.289	0.197	0.164
8	MSE	**0.097**	**0.206**	**0.248**	**0.151**	**0.201**	**0.241**	**0.135**	**0.091**
	MAE	**0.167**	**0.219**	**0.267**	**0.162**	**0.224**	**0.235**	**0.184**	**0.097**
16	MSE	0.138	0.264	0.297	0.173	0.294	0.294	0.194	0.131
	MAE	0.253	0.268	0.307	0.191	0.281	0.279	**0.184**	0.120
Number of head		Electricity				Illness			
		96	192	336	720	96	192	336	720
1	MSE	0.341	0.197	0.168	0.192	0.132	0.167	0.120	0.197
	MAE	0.276	**0.189**	0.209	0.172	0.134	0.167	0.146	0.184
4	MSE	0.297	0.206	0.219	0.167	0.128	0.204	0.101	0.164
	MAE	0.249	0.211	0.209	0.206	0.143	0.197	0.113	0.138
8	MSE	**0.207**	**0.192**	**0.134**	**0.137**	**0.098**	**0.134**	**0.097**	**0.125**
	MAE	**0.216**	**0.189**	**0.184**	**0.171**	**0.097**	**0.105**	**0.082**	**0.114**
16	MSE	0.308	0.206	0.187	0.198	0.130	0.176	0.130	0.164
	MAE	0.301	0.219	0.210	0.189	0.108	0.171	0.111	0.161

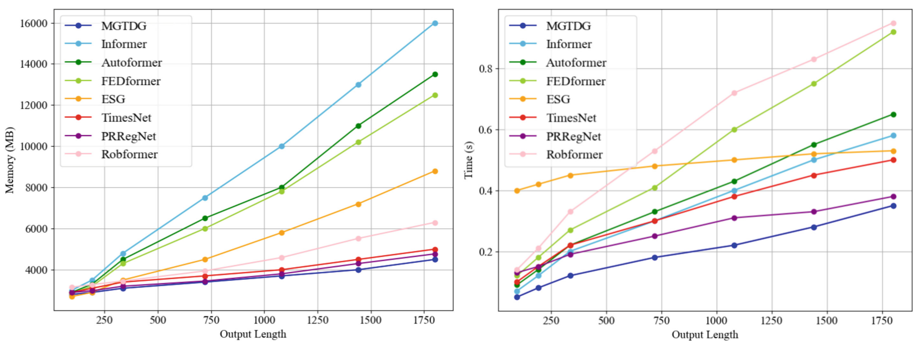

Fig. 2. Efficiency analysis. The results show that MGTDGraph uses significantly less memory than current technology and runs faster. (a) Memory efficiency analysis. (b) Operational effi-ciency analysis.

5 Conclusion

We present MGTDGraph, a novel approach for time series forecasting that leverages multi-granularity feature extraction and Graph Attention Networks (GAT). By jointly modeling node-, edge-, and subgraph-level information, our method captures both local temporal patterns and global dependencies in multivariate data. Combined with sliding windows and a multi-head mechanism, the model consistently outperforms existing baselines across multiple public datasets, particularly in scenarios with strong inter-variable coupling. Furthermore, the approach demonstrates robust performance in handling complex long-range sequences and diverse temporal structures. Future work may explore integrating external priors, such as spatial topology or expert domain knowledge, to further enhance forecasting accuracy.

Acknowledgement. This work is supported by Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (Grant No.2024B1212010006).

References

1. Arora, S., Kolambe, M.: Forecasting the future: a comprehensive review of time series prediction techniques. J. Electric. Syst. **20**(2s), 575–586 (2024)
2. Wang, X., Smith, K., Hyndman, R.: Characteristic-based clustering for time series data. Data Min. Knowl. Disc. **13**(3), 335–364 (2006)
3. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: a review of machine learning interpretability methods. Entropy **23**(1) (2021)
4. Längkvist, M., Karlsson, L., Loutfi, A.: A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recogn. Lett. **42**, 11–24 (2014)
5. Zhou, H., et al.: Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 11106–11115 (2021)
6. Kitaev, N., Kaiser, L., Levskaya, A.: Reformer: the efficient transformer. In: International Conference on Learning Representations (2020)
7. Yule, G.U.: On a method of investigating periodicities in disturbed series, with special reference to wolfer's sunspot numbers. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character **226**, 267–298 (1927)
8. Box, G.E.P., Jenkins, G.M.: Some recent advances in forecasting and control. J. R. Stat. Soc. Ser. C **17**(2), 91–109 (1968). https://doi.org/10.2307/2985674
9. Sims, C.A.: Macroeconomics and reality. Econometrica **48**(1), 1–48 (1980)
10. Fan, L., Pan, S., Li, Z., Li, H.: An ICA-based support vector regression scheme for forecasting crude oil prices. Technol. Forecast. Soc. Chang. **112**, 245–253 (2016). https://doi.org/10.1016/j.techfore.2016.04.027
11. Enayatifar, R., Sadaei, H.J., Abdullah, A.H., Gani, A.: Imperialist competitive algorithm combined with refined high-order weighted fuzzy time series (RHWFTS–ICA) for short term load forecasting. Energy Convers. Manage. **76**, 1104–1116 (2013)
12. Ravansalar, M., Rajaee, T., Kisi, O.: Wavelet-linear genetic programming: a new approach for modeling monthly streamflow. J. Hydrol. **549**, 461–475 (2017)
13. Madan, R., Mangipudi, P.S.: Predicting computer network traffic: a time series forecasting approach using dwt, arima and rnn. In: 2018 Eleventh International Conference on Contemporary Computing (IC3). pp. 1–5 (2018)

14. Sbrana, A., Debiaso~Rossi, A.L., Coelho~Naldi, M.: N-beats-rnn: deep learning for time series forecasting. In: 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 765–768 (2020). https://doi.org/10.1109/ICMLA51294.2020.00125
15. Cao, J., Li, Z., Li, J.: Financial time series forecasting model based on CEEMDAN and LSTM. Physica A **519**, 127–139 (2019)
16. Sagheer, A., Kotb, M.: Time series forecasting of petroleum production using deep LSTM recurrent networks. Neurocomputing **323**, 203–213 (2019)
17. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021)
18. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: frequency enhanced decomposed transformer for long-term series forecasting. In: International Conference on Machine Learning. pp. 27268–27286. PMLR (2022)
19. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: International Conference on Learning Representations (2018)
20. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. AAAI'19/IAAI'19/EAAI'19. AAAI Press (2019)
21. 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. vol. 34, pp. 1234–1241 (2020)
22. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph WaveNet for deep spatial-temporal graph modeling. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. p. 1907–1913. IJCAI'19. AAAI Press (2019)
23. Jiang, J., Han, C., Zhao, W.X., Wang, J.: PDFormer: propagation delay-aware dynamic long-range transformer for traffic flow prediction. In: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence. AAAI'23/IAAI'23/EAAI'23, AAAI Press (2023)
24. Han, L., Ye, H.J., Zhan, D.C.: The capacity and robustness trade-off: revisiting the channel independent strategy for multivariate time series forecasting. IEEE Trans. on Knowl. and Data Eng. **36**(11), 7129–7142 (2024)
25. Ye, J., et al.: Learning the evolutionary and multi-scale graph structure for multivariate time series forecasting. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2296–2306. KDD '22, Association for Computing Machinery, New York, NY, USA (2022)
26. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence. AAAI'23/IAAI'23/EAAI'23. AAAI Press (2023)
27. Yu, Y., Ma, R., Ma, Z.: Robformer: a robust decomposition transformer for long-term time series forecasting. Pattern Recogn. **153**, 110552 (2024)
28. Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.H., Choo, J.: Reversible instance normalization for accurate time-series forecasting against distribution shift. In: International Conference on Learning Representations (2022)
29. Hebrail, G., Berard, A.: Individual Household Electric Power Consumption. UCI Machine Learning Repository (2006)