

FilterTS: Comprehensive Frequency Filtering for Multivariate Time Series Forecasting

Yulong Wang^{1,2}, Yushuo Liu^{1,2}, Xiaoyi Duan^{1,2}, Kai Wang^{1,2,*}

¹College of Computer Science, Nankai University

²Tianjin Media Computing Center

{yl.wang, yushuo.liu, xyd}@mail.nankai.edu.cn, wangk@nankai.edu.cn

Abstract

Multivariate time series forecasting is crucial across various industries, where accurate extraction of complex periodic and trend components can significantly enhance prediction performance. However, existing models often struggle to capture these intricate patterns. To address these challenges, we propose FilterTS, a novel forecasting model that utilizes specialized filtering techniques based on the frequency domain. FilterTS introduces a Dynamic Cross-Variable Filtering Module, a key innovation that dynamically leverages other variables as filters to extract and reinforce shared variable frequency components across variables in multivariate time series. Additionally, a Static Global Filtering Module captures stable frequency components, identified throughout the entire training set. Moreover, the model is built in the frequency domain, converting time-domain convolutions into frequency-domain multiplicative operations to enhance computational efficiency. Extensive experimental results on eight real-world datasets have demonstrated that FilterTS significantly outperforms existing methods in terms of prediction accuracy and computational efficiency.

code — <https://github.com/wyl010607/FilterTS>

Introduction

Multivariate time series forecasting is crucial across various domains such as finance(Olivares et al. 2023), transportation(Bui, Cho, and Yi 2022), and energy(Zhou et al. 2021), where the data often exhibits complex temporal characteristics like periodic changes and trends(Qiu et al. 2024).

Traditional time series analysis methods predominantly focus on time-domain analysis (Zhou et al. 2021; Nie et al. 2023; Liu et al. 2024), yet they often fall short in capturing periodic information directly(Wang et al. 2023). Recent studies have increasingly applied frequency domain analysis to time series, which transforms time series into the frequency space, allowing distinct frequency components to be clearly separated (Xu, Zeng, and Xu 2024; Yi et al. 2024). Despite these advancements, these methods sometimes fail to effectively differentiate the importance of various frequency components, treating all components equally, which

*Corresponding author

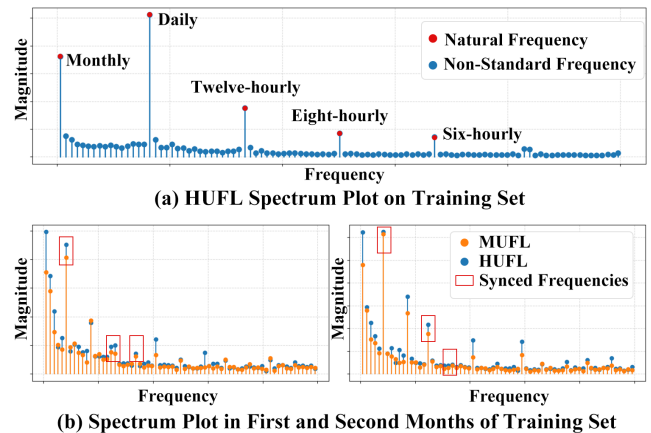


Figure 1: Illustration of Stable and Synchronized Variable Frequency Components Using the ETTh1 Dataset. (a) displays the frequency spectrum of the HUFL variable across the entire training set, highlighting dominant stable frequency components with red dots. (b) shows synchronized variable frequency components between HUFL and MUFL variables within a rapidly fluctuating input window (outlined with a red rectangle), after subtracting dominant stable frequency components.

might lead to model overfitting while neglecting crucial a priori periodic information and inter-variable dependencies.

To address these challenges, an effective strategy is the selective extraction and emphasis of frequency components that are most predictive of future observations. We categorize features in time series into (a) stable frequency components and (b) variable frequency components based on their behavior in the frequency domain. Stable components, such as natural cycles (daily, weekly, monthly) and frequencies associated with specific business processes, often appear as dominant frequencies in the series, as determined through frequency domain analysis on training data, depicted in Figure 1a. Variable components, whose characteristics may change over time due to environmental or external factors, are often elusive with static filters. Capturing these dynamically changing frequency components effectively remains a research gap. We hypothesize, based on experimen-

tal observations, that these variable frequency components are not only present within individual variables but may also be shared across different variables in a multivariate time series (Zhao and Shen 2024). Specifically, there may be synchronization in frequency and intensity changes among these variables, as shown in Figure 1b, revealing underlying inter-variable connections crucial for enhancing multivariate forecasting models.

Based on the foregoing analysis, we introduce FilterTS, an innovative multivariate time series forecasting model that enhances and precisely extracts frequency components through carefully designed filtering modules. The model operates in the frequency domain, converting time-domain convolution to frequency-domain multiplication to improve computational efficiency. FilterTS utilizes two types of filtering modules:

The Static Global Filtering Module are constructed by performing frequency analysis on the entire training set, building band-pass filters at frequencies corresponding to components with relatively high amplitudes, thereby capturing the dominant stable frequency components. The Dynamic Cross-variable Filtering Module, on the other hand, treats each variable as a filter for the others, dynamically extracting shared frequency components across variables within each lookback window. This approach enhances the capture of variable frequency components. The output sequences from these filters are then merged by a complex matrix.

The experimental results detailed in this paper substantiate the exceptional performance of the FilterTS model across eight real datasets, FilterTS demonstrates superior forecasting accuracy and computational efficiency compared to existing state-of-the-art methods. The main contributions of this paper are summarized as follows:

- We introduce FilterTS, a novel multivariate time series forecasting model leveraging filters to enhance frequency component extraction and improve prediction accuracy.
- We develop the Static Global Filtering Module, designed to capture stable periodic components, and the Dynamic Cross-Variable Filtering Module, which dynamically extracts and emphasizes significant frequency components shared across variables.
- Our empirical evaluation across eight real-world datasets demonstrates that FilterTS surpasses existing methods in forecasting accuracy and computational efficiency.

Related Work

Multivariate Time Series Forecasting Models

Multivariate Time Series (MTS) involves a set of simultaneously sampled time series data. Commonly, MTS forecasting models attempt to capture dependencies among variables, employing methods such as MLPs (Shao et al. 2022; Ekambaram et al. 2023; Zeng et al. 2023; Zhang and Yan 2023; Huang et al. 2024), CNNs (Wu et al. 2023; Luo and Wang 2024), GNNs (Wu et al. 2020; Cao et al. 2020; Cai et al. 2024), and Transformers (Zhou et al. 2021; Liu et al. 2024) to learn these relationships. However, empirical evidence suggests that models that do not explicitly model these

dependencies can still achieve strong performance (Nie et al. 2023; Zhou et al. 2023; Xu, Zeng, and Xu 2024). This may be attributed to the tendency of models that explicitly account for inter-variable dependencies to overfit complex variable interactions when the available prior knowledge is insufficient (Nie et al. 2023).

Frequency-Domain Enhanced Time Series Forecasting Models

Recent developments in time series forecasting have increasingly leveraged frequency-domain analysis to improve predictive accuracy. This approach focuses on extracting and utilizing periodic and global dependencies that are often more discernible in the frequency domain than in the time domain. FEDformer (Zhou et al. 2022) revolutionizes forecasting by applying self-attention directly in the frequency domain, shifting the focus from time-domain dynamics to spectral characteristics. FreTS (Yi et al. 2024) optimizes predictions by utilizing frequency-domain MLPs to capture static frequency features. FITS (Xu, Zeng, and Xu 2024) simplifies models by selectively filtering out high-frequency noise, maintaining only essential low-frequency information. Additionally, TSLANet (Eldele et al. 2024) incorporates adaptive high-frequency filtering techniques to mitigate noise. Despite these technological advancements, a prevalent limitation within these models is their primary focus on static frequency features, which neglects the dynamic changes among variables within the frequency domain.

Preliminary

Problem Formulation

Given a multivariate time series $X \in \mathbb{R}^{N \times L}$, where N denotes the number of variables and L the lookback window length, the task of multivariate time series forecasting is to predict the values in the future forecasting window. Specifically, for each variable $i \in \{1, \dots, N\}$, the series is represented as $X_i = [x_{i,1}, \dots, x_{i,L}]^T$. The objective is to estimate the future values $\hat{X} \in \mathbb{R}^{N \times F}$, where F denotes the length of the forecasting window. The forecast values for each variable are given by $\hat{X}_i = [\hat{x}_{i,L+1}, \dots, \hat{x}_{i,L+F}]^T$ for $i \in \{1, \dots, N\}$.

Fast Fourier Transform (FFT)

The Fast Fourier Transform (FFT)(Duhamel and Vetterli 1990) is an efficient algorithm for computing the Discrete Fourier Transform (DFT) of a sequence. Consider a univariate time series $z_t \in \mathbb{R}^L$, where t represents the time index, and L is the length of the sequence. The FFT transforms this time series from the time domain into its frequency domain representation $\mathcal{Z}_f \in \mathbb{C}^L$, where f represents the frequency index. The transformation is expressed as:

$$\mathcal{Z}_f = \sum_{t=0}^{L-1} z_t \cdot e^{-j \frac{2\pi}{L} ft}, \quad f = 0, 1, \dots, L-1 \quad (1)$$

where j is the imaginary unit. The frequency domain representation \mathcal{Z}_f can be decomposed into its real and imagi-

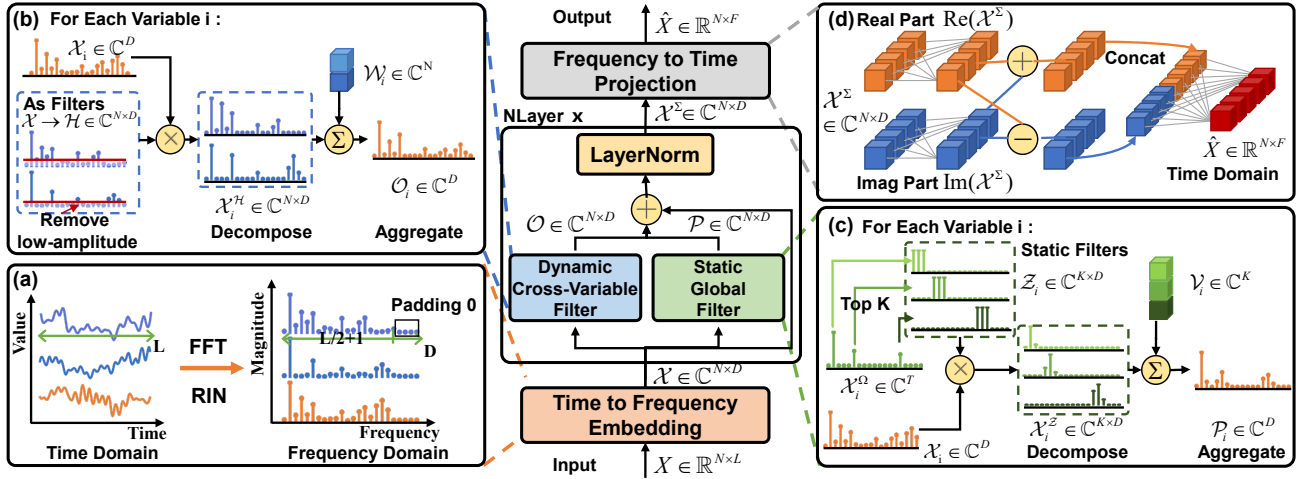


Figure 2: Overview of FilterTS, which comprises the following key modules: (a) the Time to Frequency Embedding Module, responsible for converting the time-domain series into the frequency domain; (b) the Dynamic Cross-Variable Filtering Module, designed to extract variable frequency components; (c) the Static Global Filtering Module, which focuses on capturing stable frequency components; and finally, (d) the Frequency to Time Projection Module, which transforms the frequency domain representation back into the time domain to generate the final predictions.

nary components, denoted as $\text{Re}(\mathcal{Z}_f)$ and $\text{Im}(\mathcal{Z}_f)$, respectively. The magnitude (or amplitude) of \mathcal{Z}_f , representing the strength of the frequency components, is defined as:

$$|\mathcal{Z}_f| = \sqrt{\text{Re}(\mathcal{Z}_f)^2 + \text{Im}(\mathcal{Z}_f)^2} \quad (2)$$

Method

Overall Structure

Figure 2 illustrates the comprehensive architecture of the FilterTS model, comprised of four main components: the Time to Frequency Embedding Module (T2FEmbed), the Dynamic Cross-Variable Filtering Module (DCFilter), the Static Global Filtering Module (SGFilter), and the Frequency to Time Projection Layer (F2TOut).

We first transform the given multivariate time series $X \in \mathbb{R}^{N \times L}$ into a frequency domain representation \mathcal{X} to facilitate subsequent filtering operations:

$$\mathcal{X} = \text{T2FEmbed}(X) \quad (3)$$

The resulting $\mathcal{X} \in \mathbb{C}^{N \times D}$ represents the data in a frequency domain format optimized for neural network processing, where \mathbb{C} denotes the complex domain and D represents the dimension of hidden layers.

The frequency domain representation \mathcal{X} is then concurrently fed into two distinct pathways:

$$\mathcal{O} = \text{DCFilter}(\mathcal{X}) \quad (4)$$

$$\mathcal{P} = \text{SGFilter}(\mathcal{X}) \quad (5)$$

where $\text{DCFilter}(\cdot)$ and $\text{SGFilter}(\cdot)$ denote the Dynamic Cross-Variable Filtering Module and Static Global Filtering Module, respectively. Both \mathcal{O} and \mathcal{P} are frequency domain representations in $\mathbb{C}^{N \times D}$, corresponding to the cross-variable and stable filtering outputs.

Subsequently, the outputs from the embedding and both filtering modules are aggregated:

$$\mathcal{X}^\Sigma = \text{LayerNorm}(\alpha\mathcal{X} + \beta\mathcal{O} + \gamma\mathcal{P}) \quad (6)$$

where α , β , and γ are weighting coefficients optimized during training. We apply LayerNorm to stabilize the training process.

This process can be repeated for e layers, where equations (4) to (6) are repeated e times. In each iteration, the output \mathcal{X}^Σ from the previous layer serves as the input \mathcal{X} in equation (4) for the next layer.

Finally, the aggregated frequency domain representation \mathcal{X}^Σ is fed into the Frequency-to-Time Projection $\text{LayerF2TOut}(\cdot)$, which converts the frequency domain back into the time domain, generating forecasts $\hat{X} \in \mathbb{R}^{N \times F}$ for F future steps:

$$\hat{X} = \text{F2TOut}(\mathcal{X}^\Sigma) \quad (7)$$

Time to Frequency Embedding

The Time-to-Frequency Embedding Module converts the time-domain input series into the frequency domain, leveraging the equivalence between time-domain convolution and frequency-domain multiplication to enhance computational efficiency in subsequent filtering stages.

Firstly, we apply instance normalization to each time series $X_i \in \mathbb{R}^L$ to mitigate the effects of varying data distributions (Kim et al. 2022):

$$\tilde{X}_i = \frac{X_i - \mu(X_i)}{\sigma(X_i) + \epsilon} \quad (8)$$

where $\mu(X_i)$ and $\sigma(X_i)$ are the mean and standard deviation of each individual time series, and ϵ is a small constant to ensure numerical stability.

Secondly, the Fast Fourier Transform (FFT) is applied to convert the normalized time series into the frequency domain efficiently:

$$\mathcal{X}_i^+ = \text{FFT} \left(\tilde{X}_i \parallel \text{zeros}(L) \right) \quad (9)$$

In this formulation, \tilde{X}_i is zero-padded by appending L zeros, increasing the sequence length to $2L$. This ensures that the frequency domain multiplication corresponds to linear convolution in the time domain, rather than circular convolution. The FFT then transforms this extended sequence into a complex number spectrum $\mathcal{X}_i^+ \in \mathbb{C}^{2L}$. For an in-depth introduction to FFT and extended-length effects, please see the Appendix.

Finally, to match the required model dimensionality D , while maintaining fidelity to the frequency domain representation, Fourier interpolation is performed:

$$\mathcal{X}_{i,j} = \begin{cases} \mathcal{X}_{i,j}^+ & \text{if } j < L + 1 \\ 0 & \text{if } j \geq L + 1 \end{cases} \quad \text{for } j = 1, \dots, D \quad (10)$$

Given the conjugate symmetry of the FFT for real-valued signals, only the first $L + 1$ components contain unique information. Therefore, this operation either extends the truncated frequency spectrum by zero-padding if D exceeds $L + 1$ or reduces it by selecting only the first D low-frequency components.

Dynamic Cross-Variable Filtering Module

Time Series as Filter The foundational theory of utilizing time series as finite impulse response (FIR) filters is grounded in the signal processing principle that the filtering operation can be represented as convolution in the time domain or multiplication in the frequency domain. This section discusses the transformation of a time series into an FIR filter and its practical implications.

An FIR filter is characterized by a finite sequence of N coefficients $h[m]$, defining the filter's impulse response in the time domain:

$$y[n] = \sum_{m=0}^{M-1} h[m] \cdot x[n-m] \quad (11)$$

The output signal $y[n]$ is determined by convolving the input signal $x[n]$ with the filter coefficients $h[m]$, where M indicates the memory of the filter.

Consider two time series, each of length L : one representing the input signal $x = [x_1, \dots, x_L]$ and the other acting as the filter $h = [h_1, \dots, h_L]$. The filtered output signal y of the FIR filter is obtained by convolving these two sequences:

$$y_n = \sum_{m=1}^L h_m \cdot x_{n-m}, \quad \text{for } n = 1, \dots, L \quad (12)$$

where y_n denotes the filtered time series at time n .

The convolution theorem facilitates this operation by transforming the time-domain convolution into a frequency-domain multiplication:

$$Y_f = X_f \cdot H_f, \quad \text{for } f = 1, \dots, L \quad (13)$$

where Y_f , X_f , and H_f are the discrete Fourier transforms of y , x and h at frequency f , respectively.

This principle enables significant frequency components in X_f and H_f to produce pronounced energy in Y_f , thereby accentuating the frequencies common to both sequences. This characteristic is particularly advantageous for the analysis of interactions within the frequency domain.

Generation of Dynamic Cross-Variable Filters Building on the concept of utilizing time series as filters, we leverage the frequency domain representation $\mathcal{X} \in \mathbb{C}^{N \times D}$ to generate dynamic filters. In this context, dynamic refers to the construction of filters that adapt within each look-back window, capturing key frequency components that are shared across variables.

Dynamic filters are derived by evaluating the amplitude of each frequency component. To focus on significant frequencies and reduce the influence of noise, components below a certain amplitude threshold are discarded. This threshold is dynamically set based on the amplitude distribution of each variable's frequency components:

$$\tau_i = \text{quantile}(|\mathcal{X}_i|, \alpha) \quad (14)$$

where τ_i represents the α -quantile, defining the minimum amplitude threshold for the i -th variable to consider a frequency component significant.

The dynamic filter for each frequency component $\mathcal{H}_{i,f}$ is then defined as:

$$\mathcal{H}_{i,f} = \begin{cases} \mathcal{X}_{i,f} & \text{if } |\mathcal{X}_{i,f}| > \tau_i \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where $|\mathcal{X}_{i,f}|$ representing the magnitude of the f -th frequency component for the i -th variable. The resulting dynamic filters, which capture key frequency components of each variable's real-time input, are combined to filter each variable's original series to enhance shared frequency components across variables. These dynamic filters are assembled into a matrix $\mathcal{H} = [\mathcal{H}_1, \dots, \mathcal{H}_N]$ of dimensions $\mathbb{C}^{N \times D}$.

Filtering Decomposition and Aggregation After obtaining the dynamic cross-variable filters \mathcal{H} , we return to the processing of the input time series. Upon adjusting the amplitude and phase of the frequency domain representations using the scaling matrix $\mathcal{A}^o \in \mathbb{C}^{N \times D}$, we obtain the adjusted frequency domain representation \mathcal{X}^o :

$$\mathcal{X}^o = \mathcal{X} \odot \mathcal{A}^o \quad (16)$$

For each variable i and each dynamic filter k , the adjusted representation is multiplied by the respective filter to obtain:

$$\mathcal{X}_{i,k}^{\mathcal{H}} = \mathcal{X}_i^o \odot H_k^* \quad (17)$$

where \odot denotes the element-wise multiplication. H_k^* is the conjugate of the k -th dynamic filter. Here, $\mathcal{X}_{i,k}^{\mathcal{H}} \in \mathbb{C}^D$ represents the subsequence generated by filtering the time series i using the dynamic filter derived from time series k .

Next, aggregation of these filtered subsequences is performed using a complex weight matrix $\mathcal{W} \in \mathbb{C}^{N \times N}$, which

is processed through modified softmax and ReLU functions to form a sparse matrix:

$$\mathcal{W}^* = \text{softmax}(\text{ReLU}(\mathcal{W})) \quad (18)$$

The weighted aggregation for each variable i is then computed as follows:

$$\mathcal{O}_i = \sum_{k=1}^N \mathcal{X}_{i,k}^H \cdot W_{i,k}^* \quad (19)$$

$$\mathcal{O} = [\mathcal{O}_1, \dots, \mathcal{O}_N] \quad (20)$$

Here, $\mathcal{O} \in \mathbb{C}^{N \times D}$ represents the output of the Dynamic Cross-Variable Filtering Module. It enhances shared frequency components across variables, allowing the model to effectively capture and utilize critical inter-variable interactions through dynamic filtering.

Static Global Filtering Module

The Static Global Filtering Module identifies dominant stable frequency components across the entire training dataset. By constructing band-pass filters targeting the top K high-amplitude frequencies, it extracts key frequency representations from each input sequence during filtering.

Generation of Static Global Filters Let the training dataset $X^\Omega \in \mathbb{R}^{N \times T}$ consist of multivariate time series, where N is the number of variables and T is the length of each sequence. The Fourier transform is applied to obtain the global frequency domain representation $\mathcal{X}^\Omega \in \mathbb{C}^{N \times T}$:

$$\mathcal{X}^\Omega = \text{FFT}(X^\Omega) \quad (21)$$

Since the model's input sequence $X \in \mathbb{R}^{N \times L}$ has a look-back window length L , the frequency resolution of \mathcal{X}^Ω (of length T) is higher than that of the input sequence. To align the frequency resolutions, we perform down-sampling on \mathcal{X}^Ω to reduce its length from T to L . For each variable i , the down-sampled frequency representation is obtained by summing the magnitudes of frequency components over non-overlapping windows κ of size T/L :

$$\tilde{\mathcal{X}}_{i,f}^\Omega = \sum_{m=\kappa \times f}^{\kappa \times (f+1) - 1} \mathcal{X}_{i,m}^\Omega \quad (22)$$

where $\tilde{\mathcal{X}}_{i,f}^\Omega$ represents the downsampled magnitude of frequency component f for variable i , with $i \in \{1, \dots, N\}$ and $f \in \{1, \dots, L\}$.

For each variable $\tilde{\mathcal{X}}_i^\Omega$, the top K frequency components are selected based on their magnitudes. Let $\{f_{i,1}^*, f_{i,2}^*, \dots, f_{i,K}^*\}$ denote the indices of these top K frequencies for variable i . For each selected frequency $f_{i,s}^*$, define a band-pass filter $\mathcal{Z}_{i,s}$ as:

$$\mathcal{Z}_{i,s,f'} = \begin{cases} 1 & \text{if } f' \in [f_{i,s}^* - \Delta f, f_{i,s}^* + \Delta f] \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

where Δf represents the half bandwidth of the filter, accounting for minor deviations around the dominant frequency, and $f' \in \{1, \dots, D\}$ to match the dimensionality

of the input frequency domain representation $\mathcal{X}_i \in \mathbb{C}^D$, as both the filter $\mathcal{Z}_{i,s}$ have the same dimensions \mathbb{C}^D .

These static filters of variable i are assembled into a matrix $\mathcal{Z}_i = [\mathcal{Z}_{i,1}, \dots, \mathcal{Z}_{i,K}]$ of dimensions $\mathbb{C}^{K \times D}$.

Filtering Decomposition and Aggregation The frequency domain representation of the model's multivariate input $\mathcal{X} \in \mathbb{C}^{N \times D}$, obtained through Fourier transform, is first adjusted in amplitude and phase using an amplitude scaling matrix $\mathcal{A}^p \in \mathbb{C}^{N \times D}$:

$$\mathcal{X}^p = \mathcal{X} \odot \mathcal{A}^{pend} \quad (24)$$

For each variable i and each static filter s in \mathcal{Z}_i , the adjusted frequency representation is filtered by multiplying it with the corresponding static filter:

$$\mathcal{X}_{i,k}^{\mathcal{Z}} = \mathcal{X}_i^p \odot \mathcal{Z}_{i,s} \quad (25)$$

where \odot denotes element-wise multiplication, and $\mathcal{Z}_{i,s}$ is the static filter s for variable i .

To aggregate the filtered representations, a complex weight matrix $\mathcal{V} \in \mathbb{C}^{N \times K}$ is applied, which is first processed through a modified softmax and ReLU function to enforce sparsity:

$$\mathcal{V}^* = \text{softmax}(\text{ReLU}(\mathcal{V})) \quad (26)$$

The weighted aggregation for each variable is then computed through matrix multiplication:

$$\mathcal{P}_i = \sum_{s=1}^K \mathcal{X}_{i,k}^{\mathcal{Z}} \cdot \mathcal{V}_{i,s}^* \quad (27)$$

Finally, the static filtered sequences for all variables are assembled into the output matrix $\mathcal{P} = [\mathcal{P}_1, \dots, \mathcal{P}_N]$ as the static global filtering module's output. This sequence of operations enhances static frequency components within the data, effectively leveraging the inherent static frequencies of the sequences for robust feature extraction.

Frequency to Time Projection

Finally to transform the frequency domain representation \mathcal{X}^Σ back into the time domain, we first apply a linear transformation to handle the real and imaginary components of the frequency domain data:

$$\text{Re}(\hat{\mathcal{X}}) = \text{Re}(\mathcal{X}^\Sigma) \cdot \mathbf{U}^{\text{Re}} - \text{Im}(\mathcal{X}^\Sigma) \cdot \mathbf{U}^{\text{Im}} \quad (28)$$

$$\text{Im}(\hat{\mathcal{X}}) = \text{Re}(\mathcal{X}^\Sigma) \cdot \mathbf{U}^{\text{Im}} + \text{Im}(\mathcal{X}^\Sigma) \cdot \mathbf{U}^{\text{Re}} \quad (29)$$

Here, \mathbf{U}^{Re} and \mathbf{U}^{Im} represent the weight matrices for the real and imaginary components, respectively, in the space $\mathbb{R}^{D \times D}$. This dual transformation preserves the properties of the complex numbers, such as phase relationships and amplitude variations. Subsequently, the real and imaginary parts are concatenated and then linearly transformed to map the combined complex data into the predicted time series output:

$$X^\parallel = [\text{Re}(\hat{\mathcal{X}}) \parallel \text{Im}(\hat{\mathcal{X}})] \quad (30)$$

$$\hat{X} = X^\parallel \cdot \mathbf{Q} \quad (31)$$

where \mathbf{Q} is the final linear transformation matrix with dimensions $\mathbb{R}^{2D \times F}$, and \hat{X} is the final predicted time series in $\mathbb{R}^{D \times F}$.

Models		Ours		TimeMixer		iTransformer		PatchTST		Crossformer		TimesNet		FreTS	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.321	0.360	0.333	0.371	0.334	0.368	<u>0.329</u>	<u>0.367</u>	0.404	0.426	0.338	0.375	0.335	0.371
	192	0.363	0.382	0.367	0.386	0.377	0.391	<u>0.367</u>	<u>0.385</u>	0.450	0.451	0.374	0.387	0.377	0.394
	336	0.395	0.403	<u>0.397</u>	<u>0.408</u>	0.426	0.420	0.399	0.410	0.532	0.515	0.410	0.411	0.413	0.418
	720	0.462	0.438	<u>0.460</u>	<u>0.445</u>	0.491	0.459	0.454	0.439	0.666	0.589	0.478	0.450	0.483	0.461
	Avg	0.385	0.396	0.389	0.403	0.407	0.410	<u>0.387</u>	<u>0.400</u>	0.513	0.495	0.400	0.406	0.402	0.411
ETTm2	96	0.172	0.255	<u>0.174</u>	<u>0.258</u>	0.180	0.264	0.175	0.259	0.287	0.366	0.187	0.267	0.181	0.269
	192	0.237	0.299	<u>0.239</u>	<u>0.302</u>	0.250	0.309	0.241	<u>0.302</u>	0.414	0.492	0.249	0.309	0.249	0.322
	336	<u>0.299</u>	0.398	0.296	<u>0.340</u>	0.311	0.348	0.305	0.343	0.597	0.542	0.321	0.351	0.340	0.382
	720	<u>0.397</u>	0.394	0.393	<u>0.397</u>	0.412	0.407	0.402	0.400	1.730	1.042	0.408	0.403	0.449	0.455
	Avg	0.276	0.321	0.276	<u>0.324</u>	0.288	0.332	0.281	0.326	0.757	0.611	0.291	0.333	0.305	0.357
ETTh1	96	0.374	0.391	<u>0.384</u>	<u>0.398</u>	0.386	0.405	0.414	0.419	0.423	0.448	<u>0.384</u>	0.402	0.390	0.404
	192	0.424	0.421	<u>0.439</u>	<u>0.429</u>	0.441	0.436	0.460	0.445	0.471	0.474	<u>0.436</u>	<u>0.429</u>	0.448	0.439
	336	0.464	0.441	<u>0.487</u>	<u>0.455</u>	<u>0.487</u>	0.458	0.501	0.466	0.570	0.546	0.491	0.469	0.501	0.470
	720	0.470	0.466	0.502	<u>0.482</u>	0.503	0.491	<u>0.500</u>	0.488	0.653	0.621	0.521	0.500	0.559	0.535
	Avg	0.433	0.430	<u>0.453</u>	<u>0.441</u>	0.454	0.448	0.469	0.455	0.529	0.522	0.458	0.450	0.475	0.462
ETTh2	96	0.290	0.338	<u>0.293</u>	<u>0.344</u>	0.297	0.349	0.302	0.348	0.745	0.584	0.340	0.374	0.317	0.373
	192	0.374	0.390	0.374	<u>0.396</u>	0.380	0.400	0.388	0.400	0.877	0.656	0.402	0.414	0.427	0.442
	336	0.406	0.420	0.416	<u>0.430</u>	0.428	0.432	0.426	0.433	1.043	0.731	0.452	0.452	0.526	0.505
	720	0.418	0.437	0.440	0.452	<u>0.427</u>	<u>0.445</u>	0.431	0.446	1.104	0.763	0.462	0.468	0.684	0.591
	Avg	0.372	0.396	<u>0.381</u>	<u>0.406</u>	0.383	0.407	0.387	0.407	0.942	0.684	0.414	0.427	0.489	0.478
Electricity	96	<u>0.151</u>	<u>0.245</u>	0.157	0.250	0.148	0.240	0.181	0.270	0.219	0.314	0.168	0.272	0.177	0.264
	192	<u>0.163</u>	<u>0.256</u>	0.177	0.263	0.162	0.253	0.188	0.274	0.231	0.322	0.184	0.289	0.183	0.271
	336	<u>0.180</u>	<u>0.274</u>	0.189	0.282	0.178	0.269	0.204	0.293	0.246	0.337	0.198	0.300	0.198	0.288
	720	<u>0.224</u>	0.311	0.233	0.316	0.225	0.317	0.246	0.324	0.280	0.363	0.220	0.320	0.234	0.322
	Avg	<u>0.180</u>	<u>0.271</u>	0.189	0.278	0.178	0.270	0.205	0.290	0.244	0.334	0.193	0.295	0.255	0.300
Exchange	96	0.081	0.199	0.087	0.205	0.086	0.206	0.088	<u>0.205</u>	0.256	0.367	0.107	0.234	<u>0.085</u>	0.211
	192	0.171	0.294	0.179	0.300	0.177	<u>0.299</u>	<u>0.176</u>	<u>0.299</u>	0.470	0.509	0.226	0.344	0.181	0.312
	336	<u>0.321</u>	<u>0.409</u>	0.333	0.417	0.331	0.417	0.301	0.397	1.268	0.883	0.367	0.448	0.471	0.508
	720	0.837	0.688	0.912	0.719	<u>0.847</u>	<u>0.691</u>	0.901	0.714	1.767	1.068	0.964	0.746	0.858	0.695
	Avg	0.352	0.397	0.378	0.410	<u>0.360</u>	<u>0.403</u>	0.367	0.404	0.940	0.707	0.416	0.443	0.399	0.432
Traffic	96	0.448	0.309	0.485	0.323	0.395	0.268	0.462	0.295	0.522	0.290	0.593	0.321	0.512	0.328
	192	<u>0.455</u>	0.307	0.488	0.322	0.417	0.276	0.466	0.296	0.530	<u>0.293</u>	0.617	0.336	0.507	0.325
	336	<u>0.472</u>	0.313	0.507	0.321	0.433	0.283	0.482	<u>0.304</u>	0.558	0.305	0.629	0.336	0.521	0.331
	720	<u>0.508</u>	0.332	0.549	0.335	0.467	0.302	0.514	<u>0.322</u>	0.589	0.328	0.640	0.350	0.562	0.348
	Avg	<u>0.471</u>	0.315	0.507	0.325	0.428	0.282	0.481	<u>0.304</u>	0.550	<u>0.304</u>	0.620	0.336	0.526	0.333
Weather	96	<u>0.162</u>	0.207	0.166	<u>0.213</u>	0.174	0.214	0.177	0.218	0.158	0.230	0.172	0.220	0.182	0.236
	192	<u>0.209</u>	<u>0.252</u>	<u>0.209</u>	0.251	0.221	0.254	0.225	0.259	0.206	0.277	0.219	0.261	0.219	0.217
	336	0.263	0.292	<u>0.264</u>	<u>0.293</u>	0.278	0.296	0.278	0.297	0.272	0.335	0.280	0.306	0.270	0.313
	720	<u>0.344</u>	<u>0.344</u>	0.342	0.343	0.358	0.349	0.354	0.348	0.398	0.418	0.365	0.359	0.348	0.380
	Avg	0.244	0.274	<u>0.245</u>	<u>0.275</u>	0.258	0.278	0.259	0.281	0.259	0.315	0.259	0.287	0.255	0.300
1st Count	23	28	5	2	<u>9</u>	<u>9</u>	2	1	2	0	1	0	0	0	

Table 1: Results of the multivariate long-term time series forecasting task, evaluated using MSE and MAE (lower is better). The lookback window for all models was set to 96. The best results are highlighted in **bold**, while the second-best results are underlined.

Experiments

Experimental Details

Datasets Following (Zhou et al. 2021) and (Nie et al. 2023), we evaluate our proposed model on eight widely-recognized multivariate time series forecasting datasets, spanning diverse domains such as energy, economics, transportation, and climate. Specifically, these datasets include the four ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2), Electricity, Exchange Rate, Traffic, and Weather.

Baselines We select several state-of-the-art multivariate time series forecasting models as baselines, including

TimeMixer (Wang et al. 2024), iTransformer (Liu et al. 2024), PatchTST (Nie et al. 2023), Crossformer (Zhang and Yan 2023), and TimesNet (Wu et al. 2023). Additionally, we incorporate FreTS (Yi et al. 2024), a novel model built upon frequency-domain.

Setup All experiments were conducted on an NVIDIA GeForce RTX 4090 24GB GPU. We adopted a consistent experimental setup identical to that of iTransformer (Liu et al. 2024) to ensure a fair comparison. Specifically, the lookback length for all models was fixed at 96, with prediction lengths set to $F \in \{96, 192, 336, 720\}$, Mean Squared Error (MSE)

was used as the loss function.

For detailed information on datasets, model descriptions, hyperparameter settings, and other experimental implementation specifics, please refer to the Appendix.

Main Results

Table 1 presents the predictive performance of FilterTS across eight multivariate long-term time series forecasting datasets, demonstrating superior accuracy over current state-of-the-art models in most cases. Specifically, averaging MSE across all prediction lengths, FilterTS achieved the best performance on 6 out of the 8 datasets and secured the second-best performance on the remaining 2 datasets.

The slightly weaker performance of FilterTS on the Traffic and ECL datasets can be attributed to the high number of variables in these datasets, which presents challenges for the static weight matrix in capturing complex inter-variable relationships, particularly under nonlinear conditions. In contrast, models like iTransformer, which utilize a sophisticated attention mechanism, dynamically adjust weights to better handle complex, multivariate interactions.

Despite these challenges, FilterTS reduced the average MSE by 4.24% compared to PatchTST, a representative of channel-independent models, indicating that effectively leveraging inter-variable information can significantly enhance model performance. Additionally, FilterTS outperformed models that capture inter-variable relationships using MLP (TimeMixer) and attention mechanisms (iTransformer), reducing average MSE by 3.69% and 2.02%, respectively. This suggests that the method of extracting and integrating shared frequency components across variables through filtering mechanisms is more effective than simpler fusion strategies. Furthermore, FilterTS consistently led FreTS across all tasks, underscoring that its performance benefits derive not solely from its construction in the frequency domain, but from its effective filtering strategies.

Model Analysis

Ablation Study To validate the effectiveness of the FilterTS module design, we conducted ablation studies on three datasets. Specifically, we evaluated the following variants: **w/o-SGF**: The Static Global Filtering Module was removed. **w/o-DCF**: The Dynamic Cross-Variable Filtering Module was removed. **w/o-SGF&DCF**: Both modules removed. **re-MLP**: The Dynamic Cross-Variable Filtering Module was replaced with a simple MLP to capture inter-variable relationships. **re-Attn**: The Dynamic Cross-Variable Filtering Module was replaced with an attention mechanism to capture inter-variable relationships.

Table 2 shows the results of the ablation study. We observe that removing the Static Global Filtering Module results in performance degradation of 5%, highlighting the importance of capturing dominant stable frequency components within the time series for long-term forecasting. Similarly, the removal of the Dynamic Cross-Variable Filtering Module leads to a performance drop of 2%, indicating the benefits of effectively leveraging inter-variable relationships.

Furthermore, replacing the Dynamic Cross-Variable Filtering Module with either an MLP or an attention mech-

Dataset	ETTm1		Weather		Electricity	
	MSE	MAE	MSE	MAE	MSE	MAE
FilterTS	0.385	0.396	0.244	0.274	0.180	0.271
w/o-SGF	0.392	0.399	0.250	0.276	0.197	0.282
w/o-DCF	0.390	0.400	0.250	0.278	0.184	0.274
w/o-SGF&DCF	0.405	0.407	0.267	0.286	0.208	0.286
re-MLP	0.387	0.397	0.248	0.274	0.204	0.297
re-Attn	0.392	0.398	0.254	0.281	0.181	0.273

Table 2: Ablation analysis of the FilterTS model, averaged across all prediction lengths for each dataset.

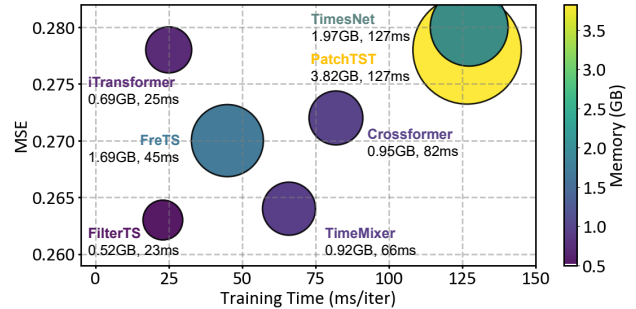


Figure 3: Performance Analysis of FilterTS: Assessing MSE, Training Time, and Memory Usage, evaluated on the Weather Dataset with a 96-In/336-Out Setup.

anism resulted in inferior performance. This suggests that selectively extracting shared frequency components across variables, rather than simply fusing inter-variable information, is a more effective approach for modeling inter-variable relationships in multivariate time series forecasting.

Model Efficiency

We compare the FilterTS model against other state-of-the-art models in terms of forecasting accuracy, memory usage, and training speed. The results, as shown in Figure 3, indicate that FilterTS outperforms the comparison models in predictive performance, achieves lower memory consumption, and faster training speed.

Conclusion

This paper introduced FilterTS, a novel multivariate time series forecasting model that enhances prediction accuracy and computational efficiency through comprehensive filtering techniques in the frequency domain. By incorporating both Static Global and Dynamic Cross-Variable Filtering Modules, FilterTS effectively captures essential frequency components. Our extensive experiments on eight real-world datasets demonstrate that FilterTS outperforms existing state-of-the-art methods in forecasting accuracy and efficiency. These results underscore the benefits of targeted frequency analysis in time series forecasting, suggesting that refining frequency components can significantly advance predictive capabilities.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Tianjin of China under Grant No. 21JCZDJC00740.

References

- Bui, K.-H. N.; Cho, J.; and Yi, H. 2022. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence*, 52(3): 2763–2774.
- Cai, W.; Liang, Y.; Liu, X.; Feng, J.; and Wu, Y. 2024. Ms-gnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 10, 11141–11149.
- Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; et al. 2020. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in Neural Information Processing Systems*, 33: 17766–17778.
- Duhamel, P.; and Vetterli, M. 1990. Fast Fourier transforms: a tutorial review and a state of the art. *Signal Processing*, 19(4): 259–299.
- Ekambaram, V.; Jati, A.; Nguyen, N.; Sinthong, P.; and Kalagnanam, J. 2023. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 459–469.
- Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; and Li, X. 2024. TSLANet: Rethinking transformers for time series representation learning. In *Forty-first International Conference on Machine Learning*.
- Huang, Q.; Shen, L.; Zhang, R.; Cheng, J.; Ding, S.; Zhou, Z.; and Wang, Y. 2024. Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11, 12608–12616.
- Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.-H.; and Choo, J. 2022. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.
- Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*.
- Luo, D.; and Wang, X. 2024. ModernTCN: A modern pure convolution structure for general time series analysis. In *The Twelfth International Conference on Learning Representations*.
- Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*.
- Olivares, K. G.; Challu, C.; Marcjasz, G.; Weron, R.; and Dubrawski, A. 2023. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 39(2): 884–900.
- Qiu, X.; Hu, J.; Zhou, L.; Wu, X.; Du, J.; Zhang, B.; Guo, C.; Zhou, A.; Jensen, C. S.; Sheng, Z.; and Yang, B. 2024. TFB: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proc. VLDB Endow.*, 17(9): 2363–2377.
- Shao, Z.; Zhang, Z.; Wang, F.; Wei, W.; and Xu, Y. 2022. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 4454–4458.
- Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and ZHOU, J. 2024. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations*.
- Wang, X.; Wang, Z.; Yang, K.; Feng, J.; Song, Z.; Deng, C.; et al. 2023. MPPN: Multi-resolution periodic pattern network for long-term time series forecasting. *arXiv preprint arXiv:2306.06895*.
- Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*.
- Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 753–763.
- Xu, Z.; Zeng, A.; and Xu, Q. 2024. FITS: Modeling time series with 10^k parameters. In *The Twelfth International Conference on Learning Representations*.
- Yi, K.; Zhang, Q.; Fan, W.; Wang, S.; Wang, P.; He, H.; An, N.; Lian, D.; Cao, L.; and Niu, Z. 2024. Frequency-domain MLPs are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36.
- Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, 9, 11121–11128.
- Zhang, Y.; and Yan, J. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*.
- Zhao, L.; and Shen, Y. 2024. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. In *The Twelfth International Conference on Learning Representations*.
- Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 12, 11106–11115.
- Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 27268–27286.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in Neural Information Processing Systems*, 36: 43322–43355.