Mathematical Biosciences and Engineering

MBE, 21(2): 3391–3421.
DOI: 10.3934/mbe.2024150
Received: 18 December 2023
Revised: 15 January 2024
Accepted: 16 January 2024
Published: 04 February 2024

http://www.aimspress.com/journal/MBE

Research article

Ultra-short-term forecasting model of power load based on fusion of power spectral density and Morlet wavelet

Lihe Liang[1], Jinying Cui[1], Juanjuan Zhao[2,3,*], Yan Qiang[1] and Qianqian Yang[3]

[1] College of Computer Science and Technology (College of Data Science), Taiyuan University of Technology, Taiyuan 030600, China
[2] School of Software, Taiyuan University of Technology, Taiyuan 030600, China
[3] College of Information, Jinzhong College of Information, Jinzhong 030800, China

* **Correspondence:** Email zhaojuanjuan@tyut.edu.cn; Tel: +8618636667123.

Abstract: An accurate ultra-short-term time series prediction of a power load is an important guarantee for power dispatching and the safe operation of power systems. Problems of the current ultra-short-term time series prediction algorithms include low prediction accuracy, difficulty capturing the local mutation features, poor stability, and others. From the perspective of series decomposition, a multi-scale sequence decomposition model (TFDNet) based on power spectral density and the Morlet wavelet transform is proposed that combines the multidimensional correlation feature fusion strategy in the time and frequency domains. By introducing the time-frequency energy selection module, the "prior knowledge" guidance module, and the sequence denoising decomposition module, the model not only effectively delineates the global trend and local seasonal features, completes the in-depth information mining of the smooth trend and fluctuating seasonal features, but more importantly, realizes the accurate capture of the local mutation seasonal features. Finally, on the premise of improving the forecasting accuracy, single-point load forecasting and quantile probabilistic load forecasting for ultra-short-term load forecasting are realized. Through the experiments conducted on three public datasets and one private dataset, the TFDNet model reduces the mean square error (MSE) and mean absolute error (MAE) by 19.80 and 11.20% on average, respectively, as compared with the benchmark method. These results indicate the potential applications of the TFDNet model.

Keywords: ultra-short-term time series prediction; series decomposition; global trend features; local seasonal features; quantile probabilistic load forecasting

1. Introduction

Power load forecasting is a vital component of power system operation and management, holding significant importance for the stable operation and intelligent scheduling of the power system [1]. However, with the advancement of the energy internet and the continual improvement of people's living standards, there has been a substantial increase in the volume and volatility of power load data. This surge in data poses significant challenges to the management of power system operations. In response to these challenges, ultra-short-term time series forecasting has become increasingly crucial. Ultra-short-term time series forecasting under power load refers to high-temporal-resolution predictions of power load for either the upcoming few hours or a day [2]. Ultra-short-term time series forecasting offers more accurate and rapid forecasts, which can provide decision support for energy management, demand response, stable power system operation, and intelligent scheduling. Research in ultra-short-term time series forecasting for power load is of paramount importance for enhancing the efficiency and stability of the power system [3].

Currently, research methods for ultra-short-term time series forecasting can be broadly categorized into two main groups: statistical-based methods and data-driven artificial neural network methods. The autoregressive integrated moving average (ARIMA) model [4] is one of the most widely applied statistical methods. The ARIMA model cleverly combines autoregression, moving average, and differencing operations along with other statistical techniques to predict stationary time series. However, with the sharp increase in historical data, data-driven artificial neural network methods have shown superior predictive performances over statistical methods. Convolutional neural networks (CNN) [5] are effective at extracting local features from time series data, though they are less capable of capturing temporal relationships between sequences. Alternatively, they are suitable for a time series with strong periodicity. On the other hand, recurrent neural networks (RNN) [6] are efficient at handling sequential data. This model leverages' memory cells to facilitate continuous information transfer within the network, thereby capturing the temporal dependencies between sequential data. However, the vanishing gradient problem limits the application of RNNs in sequence forecasting, which is a common challenge faced by most RNN models. DeepAR [7] is an RNN model that combines autoregressive methods with long short-term memory (LSTM) [8]. The core of DeepAR is to use LSTM to learn the dynamic features of historical sequences to achieve the prediction of future points in time. The residential load forecasting - multimodal graph neural network (RLF-MGNN) [9] is a model based on the combination of graph convolutional neural networks (GCN) and LSTM. The core of the model is to use the GCN to extract the linear and nonlinear features of the synchronization and causal graphs; then, the LSTM is used to achieve the ultra-short-term prediction of the sequence. Such variants of the model can effectively alleviate problems, such as gradient disappearance, that exist in the RNN, but cannot solve the recursive dependence, which leads to the problem of model performance degradation.

Recently, the Transformer model has demonstrated remarkable performance in sequence data analysis [10]. By utilizing attention mechanisms, Transformer can capture long-range dependencies among sequence elements and enable one-step computation, thus addressing the recursive dependency challenge faced by previous RNN models. Consequently, numerous variants of the Transformer model have emerged. For instance, LogTrans [11] is a transformer model based on logarithmic transformations, which uses exponentially growing time intervals for attention calculation. Reformer [12] introduces a position-sensitive, hash-based attention mechanism as an alternative to traditional attention mechanisms. Informer [13] employs a KL divergence-based sparse attention mechanism and a distillation strategy, while producing one-step output predictions to avoid error accumulation in the model results. KL divergence refers to Kullback-Leibler divergence, a mathematical concept used to measure the difference between two probability distributions. However, the attention mechanisms in these models still operate at the point level and do not fully exploit the correlations between similar sequences. Additionally, there is room for improvement in terms of the interpretability of these models.

The sequence decomposition model is an encoder-decoder model with a transformer at its core, incorporating traditional time series decomposition. The idea of this model is to decompose a time series into components with evident regularity, such as trend and seasonality, followed by separate learning of these components and progressive aggregation. As shown in Figure 1, the autocorrelation function (ACF) and partial autocorrelation function (PACF) between different lagged time series values on the ETTm1, ETTm2, Weather, and PowerLoad datasets reveal a significant correlation between the current value of the sequence and its lagged values. Furthermore, Figure 2 demonstrates that the trend and seasonality components obtained through an additive model align with the patterns of the original sequence. Therefore, the idea of time series decomposition exhibits strong feasibility. For example, based on the stochastic process theory, Wu et al. proposed a sequence decomposition model, called Autoformer [14], with an autocorrelation attention mechanism instead of a point-level connection. The Autoformer model decomposes the sequence into trend and seasonal terms and utilizes the autocorrelation attention mechanism to discover cycle-based sequence-level dependencies, realize the sequence-level connection, and break the bottleneck of information utilization. Meanwhile, Autoformer enhances the model's interpretability from the perspective of sequence decomposition. However, sequence analysis from a time-domain perspective makes it difficult to capture a more complete pattern of cycles on the one hand and compounds the computational cost of the model on the other.

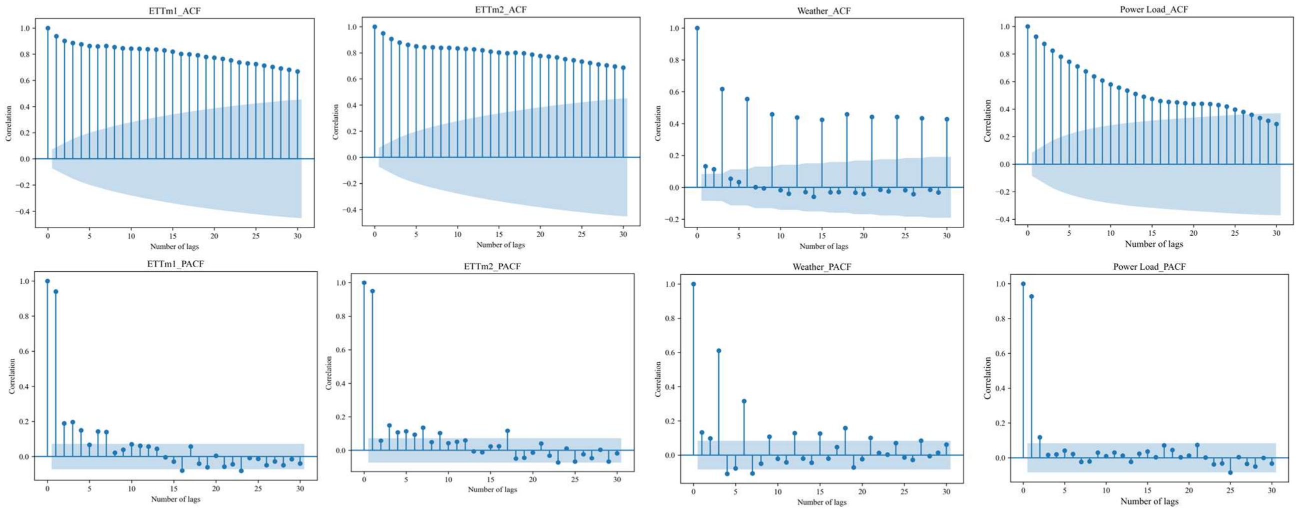

Figure 1. ACF and PACF plots.

Frequency-domain feature extraction compensates for the limitations of a time-domain analysis. Due to their sparsity and global nature, frequency domain features, can be obtained by transforming time-domain signals into frequency-domain signals using methods such as Fourier transforms and wavelet transforms. Further utilization of feature extraction modules, such as attention mechanisms and convolution mechanisms, can allow dependencies between sequences to be captured while reducing the model's computational cost. For example, FEDformer [15] utilizes the fast Fourier transform and frequency domain attention mechanism to learn the frequency domain correlation of seasonal terms, which reduces the time and space complexity while maintaining the prediction accuracy and thus saves the computational cost of the model. The TimesNet [16] model uses Fourier transform to identify multiple periodic subsequences, transforms 1D data into 2D data, and utilizes Inception convolution modules to analyze sequence transformations within and between periods. The empirical mode decomposition and dual-stage attention-based recurrent neural network (EEMD-DARNN) model [17] first utilizes the time-frequency analysis method (ensemble empirical mode decomposition) to decompose the historical sequence into a sequence of derived variables, then utilizes a two-stage attention mechanism for temporal and spatial feature selection, and finally employs an LSTM to achieve a multi-step prediction of the sequence. Due to the existence of more complex and volatile local periodic patterns in the ultra-short-term time series forecasts of power loads, the aforementioned models still have obvious shortcomings in capturing the local periodic patterns, which affects the accuracy and stability of the time series prediction.

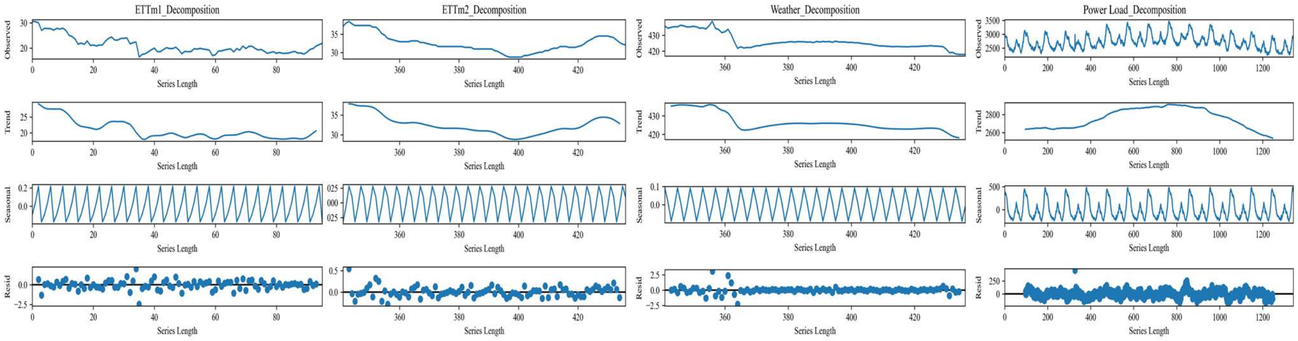

Figure 2. Series decomposition plots.

On this basis, this paper explores the energy perspective and finds that, as a frequency domain analysis method, power spectral density has unique advantages in feature extraction. The physical meaning of power spectral density is the value of signal energy per unit frequency, which is essentially expressed as the magnitude of power contained in different frequencies in the signal. Thomas et al. [18] achieved nonparametric and accurate periodicity detection by using the power spectral density as a new distance measure for extracting the most important periodic features in a sequence. Meanwhile, the Morlet continuous wavelet transform has a good time-frequency localization and multi-scale analysis capability in signal processing as compared to the Fourier transform. Therefore, a sequence decomposition model based on power spectral density and the Morlet continuous wavelet transform is proposed in this paper. The model incorporates both the advantages of power spectral density for accurate extraction of major periodic features and the wavelet transform for multiscale characterization of local periodic patterns; then, it utilizes a priori knowledge for attention-targeted steering of time series prediction, which ultimately realizes the progressive fusion prediction of trend and local seasonal terms. The main contributions of this paper are as follow:

1) An energy selection module based on power spectral density and the Morlet continuous wavelet

transform is proposed. Compared with other single-domain sequence decomposition models based on either a time or frequency domain, this module can fully exploit the trend and seasonal terms within the sequence from a multidimensional perspective; meanwhile, the feature selection method based on the power spectral density and Morlet continuous wavelet transform can effectively make up for the high temporal and spatial complexity problems of the traditional attention mechanism, as well as the shortcomings of the Fourier transform that cannot provide effective localized information. In addition, the multi-branch structure designed by the model can realize the effective fusion of global-local periodic features of the time series on a multi-scale compared with the single-branch structure;

2) An attention guidance module based on "priori knowledge" is designed. The "priori knowledge" not noticed by other models is fully utilized, and is guided into the learning of the model through the mechanism of attention, which guides the transfer of relevant time series information;

3) A sequence denoising decomposition module is introduced. A bilateral filtering layer is added to the classical sequence decomposition module, which generates a noise residual and gradually decomposes the trend and seasonal terms at the same time. The reduction of the noise residual component is utilized to suppress the abrupt fluctuations of the anomalous data and enhance the robustness of the model;

4) In order to meet the practical needs of the ultra-short-term time series forecasting of power load, this paper also conducts preliminary screening for relevant variables to reduce the negative role of some relevant covariates. As well as increasing the sequence probability prediction, the results are upgraded from a point output to a probability output, and the model results are presented in the form of quantiles.

2. Materials and methods

The power load ultra-short-term time series forecasting task aims to utilize the historical sequence value X_L^N (where L represents the sequence length and N represents the sequence dimension) to forecast the future target sequence value O_S^M of length S (in this paper, the focus is on univariate forecasting, thus M = 1). As shown in Figure 3, within the framework of the sequence decomposition model, this paper emphasizes the difficulty that the original model cannot simultaneously capture complex global seasonal patterns as well as abrupt local seasonal features, especially for high-frequency load segments that are prone to large fluctuations. To address these issues, a time-frequency energy selection module, a "priori knowledge" guidance module, and a sequence denoising decomposition module are introduced. In the encoder part, the time-frequency energy selection module and the sequence denoising decomposition module are used to obtain the dependencies of the historical sequence X_L^N. In the decoder part, the prediction result of the sequence is generated using prior knowledge, the trend term with the mean value of the historical sequence as the initial value, the seasonal term with 0 as the initial value, and gradual separation and fusion. Additionally, the prediction results are upgraded from a point output to a probabilistic output. In the next section, six parts are introduced, namely, the time-frequency energy selection module, the "priori knowledge" guidance module, the sequence denoising decomposition module, the probabilistic load prediction, the evaluation metrics, and the data preprocessing methods.

2.1. Time-frequency energy selection module

In this paper, a time-frequency energy selection module based on power spectral density and the Morlet continuous wavelet transform is proposed. The module consists of three parts: a global energy selection module, a local energy selection module, and a period-weighted feature fusion module.

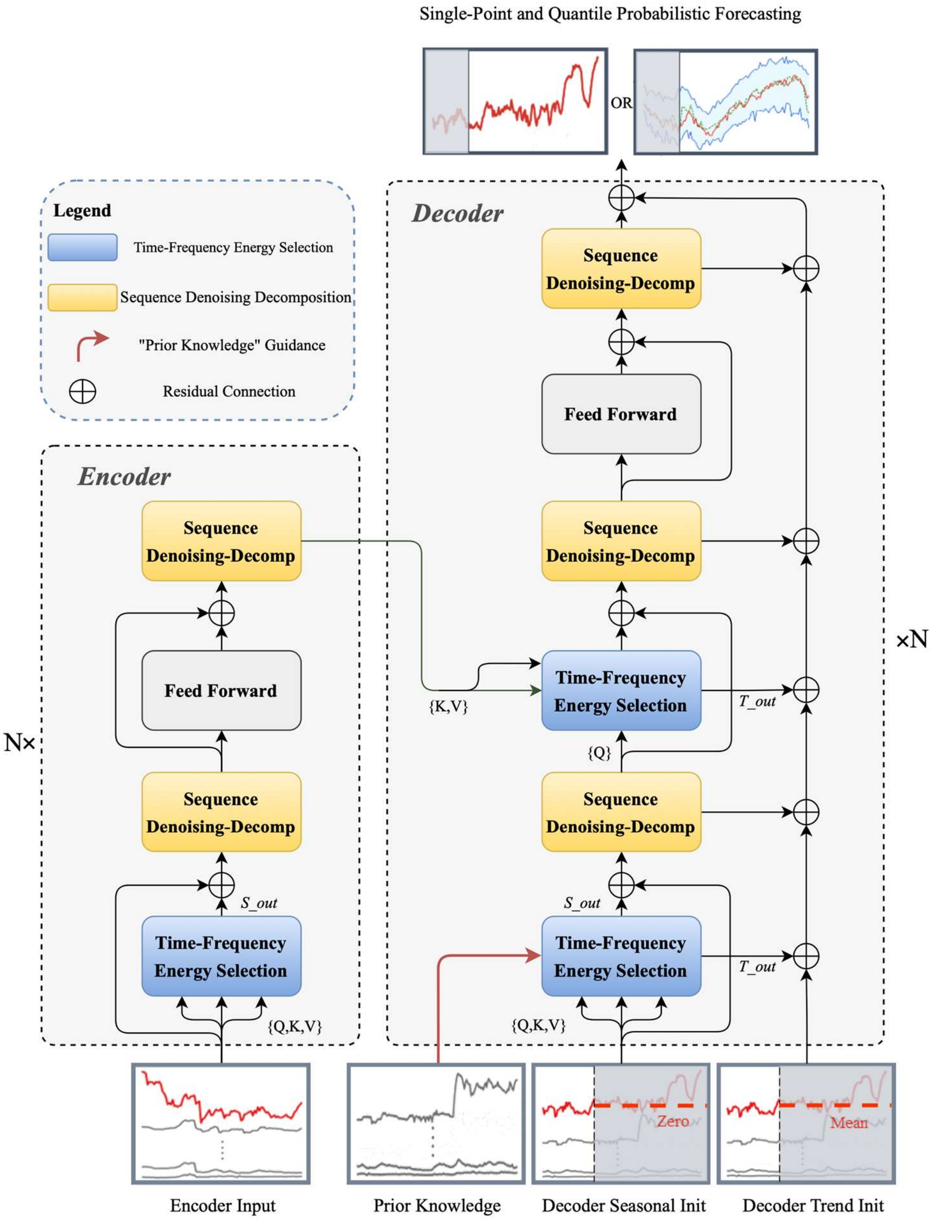
Single-Point and Quantile Probabilistic Forecasting
OR
Legend
Time-Frequency Energy Selection
Sequence Denoising Decomposition
"Prior Knowledge" Guidance
Residual Connection
Decoder
Encoder
Sequence Denoising-Decomp
Feed Forward
Time-Frequency Energy Selection
{K,V}
{Q}
{Q,K,V}
S_out
T_out
N×
×N
Zero
Mean
Encoder Input
Prior Knowledge
Decoder Seasonal Init
Decoder Trend Init

Figure 3. TFDNet model.

2.1.1. Global energy selection module

As shown in Figure 4, in the field of signaling, the magnitude of energy contained in different frequencies in the power spectral density plot reflects the degree of importance of the corresponding periodic subsequence in the original sequence. In this paper, the energy magnitude in the power spectral density plot is used to indicate the degree of structural similarity between the main periodic sequences and the original sequences, thus replacing the shape similarity metric in the original attention mechanism. In addition, a time-delay aggregation method [14] is used to fuse the information of sequences with different degrees of importance.

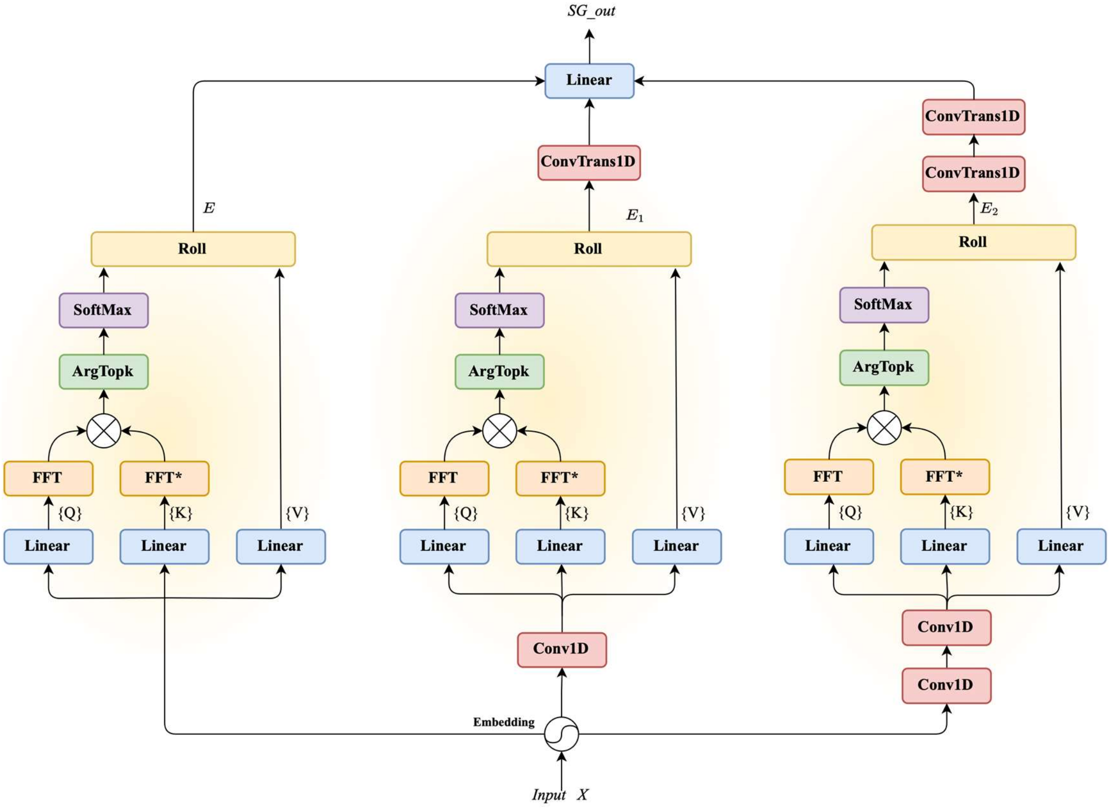

Figure 4. Global energy selection module.

According to Parseval's law and the Wiener-Hinchin theorem, the power spectral density of a discrete time series X_t of length L is shown in Eq (1):

$$\mathrm{PSD}(f) = \mathrm{FFT}(X_t) \cdot \mathrm{FFT}^*(X_t) = \left(\sum_{t=1}^{L} X_t \cdot e^{-\mathrm{j}2\pi\frac{tr}{L}}\right) \cdot \overline{\left(\sum_{t=1}^{L} X_t \cdot e^{-\mathrm{j}2\pi\frac{tr}{L}}\right)} \quad (1)$$

where PSD() represents the power spectral density function, t is the time index, $t = 1,2, \dots, L$, "*" is the conjugate operation, j is an imaginary unit, usually denoted as $\sqrt{-1}$, r is the frequency index, $r = 1,2,\dots,L$, and f stands for frequency, $f = 1,2,\dots,\frac{L}{2}$. Utilizing FFT() reduces the time and space complexities of solving the power spectral density.

According to the law of energy conservation, the time domain energy is equal to the frequency domain energy. Therefore, by utilizing the first k energy maximal signals f_i (where $i \in \{1,\dots,k\}$), the corresponding periodic subsequence can be calculated, and the specific process is shown Eqs (2) and (3):

$$\{f_1,\cdots,f_k\},\ \{w_1,\cdots,w_k\} = \mathrm{ArgTopk}\big(\mathrm{PSD}(f)\big) \tag{2}$$

$$p_i = \left\lceil \frac{L}{f_i} \right\rceil, i \in 1,\cdots,k \tag{3}$$

where ArgTopk() acts to select the frequency f_i and power spectral density value w_i corresponding to the top k values of power spectral density, and p_i represents the period corresponding to f_i. Here $k = c \cdot \log_e L$, where c is the hyperparameter. L represents the input sequence length. Finally, the time-delay aggregation technique and the attention mechanism are fused to maximize the mining of key correlations between sequences, as shown in Eqs (4) and (5):

$$\overline{w_1},\ldots,\overline{w_k} = \mathrm{SoftMax}(w_1,\cdots,w_k) \tag{4}$$

$$E(X,X,X) = \textstyle\sum_{i=1}^{k} \mathrm{Roll}(X,p_i)\overline{w_i} \tag{5}$$

where SoftMax() denotes the normalization operation, $\overline{w_i}$ represents the normalized power spectral density value of the sequence with frequency f_i, the $\mathrm{Roll}(X,p_i)$ function represents the delayed p_i operation on the sequence X, and E represents the output.

Meanwhile, inspired by computer vision network models such as UNet [19] and Feature Pyramid Network (FPN) [20], this module passes the time series X through multiple parallel 1D convolutional modules to realize the scaling of the sequences with different granularities; then, it directs the time series with different granularities to learn in parallel. The core steps are specified as follows:

```
**Algorithm**: Global energy selection module
**Input**: sequence value $X \in \mathbb{R}^{L\times N}$, where $L$ represents the length of the sequence, $N$ represents
the number of sequence dimensions, and $d_{model}$ represents the number of feature dimensions.

**Output**: $SG\_out$

1.$X_{emb} = \mathrm{Embedding}(X)$
                                                        $X \in \mathbb{R}^{L\times N}$, $X_{emb} \in \mathbb{R}^{L\times d_{model}}$
2.$X_{emb1} = \mathrm{conv1D}(X_{emb}), X_{emb2} = \mathrm{conv1D}(X_{emb1})$
                                                        $X_{emb1} \in \mathbb{R}^{\frac{L}{2}\times d_{model}}$, $X_{emb2} \in \mathbb{R}^{\frac{L}{4}\times d_{model}}$
3.**for** $j \in \{emb, emb1, emb2\}$ **do**   (The dimensional display in the loop is exemplified by $emb$.)
    3.1.$Q,K,V = \mathrm{Linear}(X_j), \mathrm{Linear}(X_j), \mathrm{Linear}(X_j)$
                                                        $Q,K,V \in \mathbb{R}^{L\times d_{model}}$
    3.2.$PSD_{QK} = \mathrm{FFT}(Q,\ dim = 0) \cdot \mathrm{FFT}^{*}(K, dim = 0)$
                                                        $PSD \in \mathbb{R}^{\frac{L}{2}\times d_{model}}$
    3.3.$frequency\_list, weight\_list = \mathrm{ArgTopk}(PSD_{QK})$
                                                        $frequency\_list, weight\_list \in k$
    3.4.$period\_list = \frac{L}{frequency\_list}$
```

```
                                                            $period\_list \in k$
    3.5. $\overline{weight\_list} = \text{SoftMax}(w_1,\cdots,w_k)$
                                                            $\overline{weight\_list} \in k$
    3.6. $E_j(Q,K,V) = \sum_{i=1}^{k} \text{Roll}(V,p_i)\overline{weight\_list}$
end for
4. $\overline{E_1} = \text{convtrans1D}(E_1), \overline{E_2} = \text{convtrans1D}(\text{convtrans1D}(E_2))$
        $E \in \mathbb{R}^{L\times d_{model}}, E_1 \in \mathbb{R}^{\frac{L}{2}\times d_{model}}, E_2 \in \mathbb{R}^{\frac{L}{4}\times d_{model}}, \overline{E_1} \in \mathbb{R}^{L\times d_{model}}, \overline{E_2} \in \mathbb{R}^{L\times d_{model}}$
5. $SG\_out = \text{Linear}(E,\ \overline{E_1},\ \overline{E_2})$
                                                            $SG\_out \in \mathbb{R}^{L\times d_{model}}$
```

2.1.2. Local energy selection module

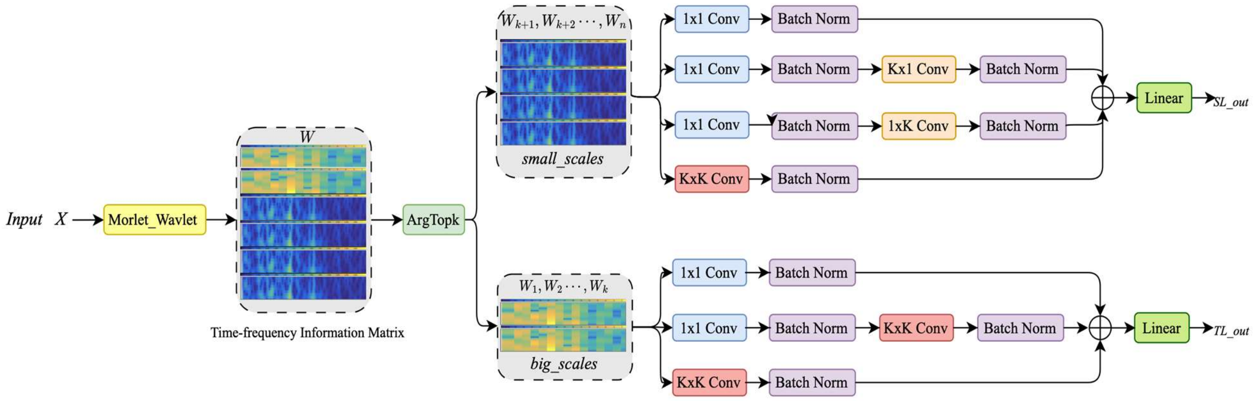

Figure 5. Local energy selection module.

As shown in Figure 5, the core of the local energy selection module is the multi-scale continuous wavelet transform with Morlet as the basis function. First, the time series X_t is the time-frequency transformed using the multi-scale Morlet transform to generate the time-frequency information matrix under different scales, as shown in Eq (6):

$$\mathrm{W}(a,b) = \left|\int_{-\infty}^{+\infty} X_t * \bar{\psi}_{a,b}(t)dt\right|^2 \tag{6}$$

where $\mathrm{W}(a,b)$ represents the time-frequency information matrix corresponding to the time node t and the translation factor b at the scale factor a, which controls the width of the wavelet function $\bar{\psi}(t)$. b is the translation factor, which controls the movement of the wavelet function. $\bar{\psi}_{a,b}(t)$ represents the Morlet wavelet function under the combination of (a,b), and $\int_{-\infty}^{+\infty}$ denotes the integration operation. "*" denotes the multiplication operation.

Different (a,b) combinations corresponding to $W(a,b)$ represent the extraction of different time-frequency features in the original sequence: a larger a represents a wider width of the Morlet wavelet function $\bar{\psi}(t)$, which is more capable of mining the long-term global sequence feature matrix; a smaller a represents a narrower width of the Morlet wavelet function $\bar{\psi}(t)$, which is more inclined to a localized, mutated sequence feature matrix. The specific formula for the Morlet wavelet function is shown in Eq (7):

$$\bar{\psi}_{a,b}(t) = \exp\left(-i\omega_0 \frac{(t-b)}{a}\right)\exp\left(-\frac{(t-b)^2}{2a^2}\right) \tag{7}$$

where ω_0 is the corner frequency parameter of the Morlet wavelet, a is the scale factor, and b is the translation factor. exp() is the exponential function.

Then, the ArgTopk() function is utilized to achieve feature extraction classification of the generated W(a,b) according to the magnitude of the scale factor a. The specific formulas are shown in Eqs (8) and (9):

$$\{W_1, W_2 \cdots, W_k\}, \{W_{k+1}, W_{k+2} \cdots, W_n\} = \text{ArgTopk}(W(a,b)) \tag{8}$$

$$big\_scales = \{W_1, W_2 \cdots, W_k\}, small\_scales = \{W_{k+1}, W_{k+2} \cdots, W_n\} \tag{9}$$

where the big_scales array is the global feature extraction of the time-frequency information matrix, and the $small\_scales$ array is the sequence localized feature extraction of the time-frequency information matrix. n represents the number of the scale factor a, $n = \frac{L}{\lceil 2^{\beta} \rceil}$, L represents the sequence length, and β is generally chosen as 1.5. $k = c \cdot \log_e n$, and c has the same meaning as described above.

Finally, to realize the output of the seasonal term SL_{out} and trend term TL_{out}, asymmetric and symmetric convolution are used to perform asymmetric and symmetric feature extraction for $small\_scales$ and big_scales, respectively,. The specific code is shown below:

Algorithm: Local energy selection module

Input: sequence value X$\in \mathbb{R}^{L\times N}$, where L represents the length of the sequence, N represents the number of sequence dimensions, and d_{model} represents the number of feature dimensions.

Output: SL_out、TL_out

```
1.$X_{emb} = \text{Embedding}(X)$
                                        X$\in \mathbb{R}^{L\times N}$, $X_{emb} \in \mathbb{R}^{L\times d_{model}}$
2.$W(a,b) = \left|\int_{-\infty}^{+\infty} X_t * \bar{\psi}_{a,b}(t)dt\right|^2$
                                        $W(a,b) \in \mathbb{R}^{L\times d_{model}}$
3.$\{W_1, W_2 \cdots, W_k\}, \{W_{k+1}, W_{k+2} \cdots, W_n\} = \text{ArgTopk}(W(a,b))$
                                        $W_k \in \mathbb{R}^{1\times d_{model}}$, $W_n \in \mathbb{R}^{1\times d_{model}}$
4.$big\_scales = \{W_1, W_2 \cdots, W_k\}, small\_scales = \{W_{k+1}, W_{k+2} \cdots, W_n\}$
                                        $big\_scales \in \mathbb{R}^{k\times d_{model}}, small\_scales \in \mathbb{R}^{(n-k)\times d_{model}}$
5.$SL\_out = \text{Linear}(S_1 + S_2 + S_3 + S_4)$ includes:
    5.1.$S_1 = \text{Batch Norm}(\text{Conv}_{1\times 1}(small\_scales))$
    5.2.$S_2 = \text{Batch Norm}\left(\text{Conv}_{k\times 1}\left(\text{Batch Norm}(\text{Conv}_{1\times 1}(small\_scales))\right)\right)$
```

```
        5.3. $S_3 = \text{Batch Norm}\left(\text{Conv}_{1\times k}\left(\text{Batch Norm}\left(\text{Conv}_{1\times 1}(small\_scales)\right)\right)\right)$
        5.4. $S_4 = \text{Batch Norm}\left(\text{Conv}_{k\times k}(small\_scales)\right)$
    6. $TL\_out = \text{Linear}(T_1 + T_2 + T_3)$ **includes**:
        6.1. $T_1 = \text{Batch Norm}\left(\text{Conv}_{1\times 1}(big\_scales)\right)$
        6.2. $T_2 = \text{Batch Norm}\left(\text{Conv}_{k\times k}\left(\text{Batch Norm}\left(\text{Conv}_{1\times 1}(big\_scales)\right)\right)\right)$
        6.3. $T_3 = \text{Batch Norm}\left(\text{Conv}_{k\times k}(big\_scales)\right)$
```

2.1.3. Period-weighted feature fusion module

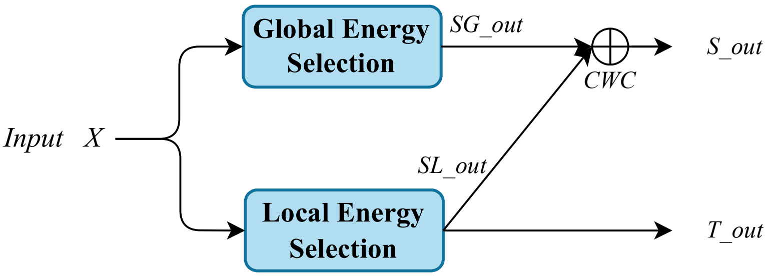

Figure 6. Period-weighted feature fusion module.

As shown in Figure 6, the period-weighted feature fusion module mainly utilizes the cycle weighting coefficients (CWC) to achieve adaptive balancing of the periodicity of the global power spectral density, as well as the periodicity of the local Morlet wavelet transform, as shown in Eq (10):

$$\gamma = \text{SoftMax}\left(W_q \cdot \tanh\left(W_g * SG_{out} + W_l * SL_{out}\right)\right) * SL_{out} \tag{10}$$

where the cycle weighting coefficient (CWC) is denoted by γ and SG_out and SL_out represent the outputs of the global energy selection module and the local energy selection module, respectively. W_q, W_g, and W_l are learnable parameter matrices for transforming the global and local features, and $tanh$ is the activation function. By learning the weights of γ, the model can adaptively adjust the proportionality of the global and local features.

2.2. "Prior knowledge" guidance module

As shown in Figure 7, in order to effectively utilize the relevant covariate information in the future "a priori" within the decoder module, in this paper, the relevant covariate sequences G and the seasonal term sequences S, which are initialized to 0, are used as inputs to the global energy selection module. $G = g^n_{\frac{L}{2}}, g^n_{\frac{L}{2}+1}, \dots, g^n_L, g^n_{L+1}, \dots, g^n_{L+\tau}$, where $g^n_{L+\tau}$ stands for the nth covariate value of the prediction interval length τ. $S = \{s_{\frac{L}{2}}, s_{\frac{L}{2}+1}, \dots, s_L, 0, \dots, 0\}$, where $s_{\frac{L}{2}+i}$ is aligned with the input value $X_{\frac{L}{2}+i}$ of the encoder module. Then, the global energy selection module learns the correlations $PSD_{\overline{QK}}$

for the G-sequence and PSD_{QK} for the S-sequence. Finally, the deterministic covariate power spectral density maps $PSD_{\bar{Q}\bar{K}}$ are used to correct and guide the original power spectral density maps PSD_{QK} with appropriateness.

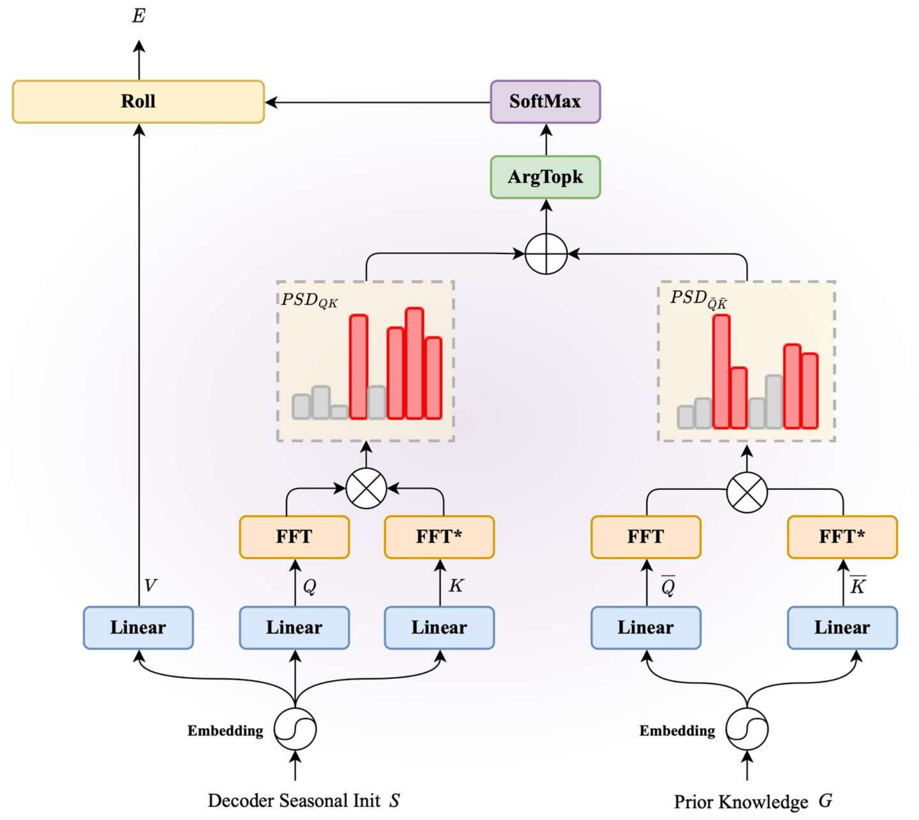

Figure 7. “Prior knowledge” guidance module.

This module employs G and S as part of the input to the decoder. It differs from the global energy selection module in Encoder by introducing a knowledge-guided branch G. The core steps are as follows:

Algorithm: “Prior knowledge” guidance module

Input: a sequence of covariates $G \in \mathbb{R}^{(\frac{L}{2}+\tau)\times M}$, and a sequence of seasonal terms $S \in \mathbb{R}^{(\frac{L}{2}+\tau)\times N}$ initialized to 0. L represents the length of the input sequence in the encoder, τ represents the length of the prediction, and d_{model} represents the size of the feature dimensions. k represents the number of selection period subsequences, N represents the number of sequence dimensions, and M represents the number of “known” relevant covariates in the future.

Output: E

1.$S_{emb} = \text{Embedding}(S), G_{emb} = \text{Embedding}(G)$

$$S_{emb} \in \mathbb{R}^{(\frac{L}{2}+\tau)\times d_{model}}, S_{emb} \in \mathbb{R}^{(\frac{L}{2}+\tau)\times d_{model}}$$

```
2.$Q,K,V = \text{Linear}(S_{emb}),\text{Linear}(S_{emb}),\text{Linear}(S_{emb})$
                                                    Q,K,V$\in \mathbb{R}^{(\frac{L}{2}+\tau)\times d_{model}}$
3. $\bar{Q},\bar{K} = \text{Linear}(G_{emb}),\text{Linear}(G_{emb})$
                                                    $\bar{Q},\bar{K} \in \mathbb{R}^{(\frac{L}{2}+\tau)\times d_{model}}$
4.$PSD_{QK} = \text{FFT}(Q)\cdot\text{FFT}^*(K), PSD_{\bar{Q}\bar{K}} = \text{FFT}(\bar{Q})\cdot\text{FFT}^*(\bar{K})$
                                                    $PSD_{QK},PSD_{\bar{Q}\bar{K}} \in \mathbb{R}^{(\frac{L}{4}+\frac{\tau}{2})\times d_{model}}$
5.$frequency\_list, weight\_list = \text{ArgTopk}\left(PSD_{QK} + PSD_{\bar{Q}\bar{K}}\right)$
                                                    $frequency\_list, weight\_list \in k$
6.$period\_list = \frac{\frac{L}{2}+\tau}{frequency\_list}$
                                                    $period\ \ \_list \in k$
7.$\overline{weight\_list} = \text{SoftMax}(w_1,\cdots,w_k)$
                                                    $\overline{weight\_list} \in k$
8.E$(Q,K,V,\bar{Q},\bar{K}) = \sum_{i=1}^{k} \text{Roll}(V,p_i)\overline{weight\_list}$
                                                    E$\in \mathbb{R}^{(\frac{L}{2}+\tau)\times d_{model}}$
```

2.3. Sequence denoising decomposition module

In order to better learn the decomposition pattern in the complex background, this module adopts a method based on the idea of sequence decomposition and bilateral filtering [21] to decompose the sequence into trend terms, seasonal terms, and residual noise components. As shown in Figure 8, without affecting the peaks and valleys of the series, the module gradually removes the appropriate amount of residual noise components through the average pooling layer and the bilateral filtering layer, so as to gradually decompose the series into trend terms and seasonal terms and capture the long-term profile and local information of the time series.

The multiple extractions of the trend term $Trend$ and separation of the noisy seasonal term $Seasonal'$ are first performed on the sequence S_out by multiple average pooling layers with different window sizes. The details are shown in Eqs (11) and (12):

$$Trend = \text{Linear}\big(\text{AveragePooling}(S\_out)\big) \tag{11}$$

$$Seasonal' = S\_out - Trend \tag{12}$$

where AveragePooling() represents the average pooling operation.

Then, the seasonal term $Seasonal$ is generated by using bilateral filtering to process the noisy sequence $Seasonal'$, thus realizing the elimination of the residual noise. The specific formulas are shown in Eqs (13) and (14):

$$Seasonal = \sum_{i\in I} w_i^t * (Seasonal')_i \quad ,I = [(t-M)_+,(t+M)_-] \tag{13}$$

$$w_i^t = e^{\left(-\frac{|i-t|^2}{2\delta_d^2}\right)} e^{\left(-\frac{|(Seasonal')_i-(Seasonal')_t|^2}{2\delta_i^2}\right)} \tag{14}$$

where $(t-M)_+$ denotes $\max(0, t-M)$ and $(t+M)_-$ denotes $\min(t+M, L)$. In this procedure, w_i^t represents the filtering weights under the time index t and the window index i, the filter window size I is set to $2M+1$, the initial values of the bilateral filter parameters δ_d and δ_i are 1.0 and 1.0, respectively, and M is set to 12.

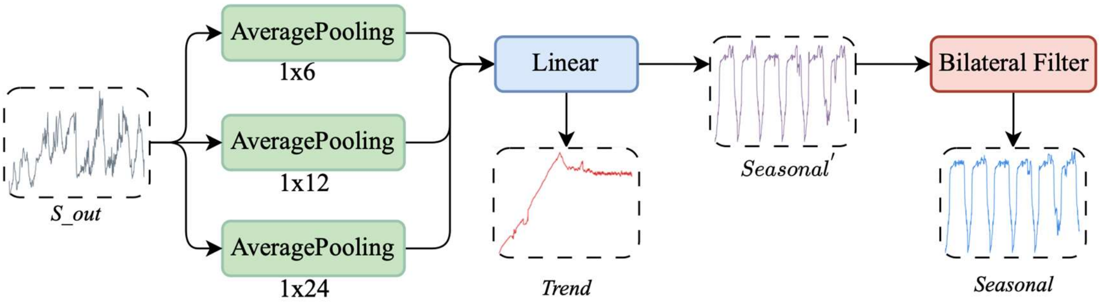

Figure 8. Sequence denoising decomposition module.

2.4. Probabilistic load prediction

Practical industrial environments frequently require predictive assessments for managing risk outcomes. Probability forecasting offers an effective approach to quantitatively balance risks. For example, in the day-ahead scheduling of the power industry, operators utilize probability forecasts to reasonably allocate peak load capacity, thus aiming to conserve resources. In power trading, traders can optimize pricing strategies using uncertain forecast information to maximize profits. Therefore, in this paper, quantile loss [22] is utilized to achieve the output of the prediction interval by setting different output quantile parameters α. The final probability loss function is achieved by training all parameters α to minimize the quantile loss. The probability loss function is shown in Eqs (15) and (16):

$$\mathcal{L}(\Omega, \boldsymbol{W}) = \sum_{y_t\in\Omega}\sum_{\alpha\in A}\sum_{\tau=1}^{\tau_{max}} \frac{\mathrm{AL}(y_t,\hat{y}(\alpha,t-\tau,\tau),\alpha)}{M\tau_{max}} \tag{15}$$

$$\mathrm{AL}(y,\hat{y},\alpha) = \alpha(y-\hat{y})_+ + (1-\alpha)(\hat{y}-y)_+ \tag{16}$$

where the training domain Ω consists of M training data samples, W represents the model parameters, AL() represents the quantile loss, and α is the output quantile parameter. A represents the set of values of the quantile parameter α. Here, y, $\hat{y}$, and τ represent the true value, predicted value, and prediction interval, respectively. The notation $(x)_+$ denotes $\max(0, x)$.

To compare the quantile loss associated with different α parameters in the testing domain $\tilde{\Omega}$, evaluation metrics "$\alpha - Risk$" are defined as follows:

$$\alpha - Risk = \frac{2\sum_{y_t\in\tilde{\Omega}}\sum_{\tau=1}^{\tau_{max}} \mathrm{AL}(y_t,\hat{y}(\alpha,t-\tau,\tau),\alpha)}{\sum_{y_t\in\tilde{\Omega}}\sum_{\tau=1}^{\tau_{max}}|y_t|} \tag{17}$$

where "*α-Risk*" represents the quantile loss values for different values of parameter α. y, $\hat{y}$, τ, and α have the same meanings as previously described, and $\tilde{\Omega}$ represents the testing domain.

2.5. Evaluation metrics

In addition to the use of "*α-Risk*" as an evaluation metric for probabilistic load forecasting, the mean square error loss function (MSE) and the mean absolute error loss function (MAE) are used as evaluation metrics for point forecasting. The formulas for the MSE and MAE are shown in Eqs (18) and (19):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{18}$$

where MSE calculates the squared difference between the predicted value and the true value. n is the number of samples, y_i is the true value, and $\hat{y}_i$ is the predicted value.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{19}$$

where MAE calculates the absolute difference between the predicted value and the true value. Again, n is the sample size, y_i is the true value, and $\hat{y}_i$ is the predicted value.

MSE is characterized by faster convergence speeds but is more sensitive to outliers, thus magnifying the differences between errors. MAE is characterized as being less susceptible to outliers, with all errors based on the same weights but with a slower convergence speed. Therefore, in order to comprehensively and fully assess the model performance, this paper adopts MSE and MAE as the evaluation indexes for point prediction.

2.6. Data preprocessing methods

In the field of time series forecasting, too many covariates may lead to a dimensional catastrophe, which increases the temporal and spatial complexity of the model, causes the model to be difficult to train and generalize, and even negatively affects the predicted series; thus, the preliminary screening of covariates is a common method. In this paper, the Pearson correlation coefficient is used to screen continuous variables, and the Correlation Ratio is used to screen discrete variables, so as to realize the pre-processing of data.

$\rho_{X_i,Y}$ represents the Pearson correlation coefficient between the continuous covariate series X_i and the predicted series Y. The formula for $\rho_{X_i,Y}$ is shown in Eq (20):

$$\rho_{X_i,Y} = \frac{\mathrm{E}(X_iY) - \mathrm{E}(X_i)\mathrm{E}(Y)}{\sqrt{\mathrm{E}({X_i}^2) - (\mathrm{E}(X_i))^2}\sqrt{\mathrm{E}(Y^2) - (\mathrm{E}(Y))^2}} \tag{20}$$

where X_i and Y denote the ith continuous covariate sequence and the predicted sequence, respectively, and E() represents the mean operation.

η represents the value of the correlation ratio between the discrete covariate series and the predicted series. The formulas for η are shown in Eqs (21) and (22):

$$\eta = \sqrt{\frac{\sum_x n_x^i \left(\bar{y}_x^i - \bar{y}^i\right)^2}{\sum_{x,j} \left(y_{xj}^i - \bar{y}^i\right)^2}} \tag{21}$$

$$\bar{y}_x^i = \frac{\sum_j y_{xj}^i}{n_x^i}, \quad \bar{y}^i = \frac{\sum_x n_x^i \bar{y}_x^i}{\sum_x n_x^i} \tag{22}$$

where y_{xj}^i represents the jth predicted sequence value of type x in the ith sequence of discrete covariates. n_x^i represents the number of categories of x in the ith sequence of discrete covariates. $\bar{y}_x^i$ represents the predicted mean value of category x in the ith sequence, and $\bar{y}^i$ represents the predicted mean value of all the categories represented in the ith sequence. $\sum_x$ represents the accumulation of the weighted variance of the category means, and $\sum_{x,j}$ represents the accumulation of the variance of all discrete series.

3. Results

3.1. Datasets

TFDNet was extensively evaluated on four datasets, including three publicly available benchmark datasets and one private dataset. Here is a brief introduction to these datasets:

Electricity Transformer Dataset (ETT): The ETT dataset is a crucial indicator of long-term electricity allocation. It consists of data collected at 15-minute intervals for two years, from July 2016 to July 2018, in two counties in China. Experiments on ETTm1 (15 minutes) and ETTm2 (15 minutes) were conducted based on the collection location. Each data point is comprised of the target value "Oil Temperature" and six related features. The dataset is divided into training, validation, and test sets in a 7:1:2 ratio.

Weather: This dataset contains data from the meteorological field, collected at 10-minute intervals, covering observations from January 2020 to January 2021. Each data point includes the target value "OT" and 20 related features. The dataset is divided into training, validation, and test sets in a 7:1:2 ratio.

PowerLoad: This dataset records the total power load values in a city-level region in China. The data is collected at 15-minute intervals, spanning from October 2020 to October 2022. Each data point includes the target value "consumption" along with 10 related features, namely "type", "temperature", "humidity", "wind speed", "windscale", "wind angle", "wind direction", "air pressure", "visibility", and "precipitation". Similarly, the dataset is divided into training, validation, and test sets in a 7:1:2 ratio.

3.2. Baselines

This study selected the following six time series forecasting methods as benchmark comparisons: Informer, Autoformer, FEDformer, TimesNet, RLF-MGNN, and EEMD-DARNN. These six methods represent different approaches and techniques in the field of time series forecasting, thus providing a comprehensive set of benchmarks for experimentation and comparison.

3.3. Implementation details

In this experiment, training was conducted using three different loss functions: mean squared error (MSE), mean absolute error (MAE), and quantile loss (probability loss). The adaptive moment estimation (ADAM) optimization method was employed, with an initial learning rate of 10^{-4}. The batch size was set to 32. The training process was early and stopped within 10 epochs. All experiments were repeated five times. The PyTorch platform was used, and a single NVIDIA TITAN XP 12GB GPU was utilized. The set of values of the quantile parameter α was $\{0.1,0.5,0.9\}$.

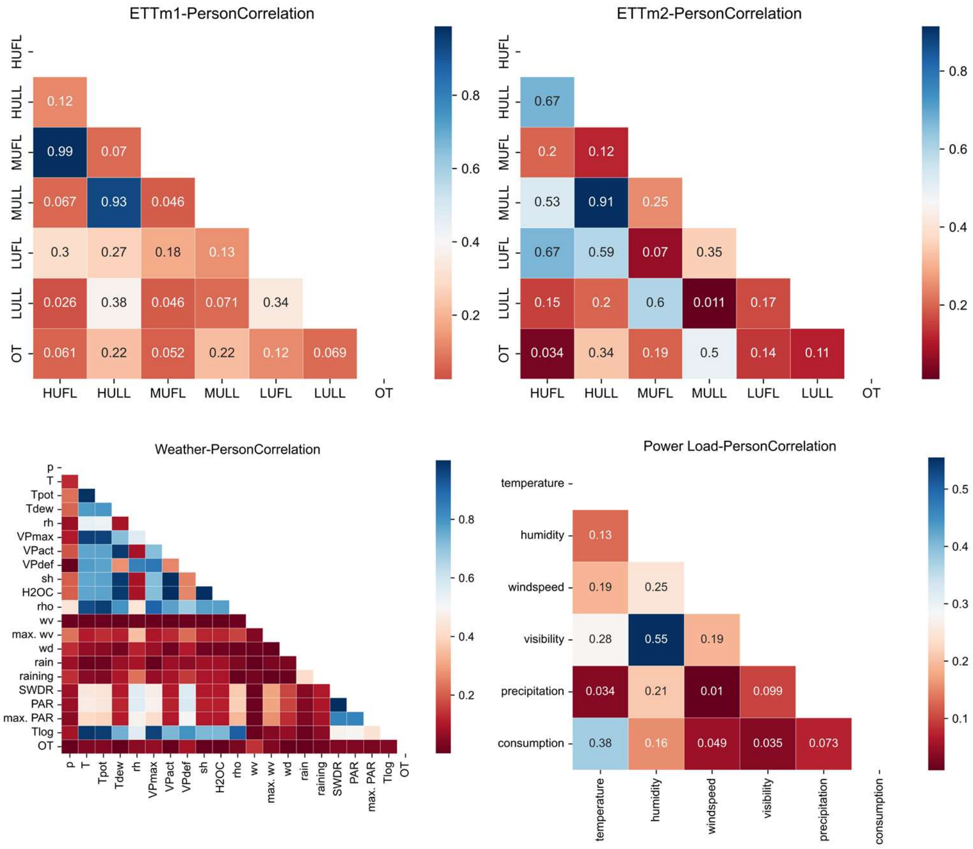

Figure 9. Heatmaps of Pearson correlation coefficients on the four datasets.

3.4. Pretreatment

Since only continuous variables exist in the ETTm1, ETTm2, and Weather datasets, only ρ was used for these three datasets. The computation and ranking of ρ were conducted for each covariate series within the predicted series, with the subsequent selection of the top u covariates as inputs to the model. In the case of the PowerLoad dataset, both ρ and η were utilized for the screening method. The calculation and ranking of ρ for each continuous covariate series were executed, along with the calculation and ranking of η for each discrete covariate series. Ultimately, the top u continuous covariates and v discrete covariates were chosen as inputs to the model.

This experiment adopted different covariate screening strategies for different datasets. Covariates with u = 3 were selected for the ETTm1 and ETTm2 datasets, covariates with u = 12 were selected for the Weather dataset, and continuous covariates with $u = 6$ and discrete covariates with $v = 1$ were chosen for the Powerload dataset. As shown in Figures 9 and 10, the heat maps indicate the degree of share of Pearson correlation coefficients for the continuous covariates in the four datasets, while the pie chart indicates the degree of share of the correlation ratio for the discrete covariates in the PowerLoad dataset.

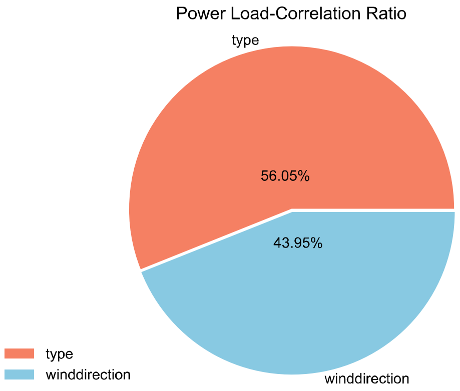

Figure 10. Pie chart of correlation ratio on PowerLoad dateset.

3.5. Comparison experiments

Table 1 summarizes the experimental results of the seven models on the four datasets. The MSE and MAE loss functions were used as evaluation metrics, where smaller values indicate more accurate predictions. Additionally, this experiment extends the prediction window length to test the models' prediction stability. The best experimental results are highlighted in bold.

When vertically observing Table 1, significant improvements across all datasets are evident for TFDNet, thereby displaying the highest "count" value (count represents the number of times each model achieved the best performance across all tasks). The prediction errors slowly increase with an extended prediction window, thus demonstrating the positive stability and scalability of TFDNet in time series forecasting.

When horizontally examining Table 1, the superior performance of TFDNet over Informer, Autoformer, FEDformer, TimesNet, RLF-MGNN, and EEMD-DARNN is evident in terms of both MSE and MAE. Compared to these models, TFDNet reduces the average MSE by 36.59, 29.56, 20.34, 9.68, 15.25, and 7.35% across the four datasets, respectively, thus indicating its superiority over the other forecasting models. Additionally, the effectiveness of the time-frequency domain analysis relative to the single-dimension analysis is verified.

Figure 11 displays the predicted curves and true value curves for the seven models across the four datasets, each with a prediction length of 96. From the graph, it's evident that Informer, Autoformer, FEDformer, and RLF-MGNN have relatively poor performances. TimesNet captures the trend well but exhibits significant fluctuations. Both EEMD-DARNN and TFDNet fit GroundTruth better and capture long-term trends and seasonal shifts of the series in a timely manner, but TFDNet fits local peaks and troughs better relative to EEMD-DARNN.

Table 1. MSE and MAE values for models with an input series length of 96 and output series lengths of {96, 192, 336, 720}.

Methods		Informer		Autoformer		FEDformer		TimesNet		RLF-MGNN		EEMD-DARNN		TFDNet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.111	0.278	0.055	0.185	0.038	0.150	0.037	0.148	0.038	0.151	**0.031**	**0.140**	0.033	0.142
	192	0.151	0.313	0.080	0.216	0.065	0.204	0.063	0.201	0.065	0.199	0.060	**0.194**	**0.058**	0.209
	336	0.428	0.595	0.087	0.218	0.074	0.213	0.071	0.211	0.072	0.213	0.071	0.215	**0.067**	**0.210**
	720	0.438	0.588	0.111	0.264	0.107	0.254	0.094	0.231	0.100	0.253	0.093	0.241	**0.088**	**0.225**
ETTm2	96	0.088	0.225	0.068	0.189	0.067	0.191	0.067	0.192	0.071	0.203	0.063	0.191	**0.061**	**0.187**
	192	0.134	0.283	0.119	0.258	0.114	0.249	0.101	0.244	0.104	0.251	0.109	0.248	**0.094**	**0.241**
	336	0.180	0.337	0.154	0.307	0.140	0.303	0.135	0.300	0.143	0.299	0.134	0.294	**0.122**	**0.281**
	720	0.302	0.440	0.184	0.343	0.215	0.373	0.192	0.351	0.198	0.355	0.185	0.333	**0.172**	**0.329**
Weather	96	0.0042	0.046	0.018	0.082	0.0037	0.048	0.0033	0.041	0.0038	0.047	0.0031	0.038	**0.0029**	**0.031**
	192	**0.0025**	**0.043**	0.0073	0.072	0.0058	0.062	0.0043	0.051	0.0048	0.059	0.0045	0.055	0.0040	0.044
	336	0.0044	0.052	0.0065	0.064	0.008	0.077	0.0057	0.068	0.0073	0.067	0.0056	0.064	**0.0041**	**0.049**
	720	0.0043	0.049	0.0085	0.074	0.018	0.094	0.0049	0.081	0.0051	0.064	0.0041	0.049	**0.0036**	**0.047**
PowerLoad	96	0.367	0.362	0.254	0.264	0.204	0.234	0.188	0.217	0.193	0.305	**0.172**	**0.211**	0.178	0.325
	192	0.422	0.424	0.303	0.345	0.283	0.345	0.250	**0.319**	0.288	0.351	0.284	0.350	**0.232**	0.355
	336	0.534	0.583	0.403	0.452	0.386	0.395	0.385	**0.370**	0.395	0.401	0.391	0.393	**0.381**	0.373
	720	0.644	0.743	0.534	0.537	0.430	0.434	0.427	0.431	0.447	0.453	0.436	0.447	**0.423**	**0.427**
Count		1	1	0	0	0	0	0	2	0	0	2	3	13	10

Table 2. Values of probability loss function for input sequence length of 96 and output sequence length of 96.

Model	ETTm1			ETTm2		
Metric	P10	P50	P90	P10	P50	P90
Autoformer	0.203 (+17.73%)	0.248 (+28.22%)	0.104 (+42.3%)	0.220 (+27.73%)	0.237 (+19.41%)	0.112 (+34.82%)
TimesNet	0.188 (+11.17%)	0.205 (+13.17%)	0.081 (+25.93%)	0.207 (+23.19%)	0.214 (+10.75%)	0.096 (+23.96%)
RLF-MGNN	0.195 (+14.36%)	0.227 (+21.59%)	0.088 (+31.82%)	0.213 (+25.35%)	0.224 (+14.73%)	0.101 (+27.72%)
EEMD-DARNN	0.176 (+5.11%)	0.194 (+8.25%)	0.077 (+22.08%)	0.194 (+18.04%)	0.204 (+6.37%)	0.088 (+17.05%)
TFDNet	0.167	0.178	0.060	0.159	0.191	0.073
Model	Weather			PowerLoad		
Metric	P10	P50	P90	P10	P50	P90
Autoformer	0.053 (+28.30%)	0.073 (+21.92%)	0.029 (+27.59%)	0.135 (+14.81%)	0.148 (+9.45%)	0.121 (+28.09%)
TimesNet	0.049 (+22.45%)	0.062 (+8.06%)	0.025 (+16%)	0.118 (+2.54%)	0.139 (+3.60%)	0.091 (+4.40%)
RLF-MGNN	0.051 (+25.49%)	0.070 (+18.57%)	0.029 (+27.59)	0.127 (+9.45%)	0.144 (+6.94%)	0.104 (+16.35%)
EEMD-DARNN	0.041 (+7.32%)	0.061 (+6.56%)	0.022 (+4.55%)	0.119 (+3.36%)	0.138 (+2.90%)	0.093 (+6.45%)
TFDNet	0.038	0.057	0.021	0.115	0.134	0.087

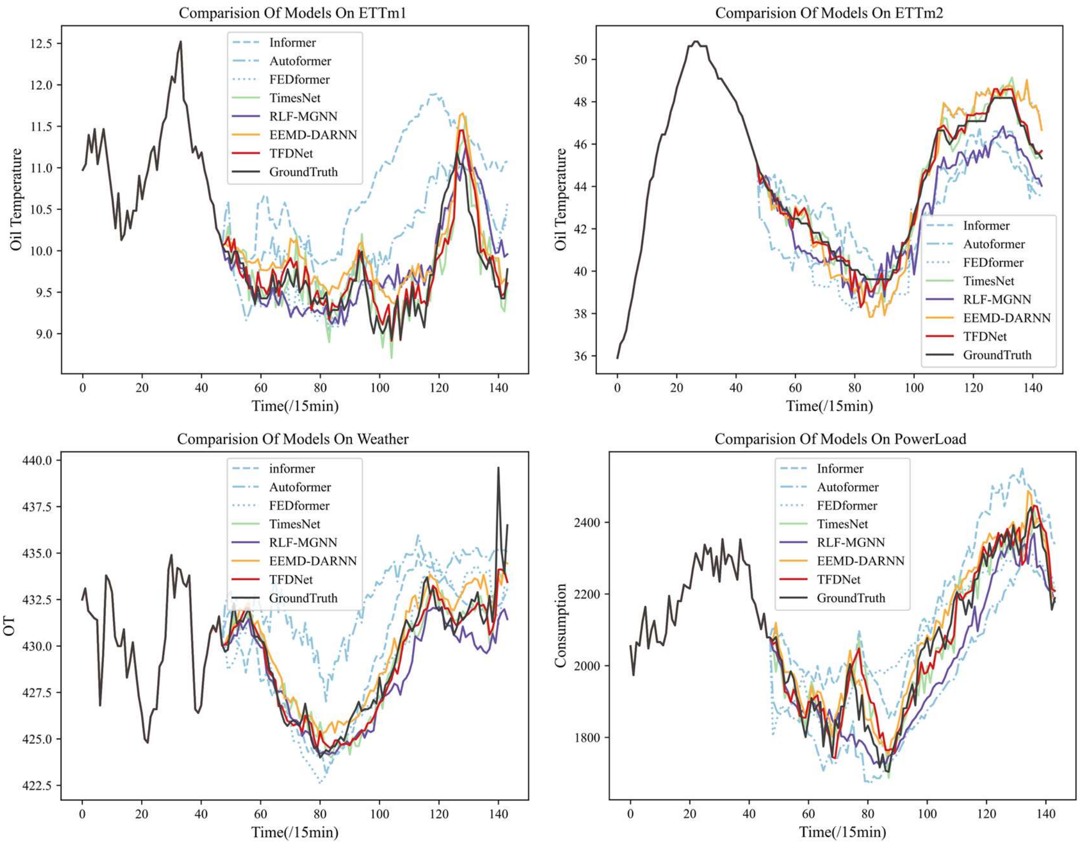

Figure 11. Prediction results for each model with an output length of 96.

Table 2 summarizes the experimental results of the probabilistic prediction of the five models on the four datasets, with the quantile loss "α-*Risk*" as the evaluation index of the probabilistic prediction. The values of α are taken as {0.1,0.5,0.9}, corresponding to {P10,P50,P90} quantile losses, respectively. A lower value indicates more accurate predictions. The percentages in parentheses represent the improvement achieved by the TFDNet model over the comparison models, namely Autoformer, TimesNet, RLF-MGNN, and EEMD-DARNN.

When horizontally observing the results in Table 2, the TFDNet proposed in this paper demonstrates a significant improvement in P10, P50, and P90 compared to Autoformer, TimesNet, RLF-MGNN, and EEMD-DARNN. TFDNet on the ETTm1 dataset improves 29.42, 16.76, 22.59, and 11.81% relative to Autoformer, TimesNet, RLF-MGNN, and EEMD-DARNN, respectively. On the ETTm2 dataset, TFDNet improves 27.32, 19.30, 22.60, and 13.82% relative to Autoformer, TimesNet, RLF-MGNN, and EEMD-DARNN, respectively. On the Weather dataset, TFDNet improves 25.94, 15.50, 23.88, and 6.14% relative to Autoformer, TimesNet, RLF-MGNN, and EEMD-DARNN, respectively. On the PowerLoad dataset, TFDNet improves by 17.46, 3.51, 10.91, and 4.24% relative to Autoformer, TimesNet, RLF-MGNN, and EEMD-DARNN, respectively.

When vertically observing the results in Table 2, TFDNet outperforms the other four benchmark models in all datasets. The results show that TFDNet is not only suitable for point prediction, but also for probabilistic prediction problems. On the ETTm1 dataset, TFDNet improves on average by 12.09, 17.81, and 30.53% in P10, P50, and P90, respectively. On the ETTm2 dataset, TFDNet improves on average by 23.58, 12.82, and 25.89% in P10, P50, and P90, respectively. On the Weather dataset, TFDNet improves on average by 20.89, 13.78, and 18.93% in P10, P50, and P90, respectively. On the PowerLoad dataset, TFDNet improved by an average of 7.54, 5.72, and 13.82% in P10, P50, and P90, respectively.

As shown in Figure 12, the probability prediction outputs of TFDNet and TimesNet on the ETTm1 and PowerLoad datasets are presented. From the graph, it can be observed that at the P50 quantile, TFDNet (green line) demonstrates a better fit to the ground truth (red line) as compared to TimesNet (gray line). Furthermore, in the P10-P90 quantile range, the range represented by TFDNet (blue area) is both comprehensive and accurate in comparison to TimesNet (pink area).

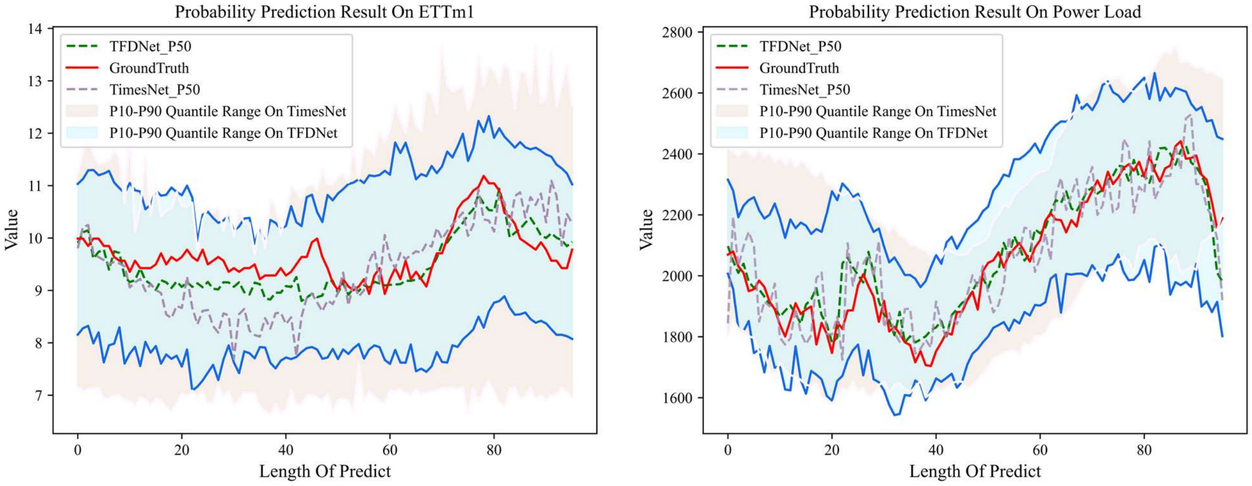

Figure 12. Output of probabilistic prediction results of length 96.

3.6. Ablation studies.

As shown in Table 3, in order to verify the effectiveness of 1) the sequence denoising decomposition module, 2) the time-frequency energy selection module, and 3) the "priori knowledge" guidance module, ablation experiments were designed on the ETTm1 dataset and the PowerLoad dataset. The length of the input sequence was 96, and the length of the predicted output was {96, 192, 336, 720}. The evaluation metrics were based on the MSE loss function. Autoformer was used as a prototype, and 1–8 represent eight cases of 000, 001, 010, 011, 100, 101, 110, and 111, respectively (1 represents using the module substitution in the corresponding position, and 0 represents not using the module substitution in the corresponding position). "–" means no experimental results because there was no "a priori" covariate information in the ETTm1 dataset, so the "priori knowledge" guidance module was not used.

As can be seen from Table 3, the "priori knowledge" guidance module, the time-frequency energy selection module, and the sequence denoising decomposition module all produce different degrees of improvement in the experimental results. For the ETTm1 dataset, the two variants (3 and 5) improve the results by 24.25 and 17.53% for 1, respectively. For the PowerLoad dataset, three variants (2, 3, and 5) improved the results by 0.79, 14.70, and 2.75% against 1, respectively.

As can be seen from Table 3, the three modules have different levels of performance improvement for the experimental results. In the case of the ETTm1 dataset, a comparison among 1, 3, and 5 reveals that the time-frequency energy selection module leads to the most significant improvement in experimental results. Regarding the PowerLoad dataset, the time-frequency energy selection module yields the greatest improvement, followed by the sequence denoising decomposition module, and the “priori knowledge” guidance module displays the smallest enhancement.

Table 3. Ablation experiments conducted to validate the necessity of the three modules.

DataSet	ETTm1				PowerLoad			
Predicted Length	96	192	336	720	96	192	336	720
1	0.055	0.080	0.087	0.111	0.254	0.303	0.403	0.534
2					0.250	0.302	0.401	0.530
3	0.037	0.061	0.069	0.089	0.194	0.257	0.388	0.447
4					0.188	0.252	0.386	0.445
5	0.041	0.069	0.075	0.092	0.239	0.298	0.399	0.521
6	–	–	–	–	0.231	0.292	0.398	0.516
7	–	–	–	–	0.180	0.241	0.382	0.428
8	0.033	0.058	0.067	0.088	0.178	0.232	0.381	0.423

4. Discussion and analysis

The baseline methods mentioned in the paper generally have the disadvantages of being difficult to capture local mutation features and having poor stability. Therefore, this paper will further analyze and verify the effectiveness of the time-frequency energy selection module in solving the difficulty problem in capturing local mutation features, the effectiveness of the “priori knowledge” guidance module, the sequence denoising decomposition module in solving the problem of poor stability, and the reasonableness of the model parameters. In the following, this paper analyzes the distribution of the time-frequency energy selection module, the distribution of the “priori knowledge” guidance module, the analysis of the sequence denoising decomposition module, the distribution of the predicted output sequences, and the model parameters for further experimental analysis.

4.1. Time-frequency energy selection module distribution analysis

4.1.1. Global distribution analysis

The power spectrum energy distribution of the global energy selection module on the ETTm1, ETTm2, Weather, and PowerLoad datasets is further demonstrated, as shown in Figure 13. When the input branch length L is 96, the orange part (the larger value of power spectral energy) is relatively concentrated, indicating that the model captures more relevant sequences in the closest time at this point. When the input branch length $\frac{L}{2}$ is 48, the orange part is more dispersed, which indicates that the model captures relevant sequences in the relatively distant time. When the input branch length $\frac{L}{4}$ is 24, the orange part is most dispersed, which indicates that the model mainly captures relevant sequences in the farthest time. This is due to the multi-branching structure in the global energy selection module and the accurate selection of periodic subsequences with different granularities by the power spectral density. When changing the length of the input sequence, the power spectral density is able to mine the deeper periodic relationships in the original sequence; then, through the effective fusion between the multi-granularity sub-sequences, PSDformer is able to comprehensively capture the sequence correlations in the short-, intermediate-, and long-term, which further validates the effectiveness of the global energy selection module.

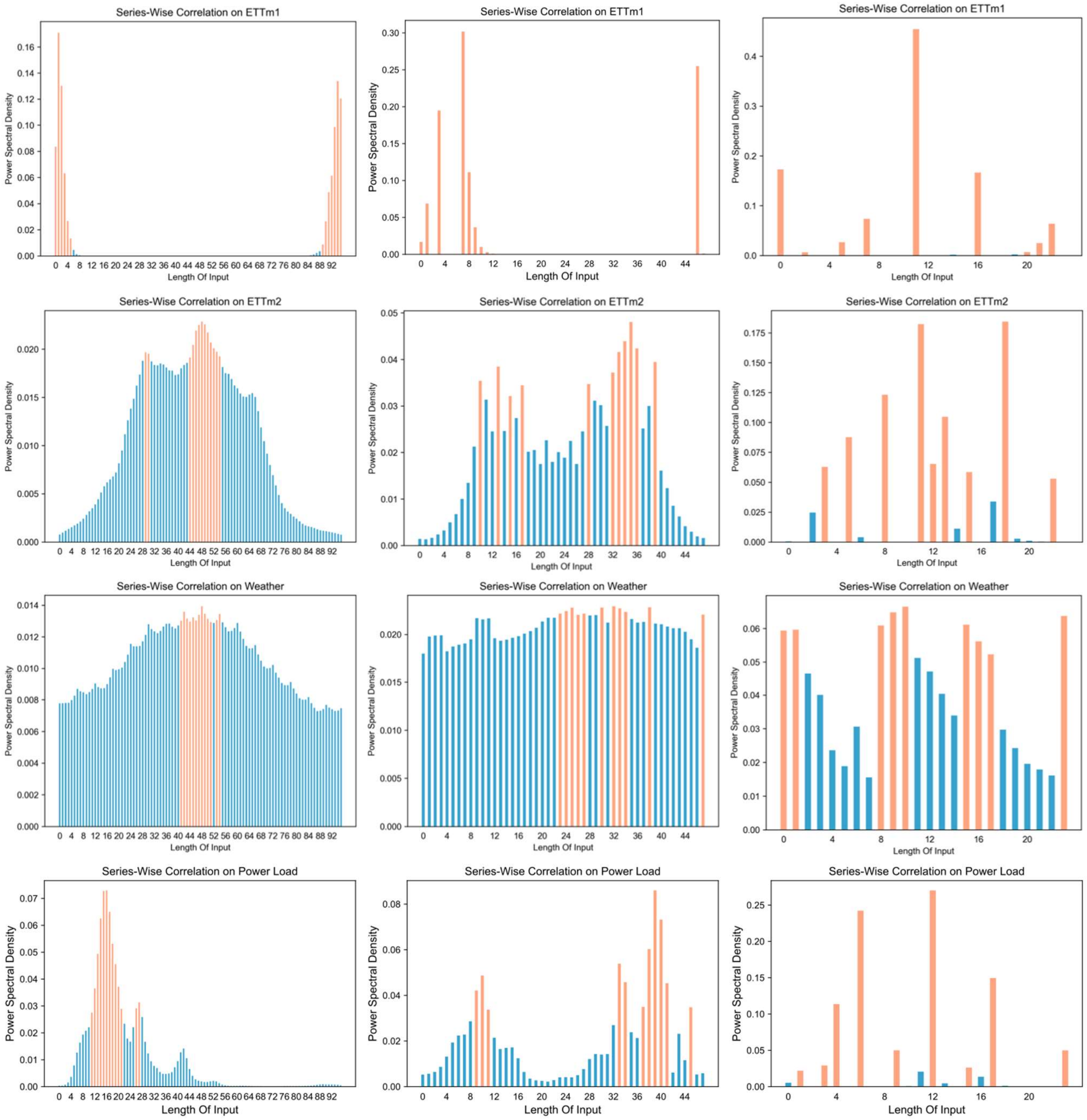

Figure 13. Power spectral density of the global energy selection module at three granularities in the four datasets.

4.1.2. Localized distribution analysis

As shown in Figure 14, the localized distribution validation experiments were performed on the ETTm2, Weather, and PowerLoad datasets. The time-frequency information matrix W of $small\_scales$ and big_scales generated using the Morlet wavelet transform in the local energy selection module was found to have a more obvious stratification. In this experiment, the big_scales and $small\_scales$ scale parameter c were set to 3. big_scales displayed the characteristic of wide distribution, while $small\_scales$ displayed the characteristic of multi-point distribution and localization. On the ETTm2 dataset, A1 and A2 are localized distributions with obvious volatility, which can be well captured by $small\_scales$ in the time-frequency plots. On the Weather dataset, A1, A2, and A3 are localized distributions with obvious volatility, which can be well captured by $small\_scales$ in the time-frequency plots. Because there are no localized distributions with significant volatility, the distributions are concentrated on big_scales for the PowerLoad dataset. Since the Morlet wavelet transform used in the local energy selection module has the characteristic of time-frequency localization, when the appropriate ratio of big_scales and $small\_scales$ is set, the use of asymmetric convolution can be targeted to learn the local mutation characteristics of the sequence and improve the model's ability to fit the peaks and valleys of the sequence, which verifies the effectiveness of the local energy selection module.

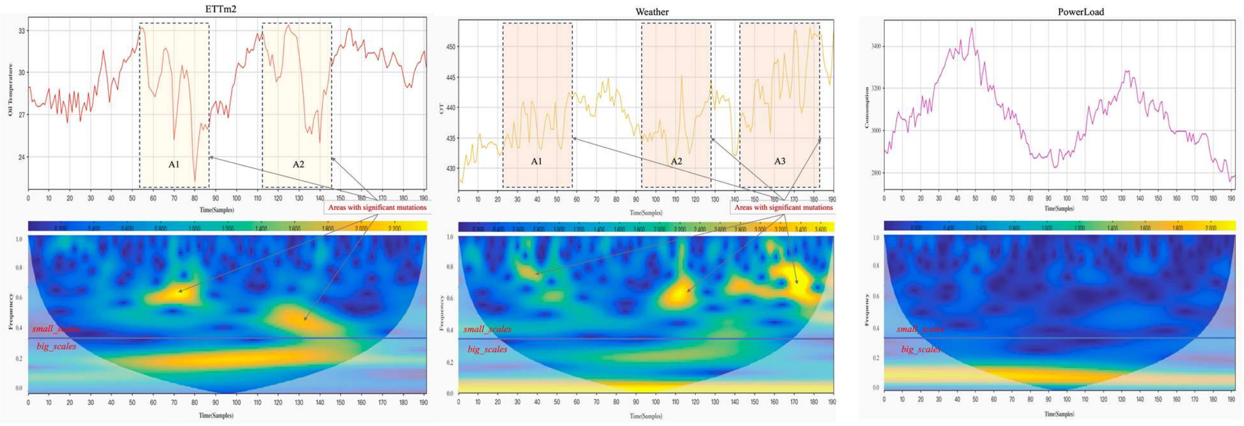

Figure 14. Localized distribution analysis plots on ETTm2, Weather and PowerLoad datasets.

4.2. "Priori knowledge" guidance module distribution analysis

The power spectrum energy magnitude was extracted from four datasets for the two cases of no-knowledge guidance and knowledge guidance, thus verifying the impact of the "priori knowledge" guidance module on the model. Since there was no explicit "priori knowledge" in the ETTm1, ETTm2, and Weather datasets, this analytical experiment was simulated using a random covariate as the "priori knowledge". As shown in Figure 15, the power spectral energy distribution of the four data sets with knowledge guidance is significantly different and more widely distributed than that without knowledge guidance. Due to the directed learning of sequence features in the knowledge-guided branch of the "priori knowledge" guidance module, the model is able to capture possible unconventional changes in the future, thus improving its stability.

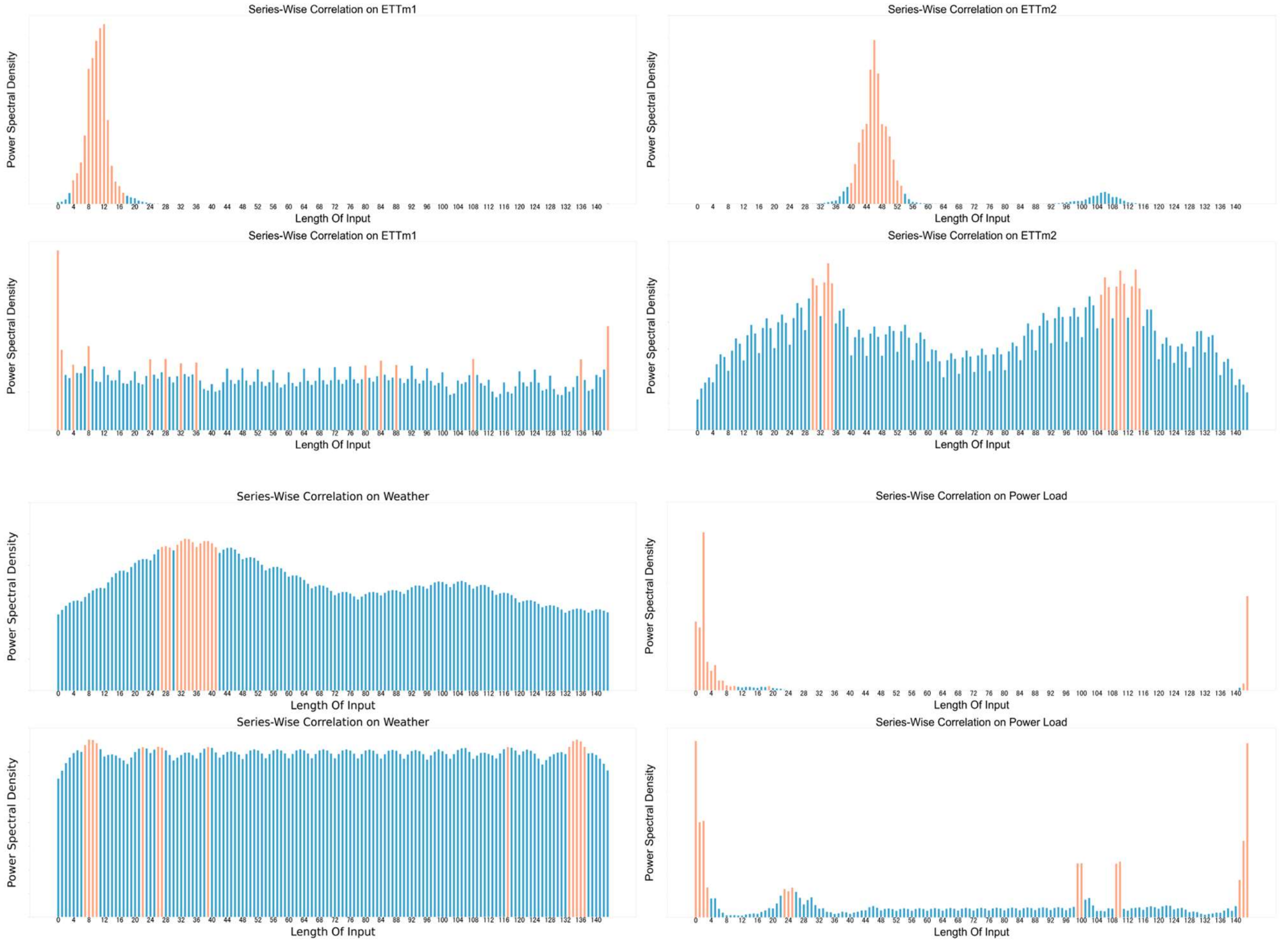

Figure 15. Comparison of power spectral density with and without the "prior knowledge" guidance module.

4.3. Analytical experiments with sequence denoising decomposition modules

Figure 16 illustrates the visual sequence decomposition process performed on the Weather dataset to validate the necessity of sequence denoising decomposition modules. The four plots in the figure represent the initial decomposition and the output of the three sequence denoising decomposition modules. As can be seen in Figure 16, the model is able to gradually aggregate, refine the trend and seasonal terms of the sequence, and effectively remove the noise components from the sequence. In addition, the module can effectively suppress abnormal fluctuations in the data and shows remarkable results in long-term sequence prediction tasks. This remarkable effect is mainly attributed to the fact that the sequence decomposition architecture, which consists of multiple sequence denoising decomposition modules of TFDNet, is able to fit the trend and seasonal terms asymptotically; at the same time, the bilateral filtering layer in the sequence denoising decomposition module is able to effectively filter the residual noise components in the sequence. It should be noted that all data are normalized with a bias of + 0.02 to the original series and trend terms.

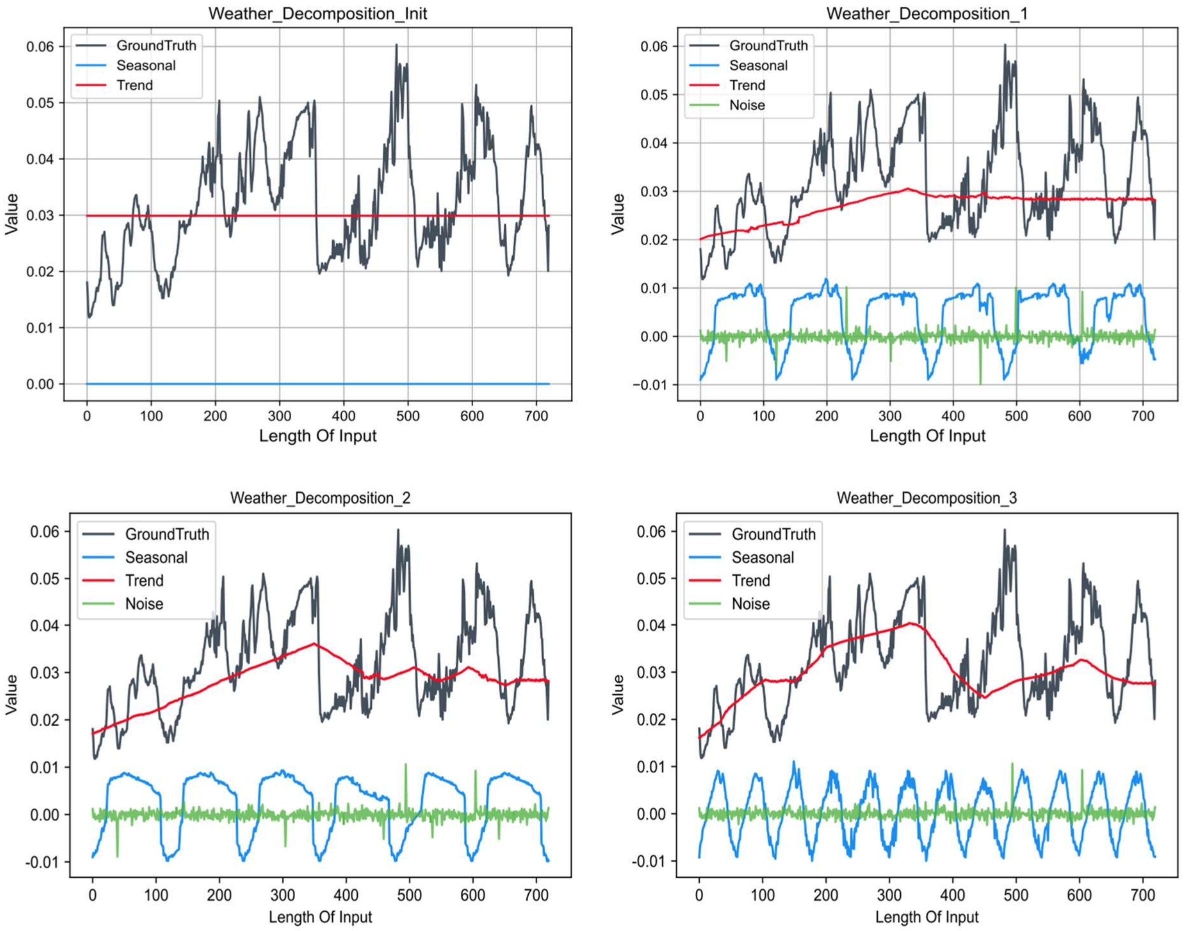

Figure 16. Asymptotic decomposition of trend term, seasonal term, and noise residual term for three sequence denoising decomposition modules in decoder.

4.4. Predicted output distribution analysis.

As shown in Table 4, the Kolmogorov-Smirnov test was utilized to quantitatively evaluate the similarity in the distribution of input and predicted output sequences for the similar sequence decomposition models on the ETTm1 and PowerLoad datasets. The experiments were conducted with an input sequence length of 96 and different prediction output lengths of {96, 192, 336, 720}. In both datasets, the P-value was set to 0.01, and the original assumption was that the two series were from the same distribution.

By comparing the $P-value$, when the length of the predicted output result is 96, the $P-value$ of the four models is greater than 0.01, which indicates that the predicted output sequence comes from the same distribution as the input sequence with a higher probability, which is due to the sequence decomposition mechanism adopted by all four models. However, as the prediction output window increases, the $P-value$ values of all four models show different magnitudes of decrease. Among them, the EEMD-DARNN model shows a larger decrease, mainly due to the poor ability of EEMD-DARNN to capture the characteristics of the "inflection points" region.

Compared with Autoformer, FEDformer, and EEMD-DARNN, the count number of this paper's method is 5, which is larger than Autoformer and FEDformer's 1 and 2, respectively. Compared with the $P - value$ of the real output, the model proposed in this paper maintains a higher $P - value$ while being closer to the $P - value$ of the real distribution. This implies that the model is better able to capture unanticipated changes in the future, thus validating the effectiveness of the "priori knowledge" guidance module.

Table 4. The Kolmogrov-Smirnov test [23] for the input and predicted sequences. Larger values represent a higher probability that the two series are from the same distribution and a lower probability of rejecting the original hypothesis.

Methods		Autoformer	FEDformer	EEMD-DARNN	TFDNet	True
Metric		$P - value$	$P - value$	$P - value$	$P - value$	$P - value$
ETTm1	96	0.029	**0.050**	0.044	0.042	0.031
	192	0.016	0.015	0.014	**0.027**	0.024
	336	0.009	0.012	0.008	**0.018**	0.017
	720	0.006	**0.010**	0.003	0.009	0.005
PowerLoad	96	0.023	0.028	0.039	**0.046**	0.034
	192	**0.020**	0.019	0.018	0.018	0.015
	336	0.007	0.010	0.006	**0.012**	0.013
	720	0.003	0.008	0.003	**0.009**	0.011
Count		1	2	0	5	NA

4.5. *Parametric analysis*

The method for selecting the number of multiple granularities is as follows. The effect on the model accuracy was analyzed by varying the number of global energy selection modules at different granularities of the sequence. As shown in Figure 17, except for Size_1, the MSE values of Size_3, Size_5, and Size_7 gradually converge with an increase in the number of epochs, though the MSE of Size_3 converges faster compared to the other two. The reason is that the multi-granularity learning by the branching structure of the global energy selection module does capture the long, medium, and short features of the sequence, which is a great improvement for the model performance. However, too much branching structure may aggravate the efficiency of the model training.

The size of the hyperparameter c determines the number of ArgTopk() selection cycle subsequences and the ratio of big_scales to $small\_scales$ allocation. In this paper, experiments were conducted on the PowerLoad dataset with an input sequence length of 96 and a predicted sequence length of 96. The effect of different c on the model accuracy was verified by setting c to 1, 2, 3, and 4. As shown in Figure 18, except for c = 1, the MSE of c = 2, c = 3, and c = 4 gradually converge with an increase in the number of training rounds, though the curve for c = 3 converges faster. The analytical analysis of the choice of hyperparameter c shows that when c =3, the model is able to reduce the computational complexity while reducing the MSE of the model by better capturing the timing features using the global energy selection module and the local energy selection module.

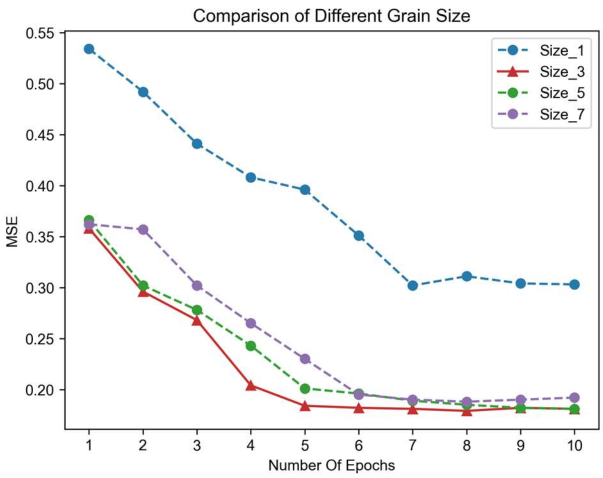

Figure 17. MSE values under different granularity energy selection models. Analytical experiments on the PowerLoad dataset with an input sequence length of 96 and a predicted sequence length of 96.

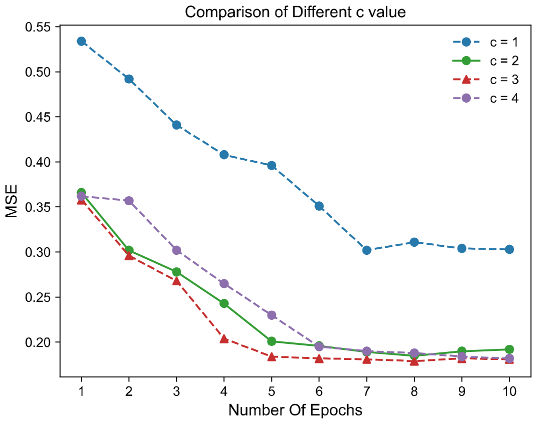

Figure 18. MSE values in the time-frequency energy selection module for different c.

The method for selecting the number of covariates is as follows. In order to select appropriate and relevant covariate data for continuous and discrete variables, three sets of parametric analysis experiments in the PowerLoad dataset were conducted to detect the effect of different numbers of covariates on the model performance. As shown in Figure 19, when the number of discrete covariates is fixed, the MSE value is at its minimum when the first six continuous variables with the strongest correlation are selected. When fixing the number of continuous covariates and selecting the first most relevant discrete variable, the MSE value is the minimum. When the combination of 6 continuous and 1 discrete variable is selected, the MSE value is the minimum, which is better than the two extreme combinations of contimuous_0_discrete_0 and continuous_8_discrete_2. Through these three covariate parameter analysis experiments, it can be verified that the preprocessing step of screening covariates can not only effectively improve the computational efficiency of the model, but also improve the accuracy of the model to a certain extent and reduce the negative impact of some covariates on the model.

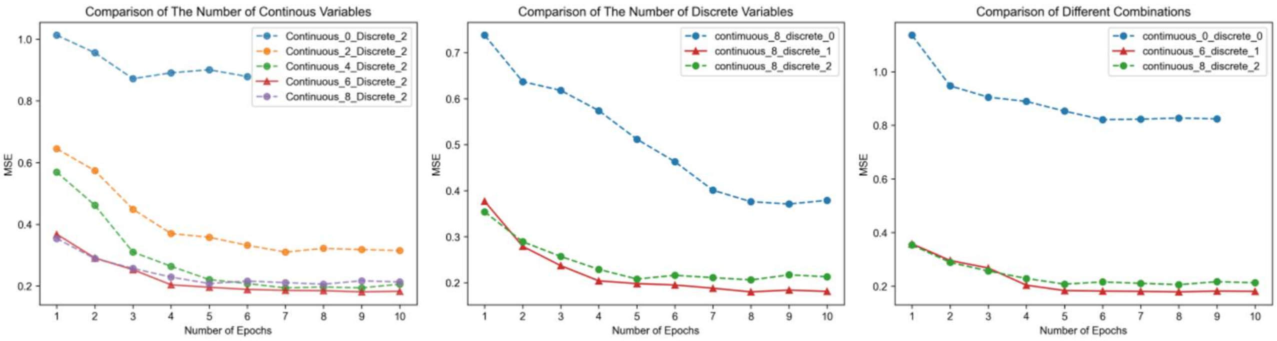

Figure 19. Analytical experiments with combinations of covariates.

5. Conclusions

In this paper, a sequence decomposition model based on power spectral density and the Morlet continuous wavelet transform is proposed for the task of ultra-short-term time series prediction in the context of power load. The model digs out the time-frequency domain feature relationship of the same sequence in different scales from the perspective of the time-frequency domain and introduces a time-frequency energy selection module, a "priori knowledge" guidance module, a sequence denoising decomposition module, and a probabilistic load prediction output to perform deep mining and the expression of sequences in order to solve the problems of the current ultrashort-term time series prediction, such as poor accuracy, difficulty in capturing local mutation features, etc. Through comparative experiments on four different datasets, the advanced nature of the model was validated. In addition, the validity of the three modules was further verified through ablation experiments, module analysis visualization experiments, and a parameter analysis. The limitation of the model is the manual selection of the hyperparameter scale factor c. Too high a value of c increases the computational complexity of the model, thus leading to a reduction in prediction efficiency; too low a value of c weakens the ability of the energy selection module to capture sequence features, thus leading to a reduction in the predictive ability of the model. Future research will be devoted to automating the optimal selection of hyperparameters for the model while maintaining the model's accuracy and efficiency (e.g., using Automated Machine Learning Library (AutoML)-based and algorithmic-based hyperparameter optimization methods to improve the utility and generality of the model).

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors gratefully acknowledge the support of the following foundations: The National Nature Science Foundation of China (No. U21A20469), Central Funds to Guide Local Science and Technology Development (No. YDZJSX2022C004).

The authors gratefully acknowledge that the data support of Shanxi Heli Innovation Technology

Holding Company Ltd. from China is greatly appreciated.

Conflict of interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and publication of this article.

Reference

1. A. Haque, S. Rahman, Short-term electrical load forecasting through heuristic configuration of regularized deep neural network, *Appl. Soft Comput.*, **122** (2022), 108877. https://doi.org/10.1016/j.asoc.2022.108877
2. J. Ma, M. Yang, X. Han, Z. Li, Ultra-short-term wind generation forecast based on multivariate empirical dynamic modeling, *IEEE Trans. Ind. Appl.*, **54** (2017), 1029–1038. https://doi.org/10.1109/TIA.2017.2782207
3. Y. Dai, X. Yang, M. Leng, Forecasting power load: A hybrid forecasting method with intelligent data processing and optimized artificial intelligence, *Technol. Forecast. Soc. Change*, **182** (2022), 121858. https://doi.org/10.1109/TIA.2017.2782207
4. R. Ospina, A. Gondim, V. Leiva, C Castro, An overview of forecast analysis with ARIMA models during the COVID-19 pandemic: Methodology and case study in Brazil, *Mathematics*, **11** (2023), 3069. https://doi.org/10.3390/math11143069
5. F. Yuan, Z. Zhang, Z. Fang, An effective CNN and Transformer complementary network for medical image segmentation, *Pattern Recognit.*, **136** (2023), 109228. https://doi.org/10.1016/j.patcog.2022.109228
6. M. Saraswat, Srishti, Leveraging genre classification with RNN for Book recommendation, *Int. J. Inf. Technol.*, **14** (2022), 3751–3756. https://doi.org/10.1007/s41870-022-00937-6
7. D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, DeepAR: Probabilistic forecasting with autoregressive recurrent networks, *Int. J. Forecast.*, **36** (2020), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001
8. W. Zha, Y. Liu, Y. Wan, R. Luo, D. Li, S. Yang, et al., Forecasting monthly gas field production based on the CNN-LSTM model, *Energy*, **260** (2022), 124889. https://doi.org/10.1016/j.energy.2022.124889
9. Y. Wang, L. Rui, J. Ma, A short-term residential load forecasting scheme based on the multiple correlation-temporal graph neural networks, *Appl. Soft Comput.*, **146** (2023), 110629. https://doi.org/10.1016/j.asoc.2023.110629
10. B. Tang, D. S. Matteson, Probabilistic transformer for time series analysis, *Adv. Neural Inf. Process. Syst.*, **34** (2021), 23592–23608.
11. S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y. X. Wang, et al., Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, in *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.
12. N. Kitaev, L. Kaiser, A. Levskaya, Reformer: The efficient transformer, preprint, arXiv:200104451.

13. H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, et al., Informer: Beyond efficient transformer for long sequence time-series forecasting, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2021), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325
14. H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, in *Advances in Neural Information Processing Systems*, (2021), 22419–22430.
15. T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, in *Proceedings of the 39th International Conference on Machine Learning*, (2022), 27268–27286.
16. H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, M. Long, TimesNet: Temporal 2D-variation modeling for general time series analysis, preprint, arXiv:221002186.
17. F. Yang, X. Fu, Q. Yang, Z. Chu, Decomposition strategy and attention-based long short-term memory network for multi-step ultra-short-term agricultural power load forecasting, *Expert Syst. Appl.*, **238** (2024), 122226. https://doi.org/10.1016/j.eswa.2023.122226
18. T. Donoghue, M. Haller, E. J. Peterson, P. Varma, P. Sebastian, R. Gao, et al., Parameterizing neural power spectra into periodic and aperiodic components, *Nat. Neurosci.*, **23** (2020), 1655–1665. https://doi.org/10.1038/s41593-020-00744-x
19. H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, et al., Unet 3+: A full-scale connected unet for medical image segmentation, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, 1055–1059. https://doi.org/10.1109/ICASSP40776.2020.9053405
20. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 2117–2125.
21. B. H. Chen, Y. S. Tseng, J. L. Yin, Gaussian-adaptive bilateral filter, *IEEE Signal Process. Lett.*, **27** (2020), 1670–1674. https://doi.org/10.1109/LSP.2020.3024990
22. Z. Ni, C. Zhang, M. Karlsson, S. Gong, A study of deep learning-based multi-horizon building energy forecasting, *Energy Build.*, **203** (2024), 113810. https://doi.org/10.1016/j.enbuild.2023.113810
23. J. R. Lanzante, Testing for differences between two distributions in the presence of serial correlation using the Kolmogorov–Smirnov and Kuiper's tests, *Int. J. Climatol.*, **41** (2021), 6314–6323. https://doi.org/10.1002/joc.7196