SCFormer: Structured Channel-wise Transformer with Cumulative Historical State for Multivariate Time Series Forecasting

Shiwei Guo[1,2,3], Ziang Chen[1,2,3], Yupeng Ma[1,3(✉)], Yunfei Han[1,3], and Yi Wang[1,3]

[1]Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumqi, China
{ypma,hanyf,wangyi}@ms.xjb.ac.cn
[2]University of Chinese Academy of Sciences, Beijing, China
{guoshiwei18,chenziang21}@mails.ucas.ac.cn
[3]Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi, China

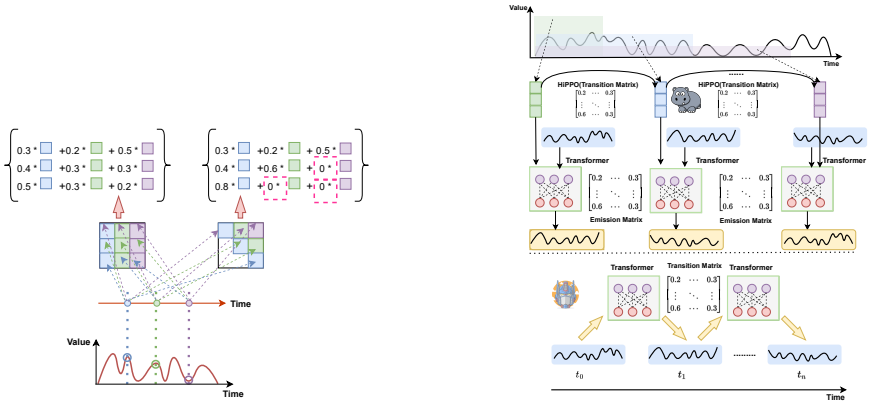
Abstract. The Transformer model has excelled in multivariate time series forecasting by using channel-wise self-attention. However, it has limitations, such as a lack of temporal constraints when extracting temporal features and ineffective use of cumulative historical data. To overcome these issues, we introduce the **S**tructured **C**hannel-wise Transformer with Cumulative Historical state (SCFormer). SCFormer adds temporal constraints to all linear transformations, including the query, key, and value matrices, as well as the fully connected layers in the Transformer. It also uses High-order Polynomial Projection Operators (HiPPO) to effectively incorporate cumulative historical data, enabling the model to utilize information beyond the look-back window during predictions. Extensive experiments on various real-world datasets show that SCFormer significantly outperforms leading baselines, demonstrating its effectiveness in improving time series forecasting. The code is publicly available at https://github.com/ShiweiGuo1995/SCFormer

Keywords: Transformer · Multivariate Time series forecasting · HiPPO.

1 Introduction

The Transformer, a versatile sequence model, has been widely applied in various fields, including NLP, computer vision, and bioinformatics. Transformer-based models have also achieved significant progress in time series forecasting [6,12,11]. Notably, recent studies have demonstrated that channel-wise Transformers [7] can effectively capture relationships among multiple temporal variables, resulting in substantial reductions in prediction errors.

However, channel-wise Transformers face two main challenges: (1) lacking a mechanism to capture cumulative historical states beyond the look-back window, and (2) using unconstrained linear transformations for temporal feature extraction, which violates fundamental temporal assumptions.



(a) (b)

Fig. 1: (a) Structured linear transformation (Right) *vs.* Linear transformation (Left). (b) Markov forecasting process (Bottom) *vs.* Forecasting process with cumulative historical state (Top).

Most current forecasting frameworks rely on a fixed-size historical window, referred to as the look-back window, to predict the next segment of a time series. This approach can be viewed as a first-order Markov process, where the forecasting model approximates the transition matrix. However, this method overlooks the cumulative historical state information accumulated prior to the look-back window, which could enhance model performance if utilized effectively. In terms of feature extraction, channel-wise Transformer employs self-attention to compute correlations among channels, while temporal features are derived through linear transformations and activation functions within the Transformer. Unlike generic sequences, time series have a fundamental temporal constraint: operations on later elements should not influence anterior ones. This assumption is grounded in the sequential nature of time series, where events occurring later cannot retroactively affect earlier events. However, applying unconstrained linear transformations to input or embedded time series violates this assumption, potentially leading to incorrect feature learning and overfitting.

To address these challenges, we employ HiPPO [3] (High-order Polynomial Projection Operators) to efficiently capture the cumulative historical state. HiPPO recursively embeds long and variable-length time series into a fixed-size state space using orthogonal polynomial bases, providing a simple yet effective memory mechanism that incorporates historical information beyond the look-back window. In this framework, the cumulative historical state functions as the memory state [1], the HiPPO matrix serves as the transition matrix to model memory updates, and the channel-wise Transformer operates as the emission matrix for forecasting. Fig. 1(b) highlights the differences between this approach and traditional forecasting methods.

We propose using structured matrices to enforce temporal constraints on linear transformations in channel-wise Transformer. For example, a triangular matrix preserves temporal order by ensuring that elements in the time series embeddings are not influenced by future values, as illustrated in Fig. 1(a). In this structure, weights assigned to successor elements are set to zero, effectively excluding them from feature computations. Moreover, since 1D convolutions inherently respect temporal order, substituting linear transformations in Transformers with 1D convolutions also enforces this constraint. As demonstrated in Chapter 2.3, multi-layer 1D convolutions are mathematically equivalent to a triangular matrix with shared parameters, and the convolution operation can be expressed as a linear transformation using such matrices. This structured design is applied to all linear operations in Transformer, including those in feed-forward layers and the query, key, and value matrices.

2 METHOD

Our method consists of two key components: (1) utilizing HiPPO to retain the cumulative historical state and (2) employing structured matrices to enforce temporal constraints on linear transformations in the channel-wise Transformer. The model integrates the look-back window and the cumulative historical state into a unified time series representation and applies a structured channel-wise Transformer to extract temporal and channel correlation features from this time series. SCFormer incorporates a single-layer fully connected network as the decoder and uses Mean Square Error (MSE) as the loss function. The architecture of SCFormer is illustrated in Fig. 2.

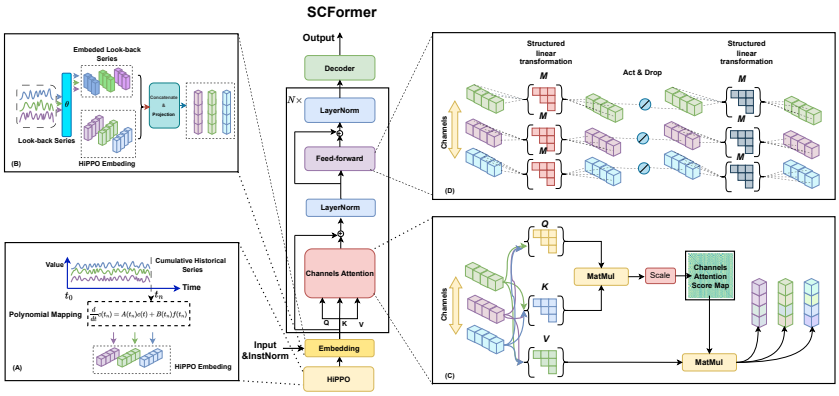


Fig. 2: Overall structure of SCFormer. (A) Cumulative historical state via HiPPO; (B) Embedding; (C) Structured channel-wise self-attention; (D) Structured feed-forward layer.

2.1 Cumulative Historical State

The accumulated history includes the entire sequence from the start of the time series up to the current look-back window. As the fixed-size look-back window slides forward, the accumulated history becomes a variable-length series, growing longer over time, which makes it challenging for the model to utilize effectively. To address this, we use HiPPO to compute the cumulative historical state, enabling the model to access richer historical information. HiPPO projects variable-length series onto orthogonal higher-order polynomial bases, embedding the cumulative historical state into a fixed-dimensional space represented by coefficients. In fact, these coefficients represent the optimal parameters when approximating the cumulative historical series using the orthogonal polynomial basis. This process can be computed efficiently using state-space equations, making it particularly suitable for variable-length sequences. For a time series $\mathbf{x}$, the cumulative historical state c_{k+1} can be computed recursively as follows:

$$\begin{aligned} c_{k+1} &= (1 - \frac{A}{k})c_k + \frac{1}{k}B\mathbf{x}_k, \\ A_{nk} &= \begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & if \quad n > k, \\ n+1 \quad if \quad n = k, \\ 0 \quad if \quad n < k \end{cases} \\ B_n &= (2n+1)^{\frac{1}{2}} \end{aligned} \tag{1}$$

Here, c_k represents the cumulative historical state of $\mathbf{x}_{:\leq k}$. Essentially c_k is the projection coefficient of the history series of $\mathbf{x}_{:\leq k}$ on the orthogonal polynomial basis. $\mathbf{x}_{:\leq k}$ represents the historical series from the beginning timestamp of the $\mathbf{x}$ up to the k-th timestamp. n represents the degree of the orthogonal polynomial in HiPPO, which also defines the dimensionality of the cumulative historical state, while k serves as the timestamp indicator. SCFormer embeds the cumulative historical state and the look-back window into a unified time series. This unified representation encapsulates both global information from the cumulative history and local dependencies from the adjacent window, offering a more comprehensive characterization of temporal patterns. For a look-back window l and its corresponding cumulative historical state (HiPPO embedding) c, this integration is achieved through concatenation and MLP.

2.2 Triangular Matrix and Temporal Constraint

In channel-wise Transformer, the self-attention mechanism involves multiple linear transformations, but these lack temporal constraints. The core issue is that standard matrix multiplication disrupts the series' temporal order, as future elements can influence past ones. For instance, for the i-th element x_i in the time series x, we calculate its corresponding feature a_i using a linear transformation. According to the matrix multiplication formula:

$$a_i = \sum_j w_{ij} x_j \tag{2}$$

It is evident that all elements in the time series x are involved in the calculation, which is unreasonable. For the set $M = \{x_j, j > i\}$, containing elements that occur after x_i, these elements should not influence the generation of a_i.

To address this issue, one approach is to set a portion of the matrix elements $W = \{w_{ij}, j > i\}$ to zero. Clearly, this results in a triangular matrix. An upper or lower triangular matrix does not affect temporal constraints, as it merely pertains to whether the growth direction of time is represented using proximal or distal methods. Without loss of generality, this paper adopts an upper triangular matrix as the structured matrix. All linear transformations in the channel-wise Transformer, including the query, key, and value matrices, should follow this structured approach. For the input $\mathbf{Z} \in \mathcal{R}^{d \times C}$, SCFormer applies a channel-wise self-attention mechanism with temporal constraints enforced by structured matrices. Specifically, it calculates the attention scores between channels as follows:

$$\begin{aligned} &\mathbf{Q}, \mathbf{K}, \mathbf{V} = \delta(\mathbf{A}\mathbf{Z} + \mathbf{a}), \delta(\mathbf{B}\mathbf{Z} + \mathbf{b}), \delta(\mathbf{C}\mathbf{Z} + \mathbf{e}) \\ &s.t \qquad \mathbf{A}_{ij}, \mathbf{B}_{ij}, \mathbf{C}_{ij} = 0, \quad if \quad i > j \end{aligned} \tag{3}$$

$$attn^i = softmax(\frac{\mathbf{Q}^i(\mathbf{K}^i)^T}{\sqrt{d/H}}) \tag{4}$$

Here, d represents the length of the embedded time series $\mathbf{Z}$, C denotes the number of channels, and $attn^i$ refers to the attention scores of the i-th head in the multi-head attention mechanism. H denotes the number of the multi-head. $\mathbf{A}$,$\mathbf{B}$ and $\mathbf{E}$ represent the mapping matrix of query, key and value, and $\mathbf{a}$,$\mathbf{b}$ and $\mathbf{e}$ are the corresponding biases. δ denotes the Relu activation function.

Subsequently, the output corresponding to each head is obtained using its respective attention scores. Finally, these outputs are concatenated and passed through a structured linear transformation to produce the final output.

$$\tilde{\mathbf{X}}^i = attn^i \mathbf{V}^i \tag{5}$$

$$\tilde{\mathbf{X}} = \delta(\mathbf{F} Concat([\tilde{\mathbf{X}}^1, \tilde{\mathbf{X}}^2, ..., \tilde{\mathbf{X}}^H]) + \mathbf{f}) \qquad s.t \quad \mathbf{F}_{ij} = 0, \quad if \quad i > j \tag{6}$$

$\mathbf{F}$ represents the weight matrix in the feed-forward layer, and $\mathbf{f}$ is the corresponding bias. It is worth emphasizing that SCFormer captures temporal features through structured linear transformations and activation functions, while the self-attention mechanism is used to compute correlation features between channels. SCFormer is constructed by stacking multiple layers of channel-wise self-attention mechanisms with structured linear transformation.

2.3 Convolutional Self-attention

Another approach to enforcing temporal constraints in the self-attention mechanism is through the use of 1D convolutions. Replacing all linear operations in the Transformer with 1D convolutions ensures that the self-attention mechanism

inherits the temporal properties of the convolution. In fact, multi-layer 1D convolutions are mathematically equivalent to a linear transformation implemented using a triangular matrix, offering a more structured approach with shared parameters. For an input series $\mathbf{z} \in \mathcal{R}^d$, a convolution with a kernel size of k and stride 1 can be represented as a linear transformation based on a structured matrix $\mathbf{K}$, assuming zero bias for simplicity.

$$\mathbf{K} = \begin{bmatrix} w_1 & w_2 & \cdots & w_k & \cdots & 0 & 0 & 0 \\ 0 & w_1 & \cdots & w_k & \cdots & 0 & 0 & 0 \\ 0 & 0 & w_1 & w_2 & \cdots & w_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & w_1 \end{bmatrix} \tag{7}$$

$$K * \mathbf{z} = \mathbf{K}\mathbf{z} \tag{8}$$

The matrix $\mathbf{K}$ is essentially a Toeplitz matrix. Here, $*$ denotes the convolution operation, and w_i represents the i-th weight in the convolution kernel K. For multi-layer convolutions, let $\mathbf{K}_i$ be the matrix for each layer. Then, the multi-layer convolutions can be represented as the multiplication of matrices:

$$\mathcal{F}(\mathbf{z}, k) = (\prod_i \mathbf{K}_i)\mathbf{z} \tag{9}$$

Using mathematical induction, it can be shown that the structured form of the matrix $\mathbf{K}_i$ allows the generation of a complete upper triangular matrix with at most $\lceil \frac{d-k}{k-1} \rceil + 1$ layers of convolution. This demonstrates that multi-layer 1D convolutions can be implemented as a linear transformation based on an upper triangular matrix with shared weights. The entire convolutional self-attention mechanism can be formalized as follows.

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \delta(Conv_Q(\mathbf{Z})), \delta(Conv_K(\mathbf{Z})), \delta(Conv_V(\mathbf{Z})) \tag{10}$$

$$attn^i = softmax(\frac{\mathbf{Q}^i(\mathbf{K}^i)^T}{\sqrt{d/H}}) \tag{11}$$

$$\tilde{\mathbf{X}}^i = attn^i \mathbf{V}^i \tag{12}$$

$$\tilde{\mathbf{X}} = \delta(Conv_F(Concate([\tilde{\mathbf{X}}^1, \tilde{\mathbf{X}}^2, ..., \tilde{\mathbf{X}}^H]))) \tag{13}$$

Most of the mathematical symbols are defined in Section 2.2 and will not be repeated here.

Models	ETT	PEMS	Solar	ECL	Exchange	Weather	Traffic
SCFormer *conv*	**0.373**	0.104	0.232	0.161	0.315	0.238	0.445
SCFormer *triangular*	0.375	**0.084**	**0.227**	**0.156**	**0.299**	**0.235**	0.509
iTransformer	0.383	0.105	0.233	0.178	0.360	0.258	**0.428**
RLinear	0.380	0.515	0.369	0.219	0.378	0.272	0.626
PatchTST	0.381	0.202	0.270	0.216	0.367	0.259	0.555
Crossformer	0.685	0.222	0.641	0.244	0.940	0.259	0.550
TiDE	0.482	0.366	0.347	0.251	0.370	0.271	0.760
TimesNet	0.391	0.126	0.301	0.192	0.416	0.259	0.620
DLinear	0.442	0.311	0.330	0.212	0.354	0.265	0.625
SCINet	0.689	0.105	0.282	0.268	0.750	0.292	0.804
FEDformer	0.408	0.198	0.291	0.214	0.519	0.309	0.610

Table 1: The main experimental results on MSE. Optimal results are highlighted in bold, and suboptimal results are underlined.

3 Experiments

3.1 Datasets

Datasets. We evaluate our method on several widely used datasets. ETT(subsets: ETTh1, ETTh2, ETTm1, ETTm2), we report average performance on them; Electricity (ECL)[1]; Traffic[2]; Weather[3]; Exchange; The PEMS (subsets PEMS04 and PEMS07) [5]; Solar-Energy. Most datasets are split in a 7:1:2 ratio for training, validation, and testing.

3.2 Baselines

9 popular methods are used as baselines, including four Transformer-based methods: (1) iTransformer [7], (2) FEDformer [12], (3) PatchTST [8], and (4) Crossformer [11]; three MLP-based methods: (5) TiDE [2], (6) RLinear [4], and (7) DLinear [10]; and two CNN-based methods: (8) TimesNet [9] and (9) SCINet [5]. The baseline results are all taken from their official implements. All methods, including ours, use a fixed look-back size of 96 and predict time horizons of 96, 192, 336, and 720. For space limitations, unless otherwise specified, we only report their average performance under the MSE metric.

3.3 Experimental Results

Table 1 compares the proposed SCFormer-*conv* and SCFormer-*triangular* with baseline methods. SCFormer-*conv* replaces all linear transformations with 1D

[1] `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`
[2] `http://pems.dot.ca.gov`
[3] `https://www.bgc-jena.mpg.de/wetter/`

convolutions, while SCFormer-*triangular* employs structured triangular matrices for linear mappings. Both approaches significantly improve forecasting performance, with SCFormer-*triangular* achieving superior results. For example, SCFormer-*triangular* achieves an average MSE improvement of 12.3% over the channel-wise state-of-the-art model iTransformer [7] on the ECL dataset, 16.9% on the Exchange dataset, and 8.9% on the Weather dataset. For SCFormer-*conv*, it achieves an average MSE improvement of 2.6% on the ETT dataset and 7.7% on the Weather dataset. Considering the parameter size of SCFormer-*conv*, this performance is quite competitive. The results demonstrate that most datasets benefit from our method, emphasizing its general effectiveness.

Models	SCFormer		*/wo-HiPPO*	
Dateset	MSE	MAE	MSE	MAE
ECL	**0.156**	**0.254**	0.176	0.266
Weather	**0.235**	**0.271**	0.259	0.282
Solar	**0.227**	**0.261**	0.235	0.264

Table 2: The ablation experimental.

Models	SCFormer		*/wo-look-back*	
Dateset	MSE	MAE	MSE	MAE
ECL	**0.156**	**0.254**	0.167	0.275
Traffic	**0.509**	**0.359**	0.756	0.471
Solar	**0.227**	**0.261**	0.241	0.288

Table 3: The ablation experimental.

3.4 Ablation Study

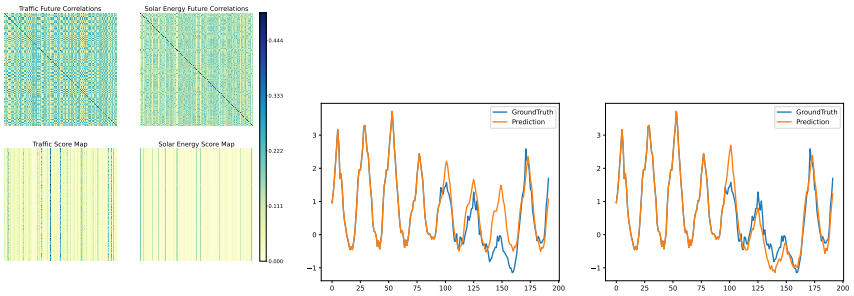
To explore the role of temporal constraints, we compare SCFormer with Transformer-HiPPO, which removes the temporal constraint but retains the HiPPO embedding. Due to space limitations, we only report results with a prediction length of 720. As shown in Table 4, SCFormer achieves better forecasting performance in most circumstances. This suggests that temporal constraints help mitigate overfitting, leading to lower prediction error. The cumulative historical state maintains the long-term state of a historical series by projecting it into an orthogonal polynomial space using HiPPO. To evaluate its impact on forecasting performance, we remove the HiPPO embedding from the model. As shown in Table 2, the model's performance significantly declines, demonstrating the effect of the cumulative historical state. To assess the necessity of the look-back window, we remove it and use only the cumulative historical state generated by HiPPO for forecasting. As shown in Table 3, the model's performance significantly drops without the look-back window. This is expected, as HiPPO represents the overall state of the time series, not direct information in the real number domain. This confirms that the cumulative historical state and look-back window provide complementary features.

3.5 Case Study

To intuitively illustrate the advantages of our method, we compare it with iTransformer using an example from the *ECL* dataset. Figure 3(b-c) shows that iTransformer's prediction curve is significantly distorted around timestamp 150, while our method provides more accurate predictions. We also plot the attention scores

Models	ETTm1	ETTm2	ETTh1	ETTh2
SCFormer *conv*	**0.460**	**0.408**	**0.483**	**0.426**
SCFormer *triangular*	0.471	0.454	0.489	0.427
Transformer *HiPPO*	0.468	0.445	0.494	0.436

Table 4: The ablation experimental results on MSE for temporal constraints using structured matrices on the *ETT* dataset.



(a) (b) (c)

Fig. 3: (a) Channels(multivariate) correlations: Left-Top: the future correlations of *Traffic*; Left-Bottom: the attention scores of *Traffic*; Right-Top: the future correlations of *Solar-Energy*; Right-Bottom: the attention scores of *Solar-Energy*. (b) Example visualization of iTransformer on ECL. (c) Example visualization of SCFormer-*triangular* on ECL.

in the model's last layer and the future correlations for the *Traffic* and *Solar-Energy* datasets in Figure 3(a). The results show that the model is able to clearly learn the correlations between channels within the prediction horizon, for example, brighter columns in future correlations correspond to darker areas in the attention scores. However, compared to the *Solar* dataset, the patterns in the *Traffic* dataset are less pronounced, which indirectly explains why the model performs less optimally on the *Traffic* dataset.

4 Conclusion

In this paper, we propose SCFormer, a multivariate time series forecasting model. SCFormer uses 1D convolutions and triangular matrices to structure the linear transformations in the channel-wise Transformer, thereby introducing temporal constraints. Additionally, we introduce a method for maintaining the cumulative historical state based on HiPPO, which serves as a simple and efficient memory mechanism, allowing the model to capture historical information beyond the

fixed look-back window. Extensive comparative experiments, ablation studies, and analytical evaluations confirm the effectiveness of the proposed method.

5 Acknowledgments.

This paper is supported by the following funds: "Tianshan Talents" Project (No.2023TSYCTD0011,No.2023TSYCLJ0022)Natural Science Foundation of Xinjiang Uygur Autonomous Region(No.2022D01E93)The Youth Innovation Promotion Association of Chinese Academy of Sciences (Grant No.2021434)

References

1. Chen, Z., Ma, M., Li, T., Wang, H., Li, C.: Long sequence time-series forecasting with deep learning: A survey. Information Fusion **97**, 101819 (2023)
2. Das, A., Kong, W., Leach, A., Mathur, S.K., Sen, R., Yu, R.: Long-term forecasting with tide: Time-series dense encoder. Transactions on Machine Learning Research (2023)
3. Gu, A., Dao, T., Ermon, S., Rudra, A., Ré, C.: Hippo: Recurrent memory with optimal polynomial projections. Advances in neural information processing systems **33**, 1474–1487 (2020)
4. Li, Z., Qi, S., Li, Y., Xu, Z.: Revisiting long-term time series forecasting: An investigation on linear mapping. arXiv preprint arXiv:2305.10721 (2023)
5. Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., Xu, Q.: Scinet: Time series modeling and forecasting with sample convolution and interaction. Advances in Neural Information Processing Systems **35**, 5816–5828 (2022)
6. Liu, Y., Wang, Z., Yu, X., Chen, X., Sun, M.: Memory-based transformer with shorter window and longer horizon for multivariate time series forecasting. Pattern Recognition Letters **160**, 26–33 (2022)
7. Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: itransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:2310.06625 (2024)
8. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: The Eleventh International Conference on Learning Representations (2022)
9. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: Timesnet: Temporal 2d-variation modeling for general time series analysis. In: The eleventh international conference on learning representations (2022)
10. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 11121–11128 (2023)
11. Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: The eleventh international conference on learning representations (2022)
12. 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)