Learning Hierarchical Representations in Temporal and Frequency Domains for Time Series Forecasting

Zhipeng Zhang[1], Yiqun Zhang[1](✉), An Zeng[1], Dan Pan[2], and Xiaobo Zhang[1]

[1] Guangdong University of Technology, Guangzhou, China
2112105278@mail2.gdut.edu.cn, {yqzhang,zengan,zxb_leng}@gdut.edu.cn
[2] Guangdong Polytechnic Normal University, Guangzhou, China
pandan@gpnu.edu.cn

Abstract. Long-term time series forecasting is a critical task in many domains, including finance, healthcare, and weather forecasting. While Transformer-based models have made significant progress in time series forecasting, their high computational complexity often leads to compromises in model design, limiting the full utilization of temporal information. To address this issue, we propose a novel hierarchical decomposition framework that disentangles latent temporal variation patterns. Specifically, we decompose time series into trend and seasonal modes and further decompose seasonal temporal changes into coarse- and fine-grained states to capture different features of temporal sequences at different granularities. We use linear layers to embed local information for capturing fine-grained temporal changes and Fourier-domain attention to capture multi-periodic seasonal patterns to extract coarse-grained temporal dependency information. This forms a time series forecasting modeling from fine to coarse, and from local to global. Extensive experimental evaluation demonstrates that the proposed approach outperforms state-of-the-art methods on real-world benchmark datasets.

Keywords: Time series forecasting · hierarchical decomposition · fourier-domain attention · deep learning · supervised learning

1 Introduction

Supervised learning is one of the most popular tasks in pattern recognition, which includes the conventional classification [1], challenging concept-drift adaptation [2], and many cutting-edge applications [3]. Time series forecasting, also being an important pattern recognition task, has been widely applied to time series datasets in various domains, e.g., finance, healthcare, weather, etc. These applications rely on historical time series to make predictions for future time series, assisting decision-making and planning. As distribution of time series data is easily influenced by missing values [4], concept-drifts [5], and unforseen external

Q. Liu et al. (Eds.): PRCV 2023, LNCS 14433, pp. 91–103, 2024.
https://doi.org/10.1007/978-981-99-8546-3_8

factors, the patterns become complex, and are usually composed of seasonality and trend components. An intuitive example is the overall growth (trend) and 24-h changes (seasonal) in electricity consumption. Most existing methods directly consider the stack of these two patterns as a whole for time series prediction, which cannot capture the potential temporal changes, and thus results in unsatisfactory prediction results, especially for long-term forecasting.

Although deep learning methods, e.g., convolutional neural networks (CNN) [6], recurrent neural networks (RNN) [7], and temporal convolutional networks (TCN) [8], have achieved better results compared to traditional methods, they possess fixed receptive fields, constraining their ability to capture long-term patterns. Recently, transformer models [9] have achieved tremendous success, owing to their ability to capture long-range dependencies in sequence data. Therefore, an increasing number of transformer-based models have been proposed demonstrating remarkable performance in long-term time series forecasting. Since their high computational cost becomes bottlenecks, the current research efforts, e.g., Logtrans [10], Reformer [11], Triformer [12] and Informer [13], mainly focus on improving the efficiency. However, these linear complexity models may limit the performance as they do not fully exploit the information and lack built-in prior structures. Moreover, they are incapable of learning necessary periodic and seasonal dependencies in complex and diverse time patterns.

To tackle the issues mentioned above, we focus on analyzing and utilizing the inherent patterns of variation in time series data. Figure 1(a) shows that the original input sequence changes without specific regularity. If we learn representations directly from raw sequences, the learned representations may not generalize well. If the observed sequence is composed of the trend and seasonality modules shown in Fig. 1(b) and Fig. 1(c), and we know that the distribution of the seasonality module changes due to different periods leading to different fine-grained and coarse-grained time changes, we can still make reasonable predictions based on the invariant trend module. Therefore, it is promising to deduce a modular architecture for modeling complex temporal variations through a multi-stage decomposition approach.

In this paper, we aim to tackle the complexity and diversity of temporal patterns in multivariate long-term time series data by decomposing the important factors and variables in multiple stages. Accordingly, a novel LHRTF architecture has been designed to model the dependency relationships across the entire temporal range, from fine-grained to coarse-grained time scales. It is based on the multi-level decomposition framework, which combines the trend extraction ability of multi-layer perceptron and the seasonal information expression ability of frequency domain attention to form a model to predict time series with complex and diverse time patterns. Experimental results demonstrate that the proposed model can effectively capture the complex temporal patterns and underlying factors in multi-level time series data, and thus achieves more accurate long-term time series forecasting. The main contributions are summarized below:

- This work addresses the intricacy and variability of temporal patterns in multi-dimensional long-term time series data. By decomposing time series into

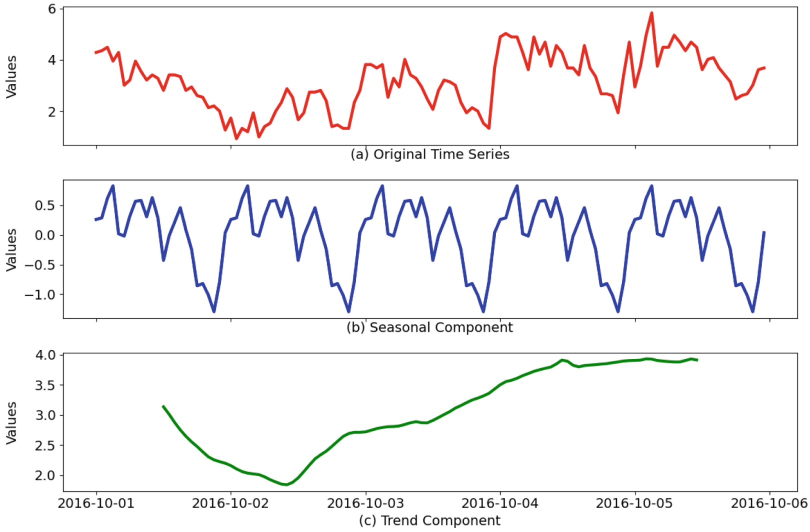

Fig. 1. Seasonal (b) and trend (c) components of a time series (a).

multiple stages, significant information can be comprehensively extracted. As a result, inherent multiple periodic and seasonal time patterns in the seasonal sequences can be captured.

- A new mechanism has been designed to extract and integrate seasonal temporal dependencies, which involves capturing fine-grained temporal dependencies in the time domain and coarse-grained temporal dependencies in the frequency domain, which forms an information extraction and integration process of seasonal temporal dependencies from local to global.
- Extensive experiments have been conducted on real-world multivariate time series datasets from various domains, including energy, economy, weather, and disease. The empirical results show that the prediction performance of the proposed model is very competitive, reducing the average MSE and MAE by 14.9% and 13.1%, respectively, compared to the state-of-the-art methods.

2 Related Work

2.1 Convolutional and Transformer Models

Deep learning has achieved remarkable success in the field of time series modeling, and various deep models have been proposed for time series modeling. Typical models include RNN and its variants [14] adopting RNN as a modeling framework and simulating the distribution of future sequences. LSTNet [7] further combines RNN and CNN to better capture the long- and short-term dependencies of time series. MTCN uses multiple convolutional layers to slide through time, thus obtaining the changing temporal relationship between multiple variables. Recently, the transformer model based on the self-attention mechanism has been utilized to better learn the long-term dependencies. For instance, Forecaster [15] introduces GCN and transformer structures, and utilizes a graph structure to obtain adjacent node information, thus capturing the long-term spatio-temporal correlation better. LogTrans [10] combines CNN and transformer structures and applies convolution to enhance the model's local attention ability. Spare Transformer [16] improves computational efficiency by reducing the number of fully connected layers. Moreover, Informer [13] proposes the ProSparse Self-Attention mechanism to enhance the model's expressive power and incorporates non-autoregressive encoding to generate future sequences.

2.2 Fourier Transform and Decomposition Models

In recent years, deep learning models based on Fourier Transform have received widespread attention in the field of time series forecasting [17,18]. Fourier Transform enables the acquisition of rich periodic information in the frequency domain, and complex temporal patterns typically contain multiple periodicities. For example, Autoformer [19] utilizes fast Fourier transform to design an autocorrelation mechanism that better aggregates information. FEDformer [20] introduces a frequency domain information enhancement mechanism that effectively captures periodic information in long-term time series, while ESTformer [21] combines exponential smoothing and transformer models to simultaneously process various time series components, including trends and seasonality. With the development of time series forecasting, there has been an increasing awareness of the need to focus on and utilize complex temporal patterns for modeling. DeepFS [22] decomposes the sequence and extracts the features of each component into an attention network for interaction and fusion, enabling the extraction of temporal features. Furthermore, MICN [23] is based on a decomposition model of convolutional neural networks and performs convolution operations in the time domain to extract multi-scale temporal dependencies of seasonal time series.

3 Proposed Approach

In this section, we first introduce our problem definition and then describe our LHRTF framework that can capture both seasonal and trend dependencies. The LHRTF overall framework is illustrated in Fig. 2, where we propose a modular architecture to capture the time dependencies of trends and seasonality separately. In Sect. 3.1, we present our multi-level decomposition strategy. In Sect. 3.2, we provide detailed information on the trend prediction module, and in Sect. 3.3, we specifically describe our approach to modeling seasonality. We begin by presenting some basic definitions to facilitate the research, discussion, and analysis in subsequent sections. The definition of the long-term sequence forecasting problem is as follows: Given a historical input sequence $X^{(h)} = [x_1, x_2, x_3, \ldots, x_n] \in \mathbb{R}^{n \times d}$ with a length of the historical sequence is n

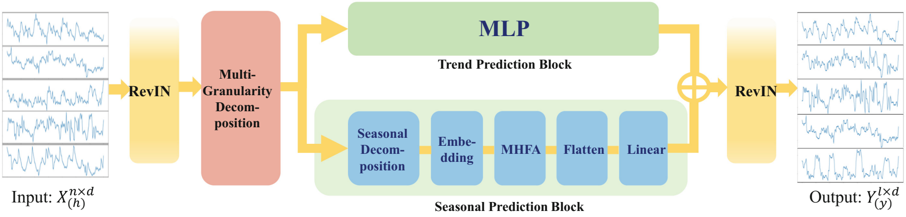
Input: $X_{(h)}^{n\times d}$
RevIN
Multi-Granularity Decom-position
MLP
Trend Prediction Block
Seasonal Decom-position
Embe-dding
MHFA
Flatten
Linear
Seasonal Prediction Block
RevIN
Output: $Y_{(y)}^{l\times d}$

Fig. 2. Overall architecture of LHRTF.

and dimensionality of d, the forecasting task is to forecast the values for the next l steps: $X^{(y)} = [x_{n+1}, x_{n+2}, x_{n+3}, \ldots, x_{n+l}] \in \mathbb{R}^{l\times d}$ (i.e., learning the mapping function as $f : H^{n\times d} \mapsto Y^{l\times d}$).

Remark 1. To address the non-stationarity problem of time series data, recent studies [24] have proposed instance normalization methods to reduce the influence of distribution shift by normalizing time series instances using mean and standard deviation. Similarly, we normalize the input time series data by reversible instance normalization (RevIN) before forecasting and finally add back the mean and deviation to the predicted sequence.

3.1 Time Series Hierarchical Decomposition

We propose a multi-level decomposition framework to address the challenge of modeling the complex patterns of time series forecasting and to obtain more latent factors for improved accuracy. Firstly, we performed a first-level decomposition by utilizing multiple averaging filters of varying scales to separate the trend and seasonal information of different patterns. The trend component was obtained by averaging the resulting patterns, while the seasonal component was obtained by subtracting the trend from the original sequence. Separating the seasonal and trend components allows for improved transferability and generalization of non-stationary time series data, which is critical for effectively modeling non-stationary time series data, the process is:

$$X_{trend} = \frac{\sum_{i=1}^{n} f\left(x_i\right)}{n}, \quad X_{season} = X - X_{trend} \tag{1}$$

where $X_{trend}, X_{season} \in \mathbb{R}^{n\times d}$, and $f\left(x_i\right)$ is an average filter with a certain scale.

To facilitate better modeling and forecasting of the seasonal component, we performed a second decomposition on the seasonal time series obtained from the first-level decomposition, resulting in multiple sub-sequences. This approach enables the extraction of deeper levels of periodic and seasonal information, the process is:

$$\sum_{i=1}^{m} X_{s(i)} = X_s^{n\times d} \tag{2}$$

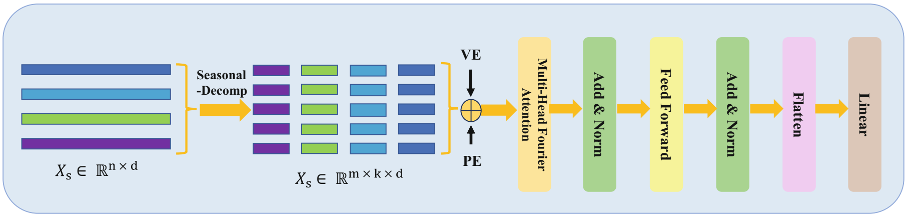
Seasonal
-Decomp
$X_s \in \mathbb{R}^{n\times d}$
$X_s \in \mathbb{R}^{m\times k\times d}$
VE
PE
Multi-Head Fourier Attention
Add & Norm
Feed Forward
Add & Norm
Flatten
Linear

Fig. 3. Seasonal Forecasting Module.

where $X_s^{(i)} \in \mathbb{R}^{\frac{n}{m}\times d}$, $X_s^{(i)}$ represents the i-th subsequence, and m represents the number of subsequences.

After the second seasonal decomposition, we can better capture the different fluctuations or frequencies in seasonal data. For example, for seasonal data with one peak per month in a year, we can decompose each month into a subsequence to better understand the seasonal fluctuations of each month. During prediction, we can utilize the seasonal pattern of each month instead of simply using the seasonal pattern of the entire year. Moreover, decomposing the seasonal component into multiple identical subsequences can more precisely characterize the seasonality, which improves the accuracy of the model in predicting seasonal changes, which is crucial for the accurate modeling of complex time series patterns.

3.2 Trend Forecasting Module

We propose to use a multi-layer perceptron (MLP) for trend forecasting. By mapping the decomposed trend sequence through an MLP, we obtain the predicted trend sequence, which can be written as:

$$Z_{trend} = MLP(X_{trend}) \tag{3}$$

where $Z_{trend} \in \mathbb{R}^{l\times d}$ represents the trend part of the predicted sequence after passing through the multi-layer perceptron network.

3.3 Seasonal Forecasting Module

The seasonal forecasting module, as depicted in Fig. 3, is designed to predict complex and diverse seasonal time variations. We leverage the advantages of the Fourier transform in obtaining a spectrum of rich multi-periodicity information and apply Fourier attention to the seasonal module to capture periodic changes at different time scales, which can be expressed as:

$$\begin{gathered} X_s = X_s^{d\times m\times k}, \quad X_s^{embed} = Embedding(X_s^{d\times m\times k}), \quad \tilde{X_s} = MHFA(X^{embed}) \\ Z_{season} = Linear(Flatten(\tilde{X_s})), \quad Y_{pred} = Z_{season} + Z_{trend} \end{gathered} \tag{4}$$

where k is the subsequence length and $Y_{pred} \in \mathbb{R}^{l \times d}$ is the prediction. The Fourier transform converts time-domain signals to frequency-domain signals by decomposing them into a combination of sinusoidal and cosinusoidal waves with varying frequencies. As a result, it excels at capturing time series with periodic changes on different time scales. Moreover, frequency-domain attention is with a smaller scale of parameters, which ensures efficient training and testing.

Embedding. We project the length of each subsequence resulting from the second decomposition into a K-dimensional hidden space using a linear layer in the time domain. This allows us to capture fine-grained temporal dependencies within each subsequence while incorporating learnable position embedding to model the temporal positions of the time series data, the process is:

$$X_d^{(i)} = W_{linear} X_s^{(i)} + W_{pos}, \quad X_s^{embed} = W_{linear}(\sum_{i=1}^{m} X_s^{(i)}) + W_{pos} \tag{5}$$

where W_{pos} and W_{linear} represent encoding of position and value, respectively.

MHFA and Flatten. We first provide Remark 2 below.

Remark 2. The convolution theorem states that the Fourier transform of the cyclic convolution of two signals is equal to the point-wise multiplication of their Fourier transforms in the frequency domain. Given a signal $x[n]$ and a filter $h[n]$, the convolution theorem can be expressed as follows:

$$\mathfrak{F}\left[f_1(t) * f_2(t)\right] = \mathfrak{F}_1(w) \bullet \mathfrak{F}_2(w) \tag{6}$$

where $f_1(t) \leftrightarrow \mathfrak{F}_1(w)$, $\mathfrak{F}$ for Fourier transform, and $*$ for convolution operation.

According to Remark 2, we know that the pointwise product of the spectra of two sequences is equivalent to their circular convolution in the time domain. Multiplying sequences with larger receptive fields reflects more global features (such as periodicity) and requires less computational cost. Although transformer-based models typically use self-attention mechanisms to capture long-term dependencies in time series, they do not consider the local dependencies among subsequences within the sequence during modeling, which may limit the utilization of local structural information in the sequence.

In contrast, we input each embedded subsequence of every dimension into the multi-head Fourier attention mechanism in sequence, which calculates the Fourier attention mechanism in the frequency domain to capture the global seasonal sequence, forming a local-to-global information extraction and integration for temporal dependencies by:

$$\mathfrak{F}\left(Q_h^{(i)}, Q_K^{(i)}, Q_V^{(i)}\right) = \mathfrak{F}\left(\left(X_d^{(i)}\right)\left(W_h^Q, W_h^K, W_h^V\right)\right) \tag{7}$$

and

$$\left(O_h^{(i)}\right)^T = FA\left(Q_h^{(i)}, K_h^{(i)}, V_h^{(i)}\right) = \mathfrak{F}^{-1}\left(Softmax\left(\frac{\mathfrak{F}\left(Q_h^{(i)}\right)\mathfrak{F}\left(K_h^{(i)}\right)^T}{\sqrt{d_k}}\right)\right)\mathfrak{F}V_h^{(i)} \tag{8}$$

where $W_h^Q, W_h^K \in \mathbb{R}^{K\times d_k}$, $W_h^V \in \mathbb{R}^{K\times K}$, and FA indicates the Fourier Attention operation. Accordingly, attention output $O_h^{(i)} \in \mathbb{R}^{K\times d_k}$ can be obtained by reducing the number of input tokens from n to m, and a linear complexity can be achieved for the frequency-domain attention mechanism. $O_h^{(i)}$ is then processed by a norm layer and a FeedForward Network (FFN) with residual connections by:

$$\tilde{X_s}^{(i)} = FFN\left(\left(O_h^{(i)}\right)^T + X_d^{(i)}\right). \tag{9}$$

Finally, the output $\tilde{X_s} \in \mathbb{R}^{K\times m\times d}$ is flattened and then mapped through a linear layer to generate the final seasonal forecasting sequence $Z_{trend} \in \mathbb{R}^{l\times d}$.

4 Experiments

4.1 Dataset

We have extensively experimented with seven real-world publicly available benchmark datasets. The ETT [13] dataset includes four subsets, namely, two hourly datasets (ETTh1 and ETTh2) and two minutely datasets (ETTm1 and ETTm2), which record six power load features and the target variable "oil temperature" collected from power transformers. The Electricity dataset [25] records the hourly electricity consumption of 321 users from 2012 to 2014. The Weather dataset [26] contains 21 weather indicators, such as humidity, air pressure, and rainfall, recorded every 10 min from July 2020 to July 2021, from nearly 1600 locations in the United States. The ILI dataset [27] records the weekly patient data of influenza-like illness (ILI) from the Centers for Disease Control and Prevention in the United States from 2002 to 2021, describing the ratio of observed ILI patients to the total number of patients. Following the same standard protocol as before, we split all the forecasting datasets into training, validation, and testing sets with ratios of 6:2:2 for ETT datasets and 7:1:2 for other datasets.

4.2 Baselines and Setup

The proposed method is compared with seven baseline methods, including five transformer-based models: LogTrans [10], Pyraformer [28], Informer [13], Autoformer [19], and FEDformer [20], and two non-transformer models: SCINet [29] and MICN [23]. All baselines follow the same evaluation protocol to ensure a fair comparison. The forecasting horizon for the ILI dataset is set to $T \in \{24, 36, 48, 60\}$, and for other datasets, it is set to $T \in \{96, 192, 336, 720\}$, consistent with the settings in the original papers. We collect the results of these time series forecasting baselines from their respective papers. For the state-of-the-art non-transformer MICN and the transformer-based FEDformer, we compare with their improved versions, i.e., MICN-regre and FEDformer-f, respectively.

4.3 Implement Details and Evaluation Metrics

Our approach employs the Adam optimizer. Our method is trained with $L2$ loss. The batch size of the ETT dataset is set to 128, and the batch size of other datasets is set to 16 and 32 respectively. Our model contains 1 frequency-domain attention with the number of heads $H = 8$, and the latent space dimension $D = 512$. All experiments use dropout with a probability of 0.05. We adopt early stopping by terminating training if the MAE on the validation set does not decrease for three consecutive rounds. The training process is stopped prematurely within 30 epochs. MSE and MAE are used as evaluation metrics for all benchmarks. All models are implemented in PyTorch and trained and tested on a single Nvidia GeForce RTX 3090 GPU.

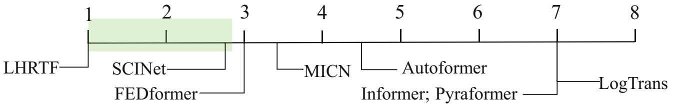

Fig. 4. Significance test using Bonferroni-Dunn test at confidence interval 95% (i.e. $\alpha = 0.05$). The light green region stands for the right side of the critical difference interval. The proposed method performs significantly better than the comparison methods that rank outside this interval. (Color figure online)

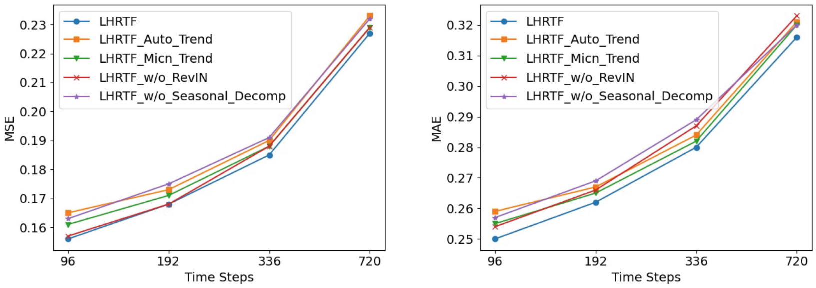

Fig. 5. Ablation results in terms of(a) MSE and (b) MAE on Electricity dataset.

4.4 Main Results

To ensure a fair comparison, we followed the same evaluation protocol where the historical range length was set to 36 for influenza-like ILI and 96 for

Table 1. Multivariate long-term series forecasting results.

Method		LHRTF		MICN		FEDformer		Autoformer		Informer		Pyraformer		LogTrans		SCINet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.161**	**0.210**	**0.161**	0.229	0.217	0.296	0.266	0.336	0.300	0.384	0.622	0.556	0.458	0.490	0.239	0.271
	192	**0.209**	**0.252**	0.220	0.281	0.276	0.336	0.307	0.367	0.598	0.544	0.739	0.624	0.658	0.589	0.283	0.303
	336	**0.268**	**0.295**	0.278	0.331	0.339	0.380	0.359	0.395	0.578	0.523	1.004	0.753	0.797	0.652	0.330	0.335
	720	0.325	**0.344**	**0.311**	0.356	0.403	0.428	0.419	0.428	1.059	0.741	1.420	0.934	0.869	0.675	0.400	0.379
	Avg.	**0.241**	**0.275**	0.243	0.299	0.309	0.360	0.338	0.382	0.634	0.548	0.946	0.717	0.696	0.602	0.313	0.322
Electricity	96	**0.156**	**0.250**	0.164	0.269	0.193	0.308	0.201	0.317	0.274	0.368	0.386	0.449	0.258	0.357	0.205	0.312
	192	**0.168**	**0.262**	0.177	0.285	0.201	0.315	0.222	0.334	0.296	0.386	0.378	0.443	0.266	0.368	0.197	0.308
	336	**0.185**	**0.280**	0.193	0.304	0.214	0.329	0.231	0.338	0.300	0.394	0.376	0.443	0.280	0.380	0.202	0.312
	720	0.227	**0.316**	**0.212**	0.321	0.246	0.355	0.254	0.361	0.373	0.439	0.376	0.445	0.283	0.376	0.234	0.338
	Avg.	**0.184**	**0.277**	0.187	0.295	0.214	0.327	0.227	0.338	0.311	0.397	0.379	0.445	0.272	0.370	0.210	0.318
ILI	24	**1.848**	**0.847**	2.684	1.112	3.228	1.260	3.483	1.287	5.764	1.677	7.394	2.012	4.480	1.444	2.782	1.106
	36	**1.947**	**0.888**	2.667	1.068	2.679	1.080	3.103	1.148	4.755	1.467	7.551	2.031	4.799	1.467	2.689	1.064
	48	**1.558**	**0.799**	2.558	1.052	2.622	1.078	2.669	1.085	4.763	1.469	7.662	2.057	4.800	1.468	2.324	0.999
	60	**1.917**	**0.892**	2.747	1.110	2.857	1.157	2.770	1.125	5.264	1.564	7.931	2.100	5.278	1.560	2.802	1.112
	Avg.	**1.817**	**0.856**	2.664	1.086	2.847	1.144	3.006	1.161	5.137	1.544	7.635	2.050	4.839	1.485	2.649	1.070
ETTh1	96	0.388	**0.412**	0.421	0.431	**0.376**	0.419	0.449	0.459	0.865	0.713	0.664	0.612	0.878	0.740	0.404	0.415
	192	0.450	**0.443**	0.474	0.487	**0.420**	0.448	0.500	0.482	1.008	0.792	0.790	0.681	1.037	0.824	0.456	0.445
	336	0.490	**0.465**	0.569	0.551	**0.459**	**0.465**	0.521	0.496	1.107	0.809	0.891	0.738	1.238	0.932	0.519	0.481
	720	**0.484**	**0.474**	0.770	0.672	0.506	0.507	0.514	0.512	1.181	0.865	0.963	0.782	1.135	0.852	0.564	0.528
	Avg.	0.453	**0.448**	0.559	0.535	**0.440**	0.460	0.496	0.487	1.040	0.795	0.827	0.703	1.072	0.837	0.486	0.467
ETTh2	96	**0.298**	**0.347**	0.299	0.364	0.358	0.397	0.358	0.397	3.755	1.525	0.645	0.597	2.116	1.197	0.312	0.355
	192	**0.379**	**0.402**	0.441	0.454	0.429	0.439	0.456	0.452	5.602	1.931	0.788	0.683	4.315	1.635	0.401	0.412
	336	0.429	0.444	0.654	0.567	0.496	0.487	0.482	0.486	4.721	1.835	0.907	0.747	1.124	1.604	**0.413**	**0.432**
	720	**0.446**	**0.456**	0.956	0.716	0.463	0.474	0.515	0.511	3.647	1.625	0.963	0.783	3.188	1.540	0.490	0.483
	Avg.	**0.388**	**0.412**	0.588	0.525	0.437	0.449	0.453	0.462	4.431	1.729	0.826	0.703	2.686	1.494	0.404	0.421
ETTm1	96	0.322	**0.361**	**0.316**	0.362	0.379	0.419	0.505	0.475	0.672	0.571	0.543	0.510	0.600	0.546	0.350	0.385
	192	**0.362**	**0.382**	0.363	0.390	0.426	0.441	0.553	0.496	0.795	0.669	0.557	0.537	0.837	0.700	0.382	0.400
	336	**0.391**	**0.403**	0.408	0.426	0.445	0.459	0.621	0.537	1.212	0.871	0.754	0.655	1.124	0.832	0.419	0.425
	720	**0.458**	**0.442**	0.481	0.476	0.543	0.490	0.671	0.561	1.166	0.823	0.908	0.724	1.153	0.820	0.494	0.463
	Avg.	**0.383**	**0.397**	0.392	0.414	0.448	0.452	0.588	0.517	0.961	0.734	0.691	0.607	0.929	0.725	0.411	0.418
ETTm2	96	**0.179**	**0.265**	**0.179**	0.275	0.203	0.287	0.255	0.339	0.365	0.453	0.435	0.507	0.768	0.642	0.201	0.280
	192	**0.240**	**0.303**	0.307	0.376	0.269	0.328	0.281	0.340	0.533	0.563	0.730	0.673	0.989	0.757	0.283	0.331
	336	**0.294**	**0.339**	0.325	0.388	0.325	0.366	0.339	0.372	1.363	0.887	1.201	0.845	1.334	0.872	0.318	0.352
	720	**0.392**	**0.397**	0.502	0.490	0.421	0.415	0.422	0.419	3.379	1.338	3.625	1.451	3.048	1.328	0.439	0.423
	Avg.	**0.276**	**0.326**	0.328	0.382	0.305	0.349	0.324	0.368	1.410	0.810	1.498	0.869	1.535	0.900	0.310	0.347

other datasets. The forecasting horizons were $\{24, 36, 48, 60\}$ for ILI dataset and $\{96, 192, 336, 720\}$ for the other datasets. Table 1 summarizes the results of multivariate time series prediction for seven datasets. The best result is highlighted in bold, while the second-best result is underlined. LHRTF consistently achieved top-tier performance across all benchmark tests, outperforming almost all baselines. Moreover, We evaluated the experimental results with a significance test using the described method [30] to confirm the validity of our method, as shown in Fig. 4. Compared to the previously best-performing model MICN, LHRTF showed significant improvements on the ILI and ETTh2 datasets, with our results achieving a relative reduction in MSE and MAE averages of 31.7% and 21.2% for ILI, and 34% and 21.5% for ETTh2, respectively. In particular, compared to the previous state-of-the-art results, we achieved an overall reduction in MSE by 14.9% and a decrease in MAE by 13.1%.

4.5 Ablation Study

To demonstrate the necessity and effectiveness of trend, seasonal modular, and multilevel decomposition modeling, we conduct related ablation studies. LHRTF-Micn-Trend replaces our MLP with a single linear layer to predict the trend module, while LHRTF-Auto-Trend uses the mean to predict the trend module in Autoformer instead of our MLP. LHRTF w/o RevIN removes instance normalization, and LHRTF w/o Seasonal Decomp indicates that the second-stage seasonal decomposition module has been removed. We conduct experiments on one large electricity dataset, and the experimental results are shown in Fig. 5. It is evident that our multilayer perceptron network performs the best, further demonstrating its effectiveness in trend modeling. The removal of instance normalization shows some errors, but we still achieve the best results compared to MICN methods, indicating the importance of normalization for non-stationary time series data. When the two-stage decomposition module is removed, the performance of the model is poor, which fully demonstrates the effectiveness and rationality of the two-stage decomposition.

5 Conclusion

We propose a multi-hierarchical decomposition framework based on a frequency-domain attention mechanism, which leverages the powerful information extraction capability of deep learning to obtain information on potential variables for time series forecasting, and provides more accurate and reliable solutions for time series forecasting. We emphasize the importance of respecting and utilizing the complex patterns of time series and propose a multi-decomposition architecture for trend and season module forecasting. In the seasonal forecasting module, we model potential different patterns at different granularities, decompose the input seasonal sequence into multiple identical subsequences, capture fine-grained local time dependencies within each subsequence through high-dimensional embeddings, and then model global time dependencies at different scales using Fourier attention. Extensive experiments demonstrate the effectiveness of our model in long-term time series forecasting, and it achieves linear complexity.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (NSFC) under grants 62102097, 61976058, and 92267107, the Science and Technology Planning Project of Guangdong Province under grants: 2023A1515012855 and 2022A1515011592, 2021B0101220006, 2019A050510041, and 2021A1515012300, the Key Science and Technology Planning Project of Yunnan Province under grant 202102AA100012, and the Science and Technology Program of Guangzhou under grants 202201010548 and 202103000034.

References

1. Zhang, Y., Cheung, Y.M.: Discretizing numerical attributes in decision tree for big data analysis. In: ICDMW, pp. 1150–1157. IEEE (2014)

2. Zhao, L., Zhang, Y., et al.: Heterogeneous drift learning: classification of mix-attribute data with concept drifts. In: DSAA, pp. 1–10. IEEE (2022)
3. Zeng, A., Rong, H., et al.: Discovery of genetic biomarkers for Alzheimers disease using adaptive convolutional neural networks ensemble and genome-wide association studies. Interdiscip. Sci. **13**(4), 787–800 (2021)
4. Zhang, Z., Zhang, Y., et al.: Time-series data imputation via realistic masking-guided tri-attention Bi-GRU. In: ECAI, pp. 1–9 (2023)
5. Zhao, M., Zhang, Y., et al.: Unsupervised concept drift detection via imbalanced cluster discriminator learning. In: PRCV, pp. 1–12 (2023)
6. Mittelman, R.: Time-series modeling with undecimated fully convolutional neural networks. arXiv preprint arXiv:1508.00317 (2015)
7. Lai, G., Chang, W.C., et al.: Modeling long-and short-term temporal patterns with deep neural networks. In: SIGIR, pp. 95–104 (2018)
8. He, Y., Zhao, J.: Temporal convolutional networks for anomaly detection in time series. In: Journal of Physics: Conference Series, p. 042050 (2019)
9. Vaswani, A., Shazeer, N., et al.: Attention is all you need. In: NeurIPS, pp. 5998–6008 (2017)
10. Li, S., Jin, X., et al.: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: NeurIPS, pp. 5244–5254 (2019)
11. Kitaev, N., et al.: Reformer: the efficient transformer. In: ICLR (2020)
12. Cirstea, R., Guo, C., et al.: Triformer: triangular, variable-specific attentions for long sequence multivariate time series forecasting. In: IJCAI, pp. 1994–2001 (2022)
13. Zhou, H., Zhang, S., et al.: Informer: beyond efficient transformer for long sequence time-series forecasting. In: AAAI, pp. 11106–11115 (2021)
14. Flunkert, V., Salinas, D., et al.: Deepar: probabilistic forecasting with autoregressive recurrent networks. arXiv preprint arXiv:1704.04110 (2017)
15. Li, Y., Moura, J.M.F.: Forecaster: a graph transformer for forecasting spatial and time-dependent data. In: ECAI, vol. 325, pp. 1293–1300 (2020)
16. Child, R., Gray, S., et al.: Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509 (2019)
17. Xu, K., Qin, M., et al.: Learning in the frequency domain. In: CVPR, pp. 1740–1749 (2020)
18. Guibas, J., Mardani, M., et al.: Adaptive fourier neural operators: efficient token mixers for transformers. arXiv preprint arXiv:2111.13587 (2021)
19. Wu, H., Xu, J., et al.: Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In: NeurIPS, pp. 22419–22430 (2021)
20. Zhou, T., Ma, Z., et al.: Fedformer: frequency enhanced decomposed transformer for long-term series forecasting. In: ICML, pp. 27268–27286 (2022)
21. Woo, G., Liu, C., et al.: Etsformer: exponential smoothing transformers for time-series forecasting. arXiv preprint arXiv:2202.01381 (2022)
22. Jiang, S., Syed, T., et al.: Bridging self-attention and time series decomposition for periodic forecasting. In: CIKM, pp. 3202–3211 (2022)
23. Wang, H., Peng, J., et al: MICN: multi-scale local and global context modeling for long-term series forecasting. In: ICLR (2023)
24. Kim, T., Kim, J., et al.: Reversible instance normalization for accurate time-series forecasting against distribution shift. In: ICLR (2021)
25. UCI: Electricity. https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014
26. Wetterstation: Weather. https://www.bgc-jena.mpg.de/wetter/
27. CDC: Illness. https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

28. Liu, S., Yu, H., et al.: Pyraformer: low-complexity pyramidal attention for long-range time series modeling and forecasting. In: ICLR (2021)
29. Liu, M., Zeng, A., et al.: Scinet: time series modeling and forecasting with sample convolution and interaction. In: NeurIPS, pp. 5816–5828 (2022)
30. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)