Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 17, No. 3, 2025

Decomposition and CNN based time series forecasting methods

Bo He [a], Yunya Bo[*], Deliang Zhang [b]

School of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China

[*] **Corresponding author**: Yunya Bo (Email: yunyay@qq.com), [a] hebo@cqut.edu.cn, [b] 814705867@qq.com

Abstract: Time series forecasting is usually based on historical observations and is used to predict trends and values of data for a certain period of time in the future. The usefulness of time series forecasting lies in its ability to reveal the intrinsic patterns in the data, such as trend, seasonality and periodicity, which are of great significance to people's decision making, planning and risk management. In this paper, the current research status of time series forecasting is sorted out, and the feature extraction methods involved in time series forecasting as well as the related theories such as deep learning are deeply analyzed, and the advantages and disadvantages of each method are compared, and on the basis of which, the future development trend of time series forecasting methods is outlooked.

Keywords: Time Series Forecasting; Modal Decomposition; Convolutional Neural Networks.

1. Introduction

Forecasting of time series plays a crucial role in both science and engineering, including financial stocks [1], traffic flow [2], disease transmission [3], weather changes [4], and so on. Recently, with the increasing demand for long-term time series forecasting, long-term time series forecasting has also been widely used in various fields. For example, more accurate traffic flow prediction can help our traffic management department to better monitor and control the traffic condition of the road, reduce congestion and improve the efficiency of road use [5]. Accurate weather forecasts also play an important role in people's planning and activity scheduling in life, which can help people avoid the inconvenience and danger caused by bad weather [6]. With the help of accurate inventory forecasts, investors can develop smarter strategies to cope with changing market conditions and reduce the risk of inventory backlogs and stock-outs [7][8]. Accurate electricity forecasting is the lifeline that enables the smooth functioning of electricity markets and energy trading, helping to balance supply and demand, reduce energy costs, and support the integration of renewable energy sources [9] [10].

In early studies, most scholars at home and abroad generally relied on traditional statistical methods to forecast time series data. Wang Y. et al [11] employed the differential integrated moving average autoregressive model (ARIMA) to forecast time series, while Faghih S. [12] used the autoregressive sliding average model (ARMA) and linear regression to model and calculate the demand for cabs. Although these traditional statistics-based methods show their unique advantages in dealing with linear relationships, their data requirements are also an obvious limitation of the method. Subsequently, the emergence of machine learning methods has been able to improve the shortcomings of traditional statistical methods, as Feng, Zhongkai, et al [13] utilized artificial neural networks (ANNs) to achieve the prediction of river flow safety, and Singh, et al [14] utilized the SVM model and the repetitive wavelet transform to predict the electric load. However, since machine learning usually treats data as independent of each other, it is unable to capture the dependencies between data, as well as their seasonality and periodicity when predicting complex time series, resulting in no significant improvement in effectiveness. After that, these deep learning methods, which are widely used in natural language processing, have been introduced into time series research because natural language processing is still characterized by temporal sequences in nature, and deep learning has made significant achievements in natural language processing tasks. Time series prediction methods based on deep learning have thus developed rapidly. In predicting multivariate long time series, our goal is to improve the accuracy of the prediction results and to extend the time horizon of the prediction as much as possible. This goal is achieved by constructing algorithms that can more efficiently capture the long-term dependent features in the time series, as well as the interrelated information in the data. Therefore, the use of deep learning techniques to construct time series prediction models with both high explanatory power and high accuracy is not only important in theoretical research, but also of great practical value in practical decision-making in various industries.

2. Methods based on time series decomposition

Decomposing time series is a common method for analyzing complex time series data, which usually splits the time series into three main components: the trend component, the seasonal component and the stochastic component. The trend component reveals the long-term evolution direction of the time series, the seasonal component reflects the cyclical fluctuations of the time series, and the stochastic component represents the irregular fluctuations or noise in the series. Decomposition methods are able to decompose a time series into a number of subseries of different sizes. These subseries tend to show more obvious smoothness and regularity compared to the original series, thus facilitating deep learning models to mine potential features as a way to improve the accuracy of time series prediction. Because the decomposed time series approach views the original series as the sum of different subseries of the decomposition by performing certain operations, the decomposed time series can be viewed as the sum of the different subseries of the original series.

- Secular Trend (T): Reflects the general direction or evolution of a time series over a long time span.
- Seasonal Variation (S): Reflects the cyclical pattern of changes in the time series due to the change of seasons.
- Cyclical Variation (C): non-strictly regular cyclical variations that occur in cycles of several years (or longer).
- Irregular Variation (I): short-term, irregular fluctuations in time series caused by various random or accidental factors.

In time series analysis, depending on different application scenarios and data characteristics, time series are usually decomposed into different components for better understanding and modeling. This decomposition is mainly based on two principles: the additive principle and the multiplicative principle. Among them, the additive model is shown in equation (1):

$$X_t = T_t + S_t + C_t + I_t \tag{1}$$

where X_t denotes the observed value of the time series at time point t, T_t denotes the long-term trend component, S_t denotes the seasonal component, C_t denotes the cyclical fluctuation component, and I_t denotes the irregular fluctuation component.

In the additive model, the four components are independent of each other, i.e., changes in one component do not affect the others. Each component is expressed as an absolute quantity and their magnitudes are consistent with those of the original time series. This model is suitable for situations where there is no interaction between the components of the time series, e.g., seasonal variations and the magnitude of the trend component do not affect each other.

The multiplication model is shown in equation (2):

$$X_t = T_t \times S_t \times C_t \times I_t \tag{2}$$

Unlike the additive model, the four components in the multiplicative model are interdependent. In this model, the long-term trend T_t is usually expressed as an absolute quantity with the same magnitude as the time series itself, while the seasonal component S_t, the cyclical volatility component C_t, and the irregular volatility component I_t are expressed as relative quantities (usually proportions or ratios). This type of modeling is appropriate for situations where there is seasonal variation in the time series or where there are interactions between the other components and the trend component, e.g., where the magnitude of the seasonal variation increases as the trend increases.

Time series decomposition methods are important in practical applications, which not only provide a clear framework for time series analysis, but also lay the foundation for subsequent modeling and forecasting. A typical example is Prophet, an open source time series forecasting model publicly provided by Facebook [15]. The Prophet model decomposes a time series into a trend term g(t), a seasonal term s(t), a holiday term h(t) and a noise term n(t). These components are combined together by summation to form a complete forecasting model. Prophet model is able to effectively adapt to a variety of complex time series data by flexibly handling trend, seasonality and holiday effects.

The core value of the time series decomposition method is that it provides a systematic way of thinking about analysis. By decomposing the time series into different components, various patterns and features in the data can be more clearly identified and understood. This idea is not only applicable to traditional statistical analysis methods, but also provides an important reference for the design of machine learning and deep learning models. Many modern time series forecasting models, such as those based on neural networks, also draw on this decomposition idea to decompose the time series modeling problem into modeling of different components, thus improving the accuracy and interpretability of the forecast. In conclusion, the time series decomposition method is not only an effective analytical tool, but also provides an important theoretical basis for more complex time series modeling and forecasting.

3. Time series forecasting methods

With the remarkable progress of deep learning in the fields of computer vision and natural language processing, its methods are gradually being introduced to the field of time series forecasting. Deep neural networks are able to characterize high-dimensional data more efficiently by building diverse network architectures, thus reducing the reliance on traditional manual feature engineering and complex model design. In addition, by defining appropriate loss functions, deep learning models can be trained in an end-to-end manner, which further simplifies the modeling process and improves model adaptability.

3.1. Method based on recurrent neural network

In order to solve the gradient vanishing and gradient explosion problems in RNN and to model complex time series, some new research works have been proposed. In order to solve the gradient vanishing and gradient explosion problems in RNNs and to model complex time series, several new research works have been proposed. These research works include the use of structures such as Long Short-Term Memory Networks [16] (LSTMs) and Gated Recurrent Units (GRUs) to enhance the ability of RNNs to better handle long sequence data and avoid the gradient problem, and the GRU-based prediction method is shown in Fig. 1. In addition, other researchers have explored RNN models that incorporate the attention mechanism, Zhu et al [17] proposed an LSTM encoder-decoder framework that incorporates the attention mechanism for the prediction of large-scale operational workloads in data centers. This approach significantly improves the accuracy of workload prediction by accurately capturing dependencies in time-series data. Song et al [18] used an adaptive LSTM model to explore the correlation between historical data points and the distance from the current time window. They divided the dataset into two parts, immediate and cumulative information, and subsequently used optimized LSTM units to replace the hidden layer in traditional RNNs as a way to capture the medium and long-term dependency features in the time series more efficiently. Memory units in the GRUs help the network capture important information in the sequences by remembering the current state and updating it to a new one, which further strengthens the handling of long dependency Capability.

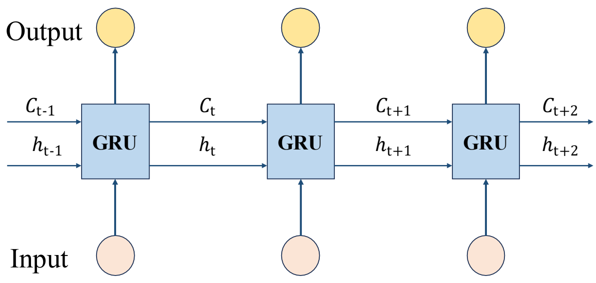

Figure 1. Schematic of the time series forecasting principle of GRU

The input sequence of $X = [x_1, x_2, \dots, x_t]$, x_t is the input of the tth time step. the extraction process of the long-term dependency of GRU is as follows:

The reset gate r_t determines how much the hidden state h_{t-1} of the previous time step influences the current candidate hidden state $\widetilde{h_t}$. As in Eq. (3), σ is the sigmod function and W_r is the trainable weight matrix.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{3}$$

The effect of the hidden state h_{t-1} of the previous time step is controlled using the reset gate r_t. The formula for the former candidate hidden state $\widetilde{\mathrm{h}_t}$ is given in (4).

$$\widetilde{h_t} = tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h) \tag{4}$$

where $\odot$ denotes elementwise multiplication, $tanh$ is the hyperbolic tangent function, and W is the trainable weight matrix.

The update gate z_t determines how much of the previous time step's hidden state h_{t-1} and how much of the current candidate hidden state $\widetilde{h_t}$ should be retained for the current hidden state h_t. Equation (5) yields the update gate z_t.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{5}$$

where W_z is the trainable weight matrix.

Use the update gate z_t to combine the hidden state h_{t-1} of the previous time step and the current candidate hidden state h_t as in (6).

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t} \tag{6}$$

3.2. Method based on Convolutional Neural Network

CNN is a deep machine learning model under supervised learning, which is extremely adaptive and good at extracting local features of data, and based on this, global feature extraction and classification. Essentially, CNN is a multilayer perceptual machine, and the key to its success lies in its unique mechanism of local connectivity and shared weights. This mechanism reduces the number of weights on the one hand, making the network easier to optimize; on the other hand, it reduces the risk of overfitting. CNN is one kind of neural network, and its weight-sharing network structure not only reduces the complexity of the network model, but also further reduces the number of weights.

The basic structure of CNN mainly consists of two special neuron layers: convolutional layer and pooling layer. In the convolutional layer, each neuron is only connected to the local region of the previous layer, so as to extract the features of that local region; while the pooling layer is used for local sensitivity analysis and secondary feature extraction to further reduce the dimensionality of the features. One-dimensional Convolutional Neural Network (1D-CNN) is a variant of CNN, which is usually used to process data with sequence dependency, such as time series and text.

Time series are characterized by their sequentially and temporal correlation, and CNN extracts the correlation features between local time steps by 1D Convolution. The specific steps are as follows:

(1) Input Layer: The input is the history window of the time series. Assume that the input time series length is *T*, and the data dimension of each time step is *d* (d=1 for univariate time series)

(2) Convolution layer: A one-dimensional convolution kernel is applied on the time axis, and the sliding window extracts the same part of the features. The convolution operation is formulated as in (7):

$$y_i = \sum_{k=1}^{K} w_k \cdot x_{i+k-1} \tag{7}$$

where x_i is the input sequence, y_i is the output sequence, w_k is the parameter of the convolution kernel, and k is the size of the convolution kernel.

Activation layer: A nonlinear activation function (e.g., ReLU) is usually used to introduce nonlinear capability to the model. The formula for the activation function of the modified layer is shown in (8).

$$f(x) = \max(0, x) \tag{8}$$

(3) Pooling layer: The pooling layer is used for down sampling, reducing data dimensions and avoiding overfitti ng, commonly used pooling methods are maximum poolin g and average pooling.

(4) Fully connected layer: The pooled features are map ped to the predicted values through the fully connected la yer.

(5) Output Layer: Predict the value of one or more fut ure time steps.

$$output = activation(W \cdot input + b) \tag{9}$$

Equation (9) shows the calculation process of the output layer. Where, W is the weight matrix; $input$ is the input feature vector; b is the bias vector, $activation$ is the activation function.

ully Convolutional Network (FCN, Fully Convolutional Network) FCN consists entirely of convolutional layers CNN effectively extracts localized features and patterns in time series by applying a series of convolution and pooling operations on the time series data. This is useful for understanding complex patterns and trends in the data. Convolutional layers can capture local dependencies and temporal dynamics in time series data, and different convolutional kernels can learn different aspects of the data, such as trends, periodicities, and anomalies. The core idea of the skip-level structure in FCN is to enhance the feature transfer efficiency by introducing Skip Connection, which passes the features from different levels of the network directly to the subsequent layers, thus reducing the information loss. This structure is widely used in deep convolutional networks, especially when dealing with complex semantic segmentation tasks. In time series prediction, the level-skipping structure can help the model to better capture long-term dependencies and multi-scale features. This process for FCNs is shown in Figure 2.The predictions from the underlying raw data (FCN-32s) are up-sampled by a factor of 2 to obtain the original size and fused (summed) with predictions from the pool4 layer (stratum 16).This portion of the network is referred to as FCN This part of the network is referred to as FCN-16s. The predictions from this part of the network are then up-sampled again by a factor of 2 and fused with the predictions from the pool3 layer, and this part of the network is referred to as FCN-8s.

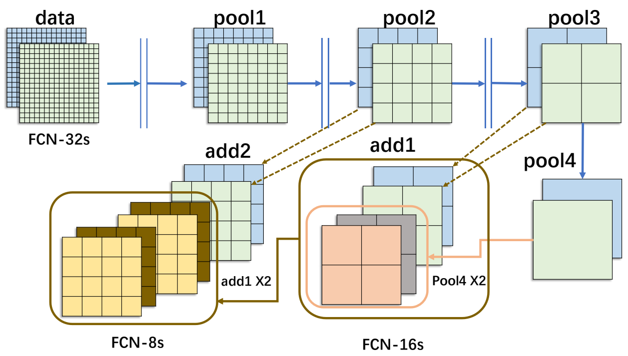

Figure 2. Structure of FCN

3.3. Method based on attention mechanism

Transformer-based models, relying on its self-attention mechanism, have the advantage of excelling in dealing with long-term dependencies in time series and maintaining the relative completeness of information in a wider range of series. As a result, Transformer has become a mainstream technique for time series (TS) forecasting, and many variant models of Transformer have been proposed for time series forecasting. For example, FED former [19] replaced the original self-attention and cross-attention mechanisms by introducing two modules, Fourier-enhanced and wavelet-enhanced, into the encoder and decoder, which effectively enhanced the model's ability to extract information on different time scales, while Pyraformer [20] built transient models on different scales. However, these models suffer from high complexity and difficulty in capturing temporal dynamics, and the computational nature of the self-attention mechanism introduces the challenge of secondary time complexity. So Transformer and its variant models are limited in long-term forecasting tasks. MLP has also been introduced in the field of time series forecasting. Recent studies have found that the underlying linear models can also show excellent prediction results in some specific scenarios. For example, LTSF-Linear [21] introduces a simple one-layer linear model that outperforms Transformer-based models in many cases. DLinear uses only two single linear layers of the network for forecasting, which reduces the complexity but it lacks sufficient model complexity to learn these complex time series features and is overly simplified, it is prone to overfitting. The N-BEATS framework [22], which is based on backward and forward residual links and very deep fully-connected layers, provides a high degree of interpretability. Informer [23] filters out dominant queries based on the similarity between the query (Query) and the key (Key) with a computational complexity of $O(Log\ L)$, thus achieving an optimization similar to LogTrans in terms of efficiency. In addition, Informer employs a generative decoder to directly generate long-term prediction sequences, which effectively avoids the errors accumulated in long-term prediction due to single-step forward prediction. Meanwhile, researchers have explored frequency-domain auto-attention mechanisms in the field of time series modeling. Autoformer [24] proposes a short-term trend decomposition architecture in which the autocorrelation mechanism is used as the attention module. Unlike traditional attention mechanisms, this mechanism generates an output with $O(L\ Log\ L)$ complexity by measuring the time-delay similarity between input signals and aggregating the top k most similar subsequences.

The key to the attention mechanism is to give the model the flexibility to focus on the portion of the sequence that has the highest relevance to the task at hand when processing the sequence data. Specifically, before the input sequence is fed into the model, the attention mechanism calculates the input elements and then assigns a weight to each element, and then performs a weighted summation calculation based on these weights, which finally generates a weighted context vector. The above processing is done by three attributes *Query*, *Key* and *Value* in the attention mechanism. As shown in Figure 3:

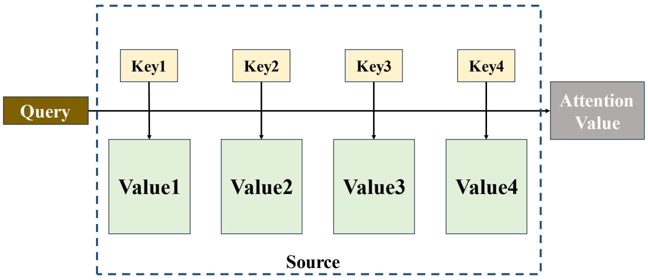

Figure 3. Schematic diagram of the principle of attention mechanism

Query: represents the information that the current element wishes to focus on. Key: represents the information that other elements can provide to the current element, which is related to the Query's requirements. Value: the information that the Query gets after matching the Query with the Key.

4. Conclusion

Although decomposition-based feature extraction methods and deep learning-based time series prediction methods have made significant progress in the field of time series, the complexity of future prediction application scenarios puts forward higher requirements, and points out the direction for further research on deep learning-based time series prediction techniques. However, there are still some unsolved problems in the current time series research, for example, most of the current research focuses on feature extraction and feature combination to meet different task objectives, but there is still a lack of effective solutions for causal inference of time series, and the interpretability of the prediction results needs to be further improved; in addition, in the actual prediction process, there are often anomalies in the data, and these anomalies will have an interference with the prediction results. Therefore, how to effectively identify and ignore these anomalies in the prediction process to reduce the prediction error and improve the robustness of the model is one of the important problems to be solved; with the explosive growth of the time series data scale, the real-time data processing faces serious challenges. The time complexity of the existing algorithms is too high to meet the real-time requirements, which to some extent restricts the widespread popularization of time series prediction techniques in practical applications. To address the above problems, future research can focus on strengthening the exploration of the following aspects: first, to improve the interpretability of time series forecasting methods, through the introduction of causal inference and other methods, so that the prediction results are more persuasive; second, to enhance the robustness of the model, through the improvement of the algorithm or the introduction of the anomaly detection mechanism, to improve the model's tolerance of abnormal data; third, to optimize the time complexity of the algorithm to enhance the data processing and meet the demand for analyzing large-scale time-series data. Through in-depth research in these directions, time series prediction and feature extraction methods based on deep learning can be made more secure, reliable, efficient and flexible, so as to better cope with the challenges of practical problems.

Acknowledgements

This work is funded by the humanities and social sciences research project of the ministry of education (No.24YJA870003) and the Chongqing University of Technology Graduate Education High Quality Development Project (No. gzlcx20253209)

References

[1] Patton A. Copula methods for forecasting multivariate time series[J]. Handbook of economic forecasting, 2013, 2: 899-960.

[2] Zhang J, Zheng Y, Qi D. Deep spatio-temporal residual networks for citywide crowd flows prediction[C]//Proceedings of the AAAI conference on artificial intelligence. 2017, 31(1).

[3] Matsubara Y, Sakurai Y, Van Panhuis W G, et al. FUNNEL: automatic mining of spatially coevolving epidemics[C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014: 105-114.

[4] Angryk R A, Martens P C, Aydin B, et al. Multivariate time series dataset for space weather data analytics[J]. Scientific data, 2020, 7(1): 227.

[5] Qin H, Zhan X, Li Y, et al. Network-wide traffic states imputation using self-interested coalitional learning[C]//Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021: 1370-1378.

[6] Zhao Y, Norouzi H, Azarderakhsh M, et al. Global patterns of hottest, coldest, and extreme diurnal variability on earth[J]. Bulletin of the American Meteorological Society, 2021, 102(9): E1672-E1681.

[7] Liu T, Ma X, Li S, et al. A stock price prediction method based on meta-learning and variational mode decomposition[J]. Knowledge-Based Systems, 2022, 252: 109324.

[8] Ma X, Zhao T, Guo Q, et al. Fuzzy hypergraph network for recommending top-K profitable stocks[J]. Information Sciences, 2022, 613: 239-255.

[9] Ma X, Li X, Fang L, et al. U-Mixer: An Unet-Mixer Architecture with Stationarity Correction for Time Series Forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2024, 38(13): 14255-14262.

[10] 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.

[11] Y.W. Wang, Shucai Ma. Time series forecasting based on hybrid ARIMA and LSTM models[J]. Computer Applications and Software, 2021, 38(2): 291-298.

[12] Faghih S, Shah A, Wang Z, et al. Taxi and mobility: Modeling taxi demand using ARMA and linear regression[J]. Procedia Computer Science, 2020, 177: 186-195.

[13] Feng Z, Niu W. Hybrid artificial neural network and cooperation search algorithm for nonlinear river flow time series forecasting in humid and semi-humid regions[J]. Knowledge-Based Systems, 2021, 211: 106580.

[14] Sasal L, Chakraborty T, Hadid A. W-transformers: a wavelet-based transformer framework for univariate time series forecasting[C]//2022 21st IEEE international conference on machine learning and applications (ICMLA). IEEE, 2022: 671-676.

[15] Taylor S J, Letham B. Forecasting at scale[J]. The American Statistician, 2018, 72(1): 37-45.

[16] Liu M, Ren S, Ma S, et al. Gated Transformer Networks for Multivariate Time Series Classification[J]. arXiv preprint arXiv: 2103.14438, 2021.

[17] Y. Zhu, W. Zhang, Y. Chen, et al.A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment[J] EURASIP J. Wirel. Commun. Netw., 2019 (2019), pp. 1-18

[18] Tanaka G, Matsumori T, Yoshida H, et al. Reservoir computing with diverse timescales for prediction of multiscale dynamics[J]. Physical Review Research, 2022, 4(3): L032014.

[19] 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.

[20] Liu S, Yu H, Liao C, et al. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting[C]//# PLACEHOLDER_PARENT_METADATA_VALUE#. 2022.

[21] Zeng A, Chen M, Zhang L, et al. Are transformers effective for time series forecasting? [C]//Proceedings of the AAAI conference on artificial intelligence. 2023, 37(9): 11121-11128.

[22] Oreshkin B N, Carpov D, Chapados N, et al. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting[J]. arXiv preprint arXiv:1905.10437, 2019.

[23] Taylor S J, Letham B. Forecasting at scale[J]. The American Statistician, 2018, 72(1): 37-45.

[24] 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.