information

Article

DFCNformer: A Transformer Framework for Non-Stationary Time-Series Forecasting Based on De-Stationary Fourier and Coefficient Network

Yuxin Jin [1], Yuhan Mao [2] and Genlang Chen [3,*]

[1] School of Computer Science and Technology (School of Artificial Intelligence), Zhejiang Sci-Tech University, Hangzhou 310018, China; 202230603069@mails.zstu.edu.cn
[2] School of Economics and Management, Zhejiang Sci-Tech University, Hangzhou 310018, China; maoyuhan@zjlgdx1.wecom.work
[3] School of Computer and Data Engineering, Ningbo Tech University, Ningbo 315199, China
* Correspondence: cgl@zju.edu.cn; Tel.: +86-133-0754-8555

Abstract: Time-series data are widely applied in real-world scenarios, but the non-stationary nature of their statistical properties and joint distributions over time poses challenges for existing forecasting models. To tackle this challenge, this paper introduces a forecasting model called DFCNformer (De-stationary Fourier and Coefficient Network Transformer), designed to mitigate accuracy degradation caused by non-stationarity in time-series data. The model initially employs a stabilization strategy to unify the statistical characteristics of the input time series, restoring their original features at the output to enhance predictability. Then, a time-series decomposition method splits the data into seasonal and trend components. For the seasonal component, a Transformer-based encoder–decoder architecture with De-stationary Fourier Attention (DSF Attention) captures temporal features, using differentiable attention weights to restore non-stationary information. For the trend component, a multilayer perceptron (MLP) is used for prediction, enhanced by a Dual Coefficient Network (Dual-CONET) that mitigates distributional shifts through learnable distribution coefficients. Ultimately, the forecasts of the seasonal and trend components are combined to generate the overall prediction. Experimental findings reveal that when the proposed model is tested on six public datasets, in comparison with five classic models it reduces the MSE by an average of 9.67%, with a maximum improvement of 40.23%.

Keywords: time-series prediction; non-stationary; attention mechanism; coefficient network; Transformer

Academic Editor: Zhigang Chu

Received: 29 November 2024
Revised: 27 December 2024
Accepted: 14 January 2025
Published: 17 January 2025

Citation: Jin, Y.; Mao, Y.; Chen, G. DFCNformer: A Transformer Framework for Non-Stationary Time-Series Forecasting Based on De-Stationary Fourier and Coefficient Network. *Information* **2025**, *16*, 62. https://doi.org/10.3390/info16010062

1. Introduction

Long-time-series forecasting (LTSF) has found extensive applications in areas such as energy demand, traffic flow, disease spread, and finance [1–5]. However, the non-stationary nature of time-series data—including continuity, seasonality, trends, and real-world noise—makes forecasting challenging. One distinct feature of non-stationary time series is the dynamic evolution of their statistical properties and joint distributions over time, which can be interpreted as distributional shifts. This shift hampers model generalization, significantly impacting forecasting performance.

Given the remarkable success that Transformers have achieved in natural language processing and their proficiency in effectively capturing long-range dependencies, forecasting models based on Transformers have recently attracted significant attention within the domain of LTSF [6]. Researchers have increasingly acknowledged the importance

of seasonal and trend decomposition in revealing underlying sequence patterns and enhancing forecasting accuracy [7,8]. This performance enhancement via decomposition is particularly consistent when applying Transformers to LTSF. Notably, studies [9,10] introduced decomposition methods, as well as strategies like self-correlation and frequency-domain-enhanced sparse attention mechanisms, to improve model performance in LTSF. However, their decomposition strategies do not fully separate trend and seasonal components; trend forecasting still relies on the attention module. Since the attention mechanism typically allocates weights by calculating correlations between each time step, it tends to focus more on local features and short-term dependencies, potentially overlooking global trend characteristics.

Currently, scholars have proposed various strategies to tackle non-stationary time series, with one of the most widely adopted methods being Reversible Instance Normalization (RevIN), introduced by Kim et al. [11]. This approach dynamically addresses non-stationarity by normalizing and denormalizing the input data, effectively adjusting the statistical properties of time series and mitigating issues caused by distributional shifts. While normalization strategies can enhance predictive performance, they may lead to "over-stabilization", especially on datasets sensitive to long-term trends and in combination with attention mechanisms, potentially diminishing the attention module's ability to detect specific time dependencies from the original non-stationary data [12]. This is because notable changes in a series, such as sharp jumps or slopes, are often due not merely to random noise but potentially to significant external events [13]. This suggests that, while reducing non-stationarity in time series can improve forecasting accuracy, preserving key dynamic features within the series remains a critical area for future research.

In order to further investigate the influence of non-stationarity on forecasting, this paper presents the DFCNformer model. This model combines stabilization preprocessing, Fourier transforms, time-series decomposition, and Transformer networks. It is characterized by two central modules: a module for forecasting the seasonal component and a module for forecasting the trend component. The primary contributions are listed below.

First, we introduce the DFCNformer forecasting model. This model has the feature of decomposing time-series data into distinct seasonal and trend components, thereby enabling a targeted and customized way to handle non-stationarity within each of these components. In contrast to existing models, the DFCNformer shows better performance in different time-series forecasting tasks, specifically by effectively alleviating the non-stationary issues present in both the seasonal and trend components.

Second, for the seasonal component, we employ a Transformer-based encoder–decoder architecture with a De-stationary Fourier Attention (DSF Attention) mechanism. This mechanism introduces a stabilization module that enhances convergence and predictive accuracy. Moreover, by incorporating a non-stationary factor within the Fourier attention mechanism, we address the issue of over-stabilization, preserving essential information such as short-term fluctuations and abrupt shifts that are often lost in the stabilization process.

Third, for the trend component, we employ a multilayer perceptron (MLP), a feed-forward neural network that approaches input–output mapping as a global optimization task. This allows us to capture smooth, gradual trend patterns without requiring complex dependency modeling. To address spatially internal and spatially external shifts caused by distributional drift within the trend component, we integrate a Dual Coefficient Net (DualCONET), which learns distribution coefficients between input and output spaces, providing the MLP with data that more closely approximate the true underlying distribution.

Fourth, our experiments' outcomes indicate that the DFCNformer reaches the state-of-the-art performance level across six real-world datasets. It persistently surpasses five traditional time-series forecasting models and demonstrates strong generalization capabilities.

The organization of this paper is as follows: In Section 2, a review of the related work in the field of time-series forecasting is carried out. Section 3 elaborates on the architecture of the DFCNformer model in detail. Section 4 delves into the experimental setup as well as the corresponding results. Finally, Section 5 draws the conclusions of this study.

2. Related Work

2.1. Time-Series Prediction Model

In the field of LTSF, research focuses on effectively capturing time dependencies and sequential characteristics. Traditional linear statistical models, such as ARIMA [14], transform non-stationary data into stationary sequences through differencing operations for modeling. This extends the methods applicable only to stationary time series to non-stationary time series. Due to their exceptional automatic feature extraction capabilities and powerful nonlinear modeling capacities, deep learning methods have been widely adopted in LSTF. Methods based on recurrent neural networks (RNNs) [15–17] have been proposed to apply autoregressive approaches to sequence modeling. Nevertheless, the inbuilt limitations of the recurrent network structure that they possess cause RNNs to be susceptible to the problems of vanishing and exploding gradients during long-term training. The dynamic nature of non-stationary time series further exacerbates these issues, leading to unstable training processes. This instability renders RNNs more vulnerable to sudden changes and noise, ultimately diminishing their generalization capabilities [18]. On the other hand, methods based on convolutional neural networks (CNNs) [19–21] utilize convolutional filters to capture local changes within a time window of the time-series data. Nevertheless, because of the localized characteristic of convolutional kernels, these models encounter difficulties when it comes to capturing long-term dependencies as well as global temporal patterns. This limitation becomes particularly pronounced when handling non-stationary sequences that involve trend shifts, periodic fluctuations, and noise interference, revealing shortcomings in their ability to process multi-scale features effectively [22].

In recent years, Transformer-based models have shown remarkable advantages in modeling long-term dependencies and multi-scale features. Their self-attention mechanism enables them to efficiently capture long-range interactions, making them particularly effective for complex temporal patterns. Although Zeng [23] and others have argued that linear models can outperform Transformer-based models in some scenarios, linear models are less capable of capturing non-stationarity and require more historical information for training, with poorer adaptability to different datasets. In contrast, the Transformer, with its strong nonlinear modeling capabilities, still holds considerable advantages in long-term sequence forecasting. The Reformer [24] model reduces the memory and computational complexity of the Transformer model when handling long sequences by introducing techniques such as Locality-Sensitive Hashing (LSH) Attention and Reversible Residual Networks. The Informer [25] introduced an efficient ProbSparse self-attention mechanism based on KL divergence, which enhances model prediction accuracy through layer-by-layer distillation and sparse self-attention processing. The Crossformer [26] introduced a Two-Stage Attention (TSA) mechanism combined with a segment embedding strategy to effectively capture cross-temporal and cross-dimensional dependencies, thereby mitigating errors induced by data fluctuations. Overall, these models mainly concentrate on refining the self-attention mechanism to improve their ability to manage complex and long-term dependencies effectively. The TDformer [27] improves the decomposition method based on Autoformer and FEDformer by adopting the approach of "detrending first, then focusing". It decomposes the time series into independent seasonal and trend components, combined with the RevIN [11] detrending preprocessing module and a frequency-domain attention mechanism for the seasonal component, allowing the model to better handle

non-stationarity in sequences. However, in the data preprocessing process of the above model, to avoid gradient explosions or vanishing, and to allow the model to train more stably and learn relationships between different time steps, the sequence data are normalized, scaling the data to a range of [0, 1]. Nevertheless, time series in the real world are intrinsically complex and non-stationary. Normalizing the data might unintentionally remove the crucial non-stationary features [12,28,29]. As a result, the ability of the attention mechanism to effectively capture and differentiate these essential patterns is weakened.

2.2. Strategies for Addressing Non-Stationarity in LTSF

The statistical attributes and joint distribution of non-stationary time series change as time progresses, which brings about diverse data distributions in different time periods or under various conditions. This discrepancy between the data distributions during model training and prediction, known as distribution shift, can significantly impact model performance [30]. In order to deal with the influence of non-stationarity on prediction precision, scholars have put forward a variety of strategies. RevIN [11] mitigates the non-stationarity by standardizing each data point using its mean and standard deviation, making the model easier to train. Fan et al. [31] analyzed intra-space and inter-space distribution shifts and noted that most existing studies focus primarily on intra-space shifts while neglecting inter-space shifts. In response, they proposed a universal neural paradigm to alleviate both types of distribution shifts. Ref. [32] introduced an adaptive normalization method based on a time-slice perspective, dynamically adjusting normalization parameters to ease the challenges posed by distribution shifts in time-series prediction. However, while these methods have improved model performance, they also weaken the discriminative power of the attention mechanism, leading to overly stabilized attention issues. To address this, Liu [12] and colleagues designed sequence stabilization modules and de-stabilization attention modules, innovatively combining the self-attention mechanism with stability adjustment techniques; although effective, this is primarily suitable for attention mechanism models.

To fully capture and model non-stationary multivariate long time series, this paper builds upon the decomposition approach used in TDformer, which separates the time series into independent seasonal and trend components. Different strategies are then applied to each component, combining the strengths of previously proposed prediction models and methods for addressing sequence non-stationarity, while addressing their respective limitations. In the seasonal component, a de-stabilized Fourier attention mechanism is designed, which incorporates the non-stable factors (inherent non-stationarities potentially removed by stabilization processes) excluded from the stabilization module. By dynamically adjusting attention weights and feature extraction methods, this attention mechanism effectively focuses on key changes in the sequence without being disrupted by stability strategies. In the trend component, to address the issue of mitigating intra-space shifts while neglecting inter-space shifts, as in TDformer using RevIN, a dual-coefficient network structure is used to mitigate the non-stationarity of intra-space and inter-space distribution shifts in data trends, with an MLP layer better training and predicting data trends. Ultimately, the predictions of the seasonal and trend components are combined. Then, following a de-stabilization strategy, the forecast output for the non-stationary time series is generated.

3. Proposed Method: DFCNformer

In this section, we present the proposed method for forecasting non-stationary time series. First, we provide an overview of the time-series problem under study. Following the problem description, we detail the architecture of the DFCNformer, with a focus on the DSF Attention for seasonal component forecasting and the Dual-CONET for trend component

forecasting. These components constitute the core of our method. Finally, we introduce the overall prediction process of the DFCNformer.

3.1. Problem Description

This study centers on forecasting non-stationary multivariate long time series. Suppose X stands for such a time series which has feature dimensions of $L \times C$. Here, L indicates the length of the sequence, while C represents the quantity of variables. We define X in the following way: $X = (x_1, x_2, \ldots, x_L)^T \in \mathbb{R}^{L \times C}$. Each x_l (where $1 \leq l \leq L$) is a C-dimensional vector that represents the values of all C variables at time l. That is to say, $x_l = (x_1^1, \ldots, x_l^c, \ldots, x_l^C) \in \mathbb{R}^C$, with x_l^c being the value of the c-th variable at time l. On the other hand, for each variable c (where $1 \leq c \leq C$), the series $x^c = (x_1^c, \ldots, x_l^c, \ldots, x_L^c)^T \in \mathbb{R}^T$ depicts its temporal development across all L time points.

The LTSF problem is centered around forecasting the future value at time $l + t$, which is likewise known as t-step prediction. Specifically, our objective is to predict $Y_{l+t} (t \in \mathbb{N}^+)$ by means of the following function, where Y represents the target variable to be predicted, which could be a forecast result or the future value of the time series at time $l + t$:

$$Y_{l+t} = f_1(X), \tag{1}$$

where $f_1(\cdot)$ is usually the nonlinear function that we intend to learn.

3.2. DFCNformer Framework

Given that the seasonality and trend are two fundamental components of non-stationary time series, incorporating them into the model design is essential [33]. Moreover, the accompanying distribution changes due to non-stationarity make deep forecasting more challenging, hence stabilization methods have been extensively explored and consistently adopted as preprocessing steps for deep model inputs [34]. When it comes to time series that have prominent trends, MLPs are excellent at capturing the progress of these trends, thus allowing for the effective learning of long-term variations. Conversely, for time series with clear seasonal patterns, frequency-domain attention proves to be more appropriate, since it can effectively capture high-frequency seasonal changes and enhance the model's capacity to comprehend and predict periodic fluctuations [35]. In this regard, we present the architectural design of the DFCNformer model (as depicted in Figure 1).

The main steps of the DFCNformer model are as follows: First, the time series is processed with a stationarization strategy to unify the statistical data of each input. The output is then transformed using the restored statistical data to enhance predictability, extracting non-stationary factors to be integrated with the subsequent attention mechanism in this step. Subsequently, a time-series decomposition module is employed to split the series into seasonal and trend components. Each of these components is then modeled through a customized strategy.

For the seasonal component, a de-stationary Fourier attention transformation is designed to capture significant non-stationary information in the seasonal component, including periodicity, seasonality, short-term variations, and abrupt changes. The non-stationary factors here are also used to mitigate over-stationarization within the attention mechanism. Fourier attention is selected because the Fourier transform converts time-series data from the time domain to the frequency domain, allowing the model to focus on global frequency patterns rather than being dependent on specific local time points. This aids the model in capturing periodic patterns and global structures within the sequence [36]. Moreover, Fourier transforms in the frequency domain are typically more robust to noise, enabling more effective identification of true cycles and trends, thereby improving the model's resilience to noise and its adaptability to non-stationary sequences [37].

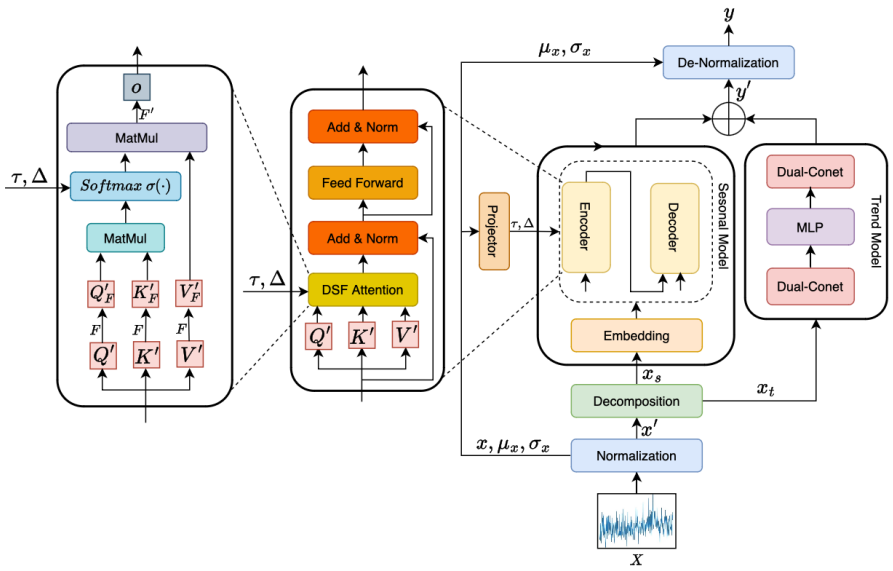

Figure 1. The overall framework of DFCNformer. Firstly, we implement a stabilization strategy for the time series. We make use of a decomposition module to split the series into seasonal and trend components. We utilize DSF Attention as the seasonal forecasting model, where the specific operation of the softmax $\sigma(\cdot)$ in this attention mechanism is given by softmax$(\cdot \times \tau + \Delta)$. For the trend forecasting model, we use an MLP and a coefficient network. The detailed structure of the coefficient network can be found in Section 3.4. The final prediction is generated by combining the outputs of the two models and applying inverse smoothing to restore the original data scale.

For the trend component, we utilize a dual coefficient network (Dual-CONET) and MLP for linear modeling. The dual coefficient network module mitigates distribution shifts in non-stationary trend components, allowing the MLP to more effectively extract and predict trend features. The final prediction is obtained by adding the seasonal and trend forecasts, followed by an inverse stationarization strategy. The advantage of DFCNformer resides in its effective combination of various modules, which makes it possible to accurately model the trends and seasonal changes in time series. This, in turn, considerably improves both the accuracy and dependability of its predictions.

The main modules of the proposed model are detailed in the following sections.

3.3. DSF Attention Fusion for Seasonal Composition

The Fourier frequency-domain attention mechanism (Fourier attention) has been shown to be one of the most effective methods for predicting data with fixed seasonality. With a computational complexity of $O(N \log N)$, it is well suited for handling large-scale time-series data efficiently. Therefore, this paper introduces improvements based on this attention mechanism.

The seasonal component X_s is fed through a linear layer to obtain $Q', K', V' \in \mathbb{R}^{L \times d_k}$, where d_k represents the corresponding temporal dimension. The basic equation for the attention mechanism is given below:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V \tag{2}$$

According to the derivation of the attention formula after applying the stationarization strategy in the non-stationary model, it can be obtained that the softmax matrix calculation of the stationary attention should include the non-stationary information of the standardization operation (see Section 3.5) such as σ_x and μ_Q, together with K, to approximate the original attention softmax matrix. The specific formula is given in Equation (3):

$$softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) = softmax\left(\frac{\sigma_x^2 Q'K'^T + 1\mu_Q^T K^T}{\sqrt{d_k}}\right), \tag{3}$$

where $\sigma_x \in \mathbb{R}^{C\times 1}$ represents the standard deviation of sequence x, and $\mu_Q \in \mathbb{R}^{d_k\times 1}$ is the average value of Q in the temporal dimension.

Since using only the stationary seasonal data would lose non-stationary information, such as σ_x, μ_Q, and K, in order for the attention mechanism to learn more data features, this paper introduces an improvement to the foundation of the attention mechanism by introducing non-stationary factors, i.e., $\tau = \sigma_x^2 \in \mathbb{R}^+$ and $\Delta = K\mu_Q \in \mathbb{R}^{L\times 1}$, resulting in the de-stationary Fourier attention mechanism (De-stationary Fourier Attention), as shown in Figure 1 on the left. Here, τ and Δ are shared by all layers in the de-stationary Fourier attention.

The current Q' and K' provide limited access to non-stationary information. The most logical source for additional non-stationary insights is the original, unnormalized x. Therefore, we should use a multilayer perceptron as the projection layer to extract τ and Δ from the non-stationary x, as detailed in Equation (4).

$$DSFAttention(Q', K', V', \tau, \Delta) = F^{-1}\left(softmax\left(\frac{\tau F(Q')F(K')^T + 1\Delta^T}{\sqrt{d_k}}\right)F(V')\right) \tag{4}$$

3.4. Dual-CONET for Trend Composition

Conventional normalization operations are ineffective in handling non-stationary data drift in trend components. To tackle this challenge, this paper utilizes a Dual-CONET and an MLP for prediction. By incorporating Dual-CONET into the normalization module, the input sequence is mapped to learnable distribution coefficients to alleviate the problem of distribution drift. The primary function of the integrated dual coefficient network is to map the input trend data into distribution coefficients, which are then used to process the input data. This processing ensures the data fall within a certain distribution range, making them more predictable after normalization. The structure of the trend component prediction model is shown in Figure 2.

The general formula of the coefficient network is as follows:

$$\varphi, \xi = Conet(x), \tag{5}$$

where φ represents the overall scale of the input data (level coefficient), and ξ represents the amplitude scale (amplitude coefficient) of the input data, which is the deviation of x relative to the mean φ. To mitigate the distribution shift issue between the input space and the output space, as well as the direct distribution shift within these spaces, two coefficient networks are employed. These networks are processed through a linear projection layer for normalization. The input sequence X_{input}, along with v_b^l and $v_p^l \in \mathbb{R}^{S\times N}$, represents the learnable weights of the two coefficient networks in the l-layer fully connected layer. The input coefficient network, *BackConet*, can be defined as shown in Equation (6).

$$\begin{cases} \varphi_{b,t}^{(i)},\ \xi_{b,t}^{(i)} = BackConet\left(X_{input}^{(i)}\right), \quad i = 1, \ldots, N, \quad X_{\text{input}^{(i)}=x_{t-S:t}^{(i)}} \\ \varphi_{b,t}^{(i)} = \sigma\left(\sum_{\tau=1}^{\dim(v_{b,i}^l)} v_{b,i\tau}^l x_{\tau-S+t}^{(i)}\right), \quad \xi_{b,t}^{(i)} = \sqrt{\mathbb{E}\left(x_t^{(i)} - \varphi_{b,t}^{(i)}\right)^2}, \\ x_{back}^{(i)} = \frac{\gamma}{\xi_{b,t}^{(i)}}\left(x_t^{(i)} - \varphi_{b,t}^{(i)}\right) + \beta \end{cases} \tag{6}$$

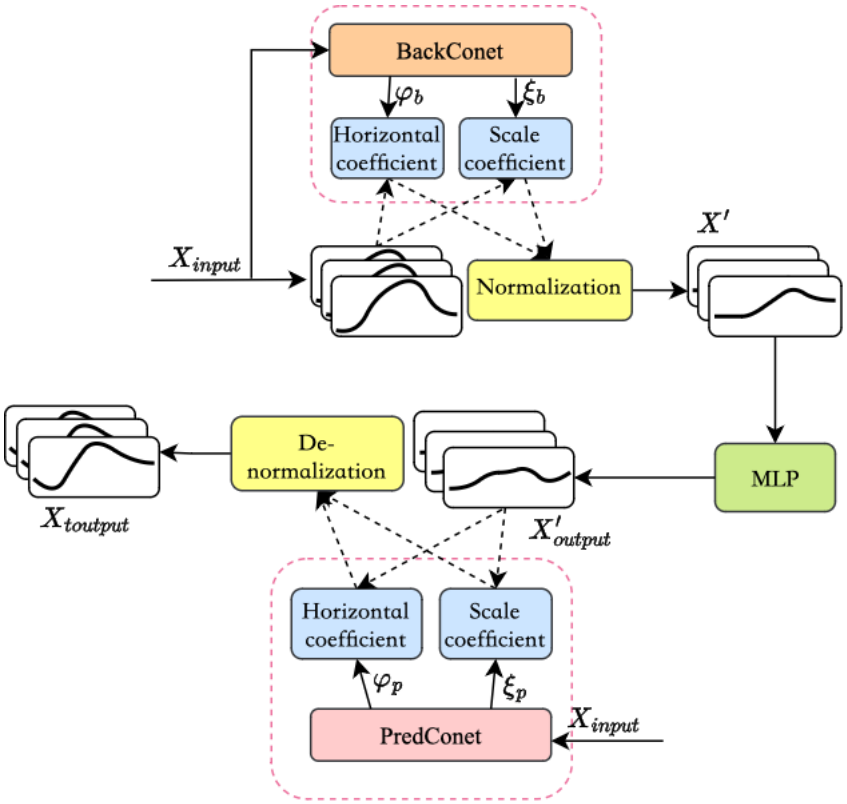

Figure 2. Trend component model (including Dual-CONET and MLP).

Here, t denotes the time step, S stands for the length of the input window, N is the quantity of time series, σ refers to the nonlinear leaky ReLU activation function, and γ and β are parameters that can be learned. The terms $\varphi_{b,t}^{(i)}$ and $\xi_{b,t}^{(i)}$ denote the horizontal and wave parameters of the input parameter network for input data. After processing by the input parameter network, X_{input} yields X_{back}, which is normalized to obtain X'. It is then input into the MLP layer for model training to improve the predictive performance of the MLP. When the MLP outputs the prediction result X'_{output}, the output parameter network is used to predict future components and returns the final normalized result. The formula for the output parameter network *PredConet* is given in Equation (7):

$$\begin{cases} \varphi_{p,t}^{(i)},\ \xi_{p,t}^{(i)} = PredConet\left(X_{input}^{(i)}\right), \quad i = 1, \ldots, N, \quad X_{input}^{(i)} = x_{t-S:t}^{(i)} \\ \varphi_{p,t}^{(i)} = \sigma\left(\sum_{\tau=1}^{\dim(v_{p,i}^{l})} v_{p,i\tau}^{l} x_{\tau-S+t}^{(i)}\right), \quad \xi_{p,t}^{(i)} = \sqrt{\mathbb{E}\left(x_t^{(i)} - \varphi_{p,t}^{(i)}\right)^2}, \\ x'^{(i)}_{output} = \dfrac{\gamma}{\xi_{p,t}^{(i)}}\left(x_t^{(i)} - \varphi_{p,t}^{(i)}\right) + \beta \end{cases} \tag{7}$$

$\varphi_{p,t}^{(i)}$ and $\xi_{p,t}^{(i)}$ represent the level and amplitude coefficients predicted by the output coefficient network model. X'_{output} is obtained after applying the inverse normalization operation to X_{output}. Although the two coefficient networks share the same input, they serve distinct purposes: *BackConet* is designed to approximate the distribution of the input data $X_{input}^{(i)}$, whereas *PredConet* leverages the input data to predict the future distribution.

3.5. Overall Forecasting Process

Current attention models primarily focus on interpolating historical data in a given context, which presents limitations in inferring linear trends. In contrast, MLPs have demonstrated more accurate performance in trend forecasting [38]. Building on this frame-

work, our proposed model comprises two parts: a seasonal part and a trend part. Each of them is specially designed to learn and predict its corresponding portion of the time series. The final prediction outcome is acquired by combining the predictions of both components. The model architecture is shown in Figure 3.

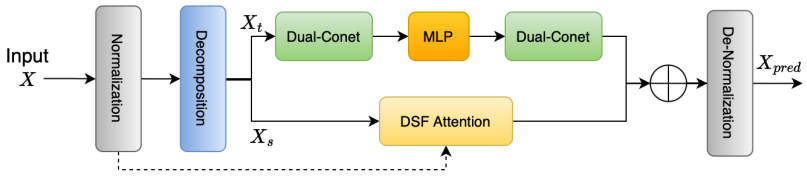

Figure 3. Overall forecasting process.

First, we perform normalization on the temporal dimension using a sliding window, and after prediction, we restore the original data characteristics through an inverse normalization operation. The normalization module can be represented as

$$\mu_x = \frac{1}{L}\sum_{i=1}^{L} x_i, \quad \sigma_x^2 = \frac{1}{L}\sum_{i=1}^{L}(x_i - \mu_x)^2, \quad x_i' = \frac{1}{\sigma_x} \odot (x_i - \mu_x), \tag{8}$$

where $\mu_x, \sigma_x \in \mathbb{R}^{C\times 1}$ and $\frac{1}{\sigma_x}$ represent the mean and standard deviation vectors, respectively. The normalization module minimizes deviations among the input time series, resulting in a more stable input distribution for the model.

The inverse normalization module is represented by the following equation:

$$\mathbf{y}' = \mathcal{H}(\mathbf{x}'), \quad \hat{y}_i = \sigma_x \odot y_i' + \mu_x \tag{9}$$

After that, we break down the time series into its seasonal and trend components. The implementation requires the application of multiple moving average filters with different sizes to capture diverse trend patterns. These patterns are then integrated into the final trend component by means of adaptive weights. Subsequently, the seasonal component is obtained by subtracting the trend component from the original time series:

$$x_t = \sigma(\omega(x)) * f(x), \quad x_s = x - x_t, \tag{10}$$

where σ represents the *softmax* operation, $\omega(x)$ denotes the data-related weights, and $f(x)$ represents the moving average filter.

For the seasonal component, we adopt the Transformer encoder–decoder architecture. We design the DSF Attention (see Section 3.3 for module design details) to address the instability in attention learning for the seasonal component. The seasonal component is first processed through an M-layer encoder, as follows:

$$\begin{cases} S_{en}^{m,1} = DSFAttention(x_{en}^{m-1}) \\ S_{en}^{m,2} = Add\&Norm(S_{en}^{m,1}) \\ S_{en}^{m,3} = FeedForward(S_{en}^{m,2}) \\ x_{en}^{m} = Add\&Norm(S_{en}^{m,3}) \end{cases} \tag{11}$$

In the above equations, $S_{en}^{m,i}$, where $i \in \{1,2,3\}$, denotes the intermediate variables within the i-th module of the m-th layer encoder.

Similarly, the seasonal component is zero-padded for the future time steps and then passed through the N-layer decoder, as follows:

$$\begin{cases} S_{de}^{n,1} = DSFAttention(x_{de}^{n-1}) \\ S_{de}^{n,2} = Add\&Norm(S_{sde}^{n,1}) \\ S_{de}^{n,3} = DSFAttention(S_{sde}^{n,2}) \\ S_{en}^{n,4} = Add\&Norm(S_{en}^{n,3}) \\ S_{de}^{n,5} = FeedForward(S_{sde}^{n,4}) \\ x_{de}^{n} = Add\&Norm(S_{de}^{n,5}) \end{cases} \tag{12}$$

In the above equations, $S_{\mathrm{de}}^{n,i}$, where $i \in \{1,2,\ldots,5\}$, represents the intermediate variables in the i-th module of the n-th layer decoder.

For the trend component, we use a three-layer MLP to forecast future trends. Meanwhile, to alleviate distribution shift issues caused by non-stationary phenomena in the trend component that cannot be resolved by normalization operations, we add a Dual-CONET module before and after the MLP (see Section 3.4 for detailed architecture):

$$X_t = DualConet(MLP(DualConet(x_t))) \tag{13}$$

By combining the seasonal forecast results from the Transformer and the trend forecast results from the MLP and then applying inverse normalization, we obtain the final output forecast.

4. Experiment and Result Analysis

4.1. Datasets

To validate the effectiveness of the model, this paper selects six publicly available multivariate time-series datasets for experimentation. These datasets are as follows:

1. **ETTh2 [26]**: Records hourly power loads of six substations and one oil temperature feature in a county in China from July 2016 to July 2018.
2. **Exchange [39]**: Covers daily exchange rates of eight different countries from 1990 to October 2010.
3. **Traffic [40]**: Records hourly lane occupancy rates from 862 different sensors on the I-80 highway in the Bay Area, California, from July 2016 to July 2018.
4. **Weather [41]**: It encompasses 21 meteorological indicators that were recorded every 10 min during the whole year of 2020 in Germany.
5. **ILI [42]**: Contains weekly records of influenza patient numbers in the United States from 2002 to June 2020.
6. **Citypower [43]**: Records weather conditions every 10 min and power consumption in three power distribution networks throughout 2017 in Dusseldorf.

The basic information about the datasets is summarized in Table 1. We follow the standard protocol, dividing each dataset into training, validation, and testing sets in a 7:1:2 ratio according to the chronological order.

Table 1. Basic information of the datasets.

Dataset	Sampling Frequency	Dimensions	Timesteps
ETTh2	Hourly	7	17,420
Exchange	Daily	8	7588
Traffic	Hourly	862	17,544
Weather	10 min	21	52,696
ILI	Weekly	7	966
Citypower	10 min	8	52,416

4.2. Computational Resources and System Setup

All experiments were conducted based on the PyTorch 2.1.2 framework on an NVIDIA GeForce RTX 4090 GPU with CUDA version 12.2. The code was written in Python 3.9.19. To verify the effectiveness of the proposed model, TDformer [27], FEDformer [10], Autoformer [9], Informer [25], and DLinear [23] were selected as baseline models for comparative analysis. Each model was trained for a maximum of 20 epochs, with early stopping based on the best performance on the validation set to avoid overfitting and improve computational efficiency. The model that performed best on the validation set during training was selected for final evaluation on the test set. In addition, dropout was applied with a rate of 0.05 to prevent overfitting on the training set and enhance the model's generalization ability. The Adam optimizer was utilized with a batch size of 32. The initial learning rate for DLinear was set to 0.05, while for other models it was initialized at 0.0001 and gradually decreased during training. This paper adopts mean square error (MSE) and mean absolute error (MAE) as evaluation metrics, where lower values reflect higher prediction accuracy.

4.3. Main Results

To ensure a fair comparison with other models, the input historical length for all models in this study was fixed at 96. Predictions were made for 96, 192, 336, and 720 timesteps on six datasets (for the ILI dataset, predictions were made for 24, 36, 48, and 60 timesteps). The experimental results are presented in Table 2, with the lowest MSE and MAE values for each dataset highlighted in bold.

In 22 out of 24 configurations with different datasets and prediction lengths, DFCNformer outperforms other benchmark models. As shown in Table 2, the MSE of our model is, on average, reduced by 9.67% compared to other benchmarks, with a maximum reduction of 40.23%. In the configurations where DFCNformer is the best, the minimum reduction is also 2.39%. Similarly, the MAE is reduced by an average of 6.22%, with a maximum reduction of 17.72%, and the minimum reduction is 0.99%. Compared with TDformer, a Transformer-based model with relatively superior predictive performance, and the linear model DLinear, DFCNformer's MSE is reduced by an average of 14.14% and 13.52%, respectively. This improvement is primarily attributed to the enhancement of the attention mechanism, using the instability factor and the mitigation of trend distribution shifts achieved by the Dual-CONET. Although DLinear demonstrated excellent predictive performance in two configurations, its performance was more modest in other configurations (e.g., steps 96 to 336 of Exchange, and steps 96 and 720 of Traffic), indicating that the model's relatively simple structure may limit its predictive stability across different datasets and prediction lengths.

Table 2. Prediction results of multivariate time-series models under different forecast horizons.

Dataset		DFCNformer		TDformer		FEDformer		Autoformer		Informer		DLinear	
		MSE	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
Traffic	96	**0.559**	**0.327**	0.575	0.335	0.572	0.358	0.671	0.429	0.741	0.414	0.649	0.396
	192	**0.571**	**0.328**	0.602	0.358	0.618	0.390	0.615	0.388	0.764	0.427	0.598	0.370
	336	**0.583**	**0.332**	0.616	0.348	0.622	0.384	0.607	0.375	0.847	0.473	0.605	0.373
	720	**0.612**	**0.341**	0.627	0.351	0.643	0.396	0.706	0.418	0.966	0.541	0.646	0.395
Exchange	96	**0.087**	**0.206**	0.091	0.210	0.151	0.281	0.147	0.278	0.908	0.774	0.098	0.217
	192	**0.176**	**0.298**	0.184	0.305	0.276	0.382	0.597	0.559	1.101	0.834	0.217	0.338
	336	**0.325**	**0.413**	0.356	0.431	0.445	0.490	0.462	0.508	1.618	1.017	0.420	0.469
	720	0.829	0.685	0.881	0.706	1.133	0.819	1.099	0.814	2.920	1.410	**0.742**	**0.651**
Weather	96	**0.176**	**0.219**	0.186	0.225	0.244	0.332	0.262	0.330	0.389	0.447	0.201	0.266
	192	**0.219**	**0.259**	0.233	0.267	0.308	0.368	0.311	0.371	0.443	0.458	0.236	0.293
	336	**0.261**	**0.301**	0.291	0.304	0.602	0.552	0.350	0.385	0.575	0.534	0.283	0.335
	720	0.357	**0.351**	0.367	0.353	0.407	0.418	0.422	0.432	1.095	0.776	**0.348**	0.383
ILI	24	**2.211**	**0.983**	2.889	1.124	2.849	1.180	3.380	1.290	5.257	1.616	2.403	1.097
	36	**2.087**	**0.960**	2.922	1.075	2.746	1.149	3.460	1.310	5.530	1.677	2.385	1.095
	48	**2.129**	**0.992**	2.843	1.068	2.731	1.128	3.130	1.200	5.537	1.646	2.349	1.089
	60	**2.360**	**1.056**	2.999	1.106	2.802	1.136	2.860	1.470	5.704	1.685	2.405	1.109
ETTh2	96	**0.207**	**0.314**	0.312	0.361	0.344	0.383	0.356	0.401	2.845	1.335	0.329	0.380
	192	**0.250**	**0.353**	0.430	0.429	0.435	0.442	0.533	0.505	6.197	2.070	0.431	0.443
	336	**0.279**	**0.368**	0.444	0.447	0.485	0.479	0.461	0.472	5.225	1.934	0.459	0.462
	720	**0.321**	**0.396**	0.458	0.470	0.468	0.479	0.459	0.476	3.689	1.622	0.774	0.631
Citypower	96	**0.205**	**0.279**	0.244	0.311	0.278	0.374	0.312	0.395	0.404	0.494	0.239	0.311
	192	**0.251**	**0.304**	0.271	0.323	0.279	0.360	0.446	0.489	0.528	0.571	0.273	0.346
	336	**0.302**	**0.336**	0.324	0.355	0.319	0.389	0.453	0.487	0.644	0.627	0.308	0.380
	720	**0.349**	**0.365**	0.373	0.382	0.459	0.499	0.504	0.523	0.817	0.701	0.350	0.426

Overall, DFCNformer is better at capturing relationships between data and demonstrates strong robustness and generalization capability in long-term time-series forecasting. Figure 4 presents a comparison of the predictions made by DFCNformer, TDformer, and DLinear on the ETTh2 and ILI datasets. An input sequence length of 96 is selected to predict the next 96 time steps (60 time steps for ILI). The dark gray curve depicts the real data, while the light gray curve illustrates the predicted results. As shown in the figure, our model exhibits smaller errors when handling data spikes, abrupt changes, and predicting trend directions.

The proposed model is also compared with Transformer-based models in terms of computational speed, parameter count, and predictive performance to assess its overall efficiency. Figure 5 presents a comparison using the Exchange dataset, with an input length of 96 and an output length of 192. The size of the circles represents the parameter count, where smaller circle areas, closer to the origin, indicate fewer parameters, faster computation, and lower error.

In terms of computation speed and parameter count, FEDformer has the largest number of parameters and the slowest computation speed (12.99 s), while DLinear has the fewest parameters and the fastest computation speed (0.689 s) among all models. DFCNformer, on the other hand, achieves the lowest MSE, outperforming the second-best model (TDformer) by 4.55%. Among Transformer-based models, DFCNformer demonstrates a speed comparable to Informer and TDformer, only 0.2 s slower than the fastest Informer. However, Informer shows relatively larger prediction errors. In terms of parameter count, FEDformer has the highest, while Autoformer has the lowest. Our model has only 3.94% more parameters than Autoformer. It is important to note that DLinear, due to its linear nature, achieves relatively good performance at a lower computational cost. However, its generalization capability is limited compared to DFCNformer, and it performs poorly on certain datasets (such as Traffic and ETTh2). This indicates that, while DLinear offers advantages in computational efficiency, it may fail to capture complex patterns in data with

nonlinear relationships or large-scale datasets. Nevertheless, the design concept of DLinear provides valuable insights for our future research, particularly in exploring how to achieve better prediction performance with limited computational resources.

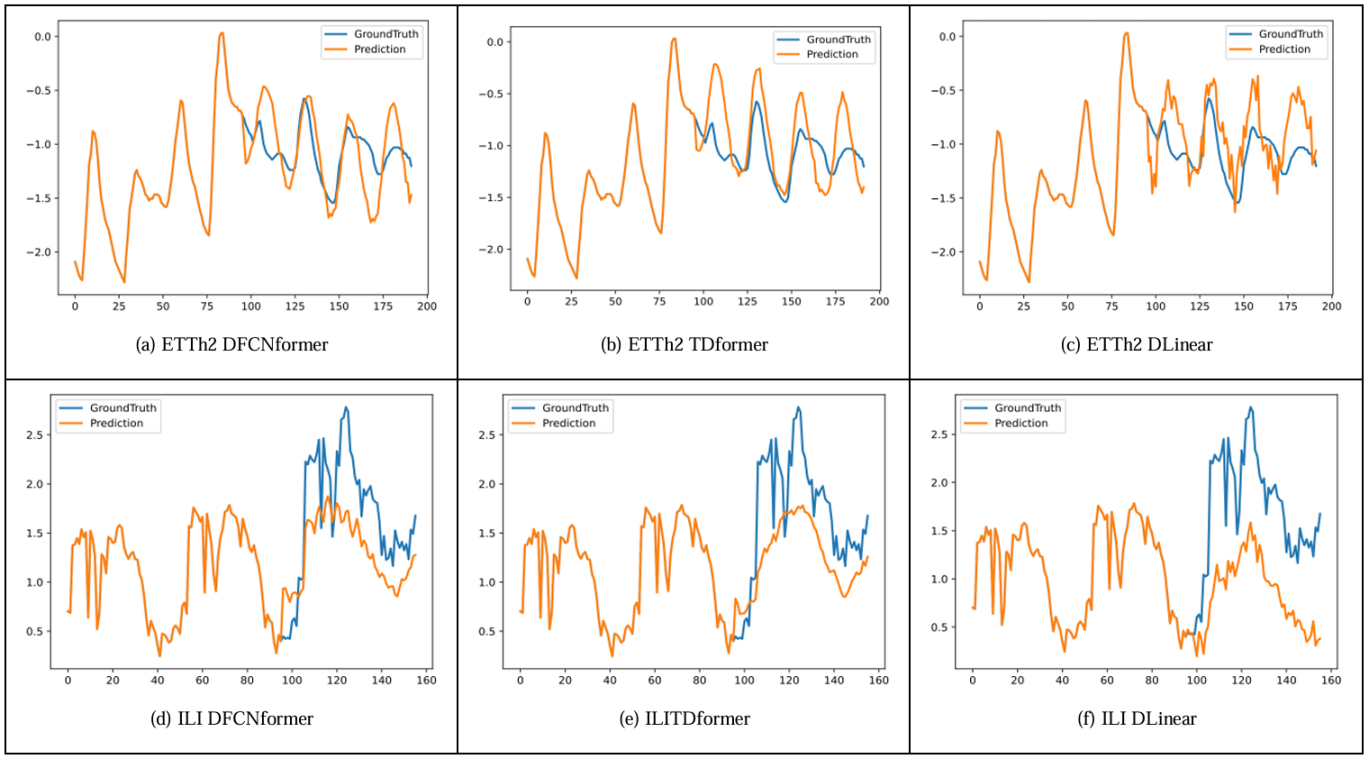

Figure 4. Comparison of predictions among DFCNformer, TDformer, and DLinear on ETTh2 and ILI.

Overall, DFCNformer outperforms other Transformer-based models in terms of average performance. Moreover, compared to DLinear, it adapts well to the characteristics of different datasets, demonstrating strong generalization across data from various domains.

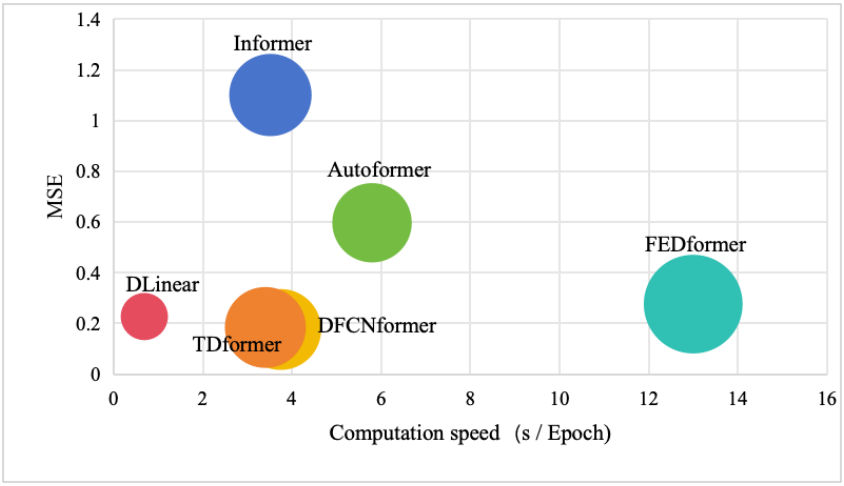

Figure 5. Comparison of efficiency across models.

4.4. Ablation Study

To independently assess the contributions of the stabilization–destabilization strategy, seasonal component, and trend component, an ablation study was conducted using the Traffic dataset, characterized by clear seasonality, and the Weather dataset, which exhibits a distinct trend. The results are presented in Table 3, with the lowest MSE and MAE values for each dataset highlighted in bold.

Table 3. The MSE and MAE results from our model's ablation study are reported. The variant of the model that specifically employs Fourier-based attention is referred to as DFCNformer-FA. DFCNformer-FA-DSFA replaces the MLP with Fourier attention (FA) to capture trends. DFCNformer-MLP-FA uses Fourier attention to capture seasonal patterns. DFCNformer w/o Dual-CONET removes the Dual-CONET module for trend modeling, and DFCNformer w/o Norm. removes the sequence stabilization-destabilization module. Average increase represents the average percentage increase in MSE/MAE compared to DFCNformer.

Method	Metric	Traffic				Average	Weather				Average
		96	192	336	720	Increase	96	192	336	720	Increase
DFCNformer	MSE	**0.559**	**0.571**	**0.583**	**0.612**	-	**0.176**	**0.219**	**0.261**	**0.357**	-
	MAE	**0.327**	**0.328**	**0.332**	**0.341**	-	**0.219**	**0.259**	**0.301**	**0.351**	-
DFCNformer-FA	MSE	0.659	0.666	0.679	0.707	16.70%	0.301	0.372	0.412	0.517	58.50%
	MAE	0.348	0.359	0.371	0.413	12.35%	0.332	0.384	0.420	0.515	45.94%
DFCNformer-FA-DSFA	MSE	0.603	0.612	0.609	0.607	4.65%	0.181	0.223	0.265	0.361	1.98%
	MAE	0.333	0.346	0.343	0.342	2.71%	0.234	0.271	0.332	0.358	5.65%
DFCNformer-MLP-FA	MSE	0.577	0.600	0.619	0.632	4.48%	0.182	0.234	0.288	0.374	6.72%
	MAE	0.339	0.350	0.355	0.356	5.42%	0.243	0.297	0.338	0.398	12.72%
DFCNformer w/o Dual-CONET	MSE	0.565	0.576	0.586	0.613	0.69%	0.193	0.242	0.301	0.378	10.28%
	MAE	0.330	0.339	0.344	0.345	2.41%	0.231	0.274	0.307	0.630	27.56%
DFCNformer w/o Norm.	MSE	0.589	0.612	0.621	0.644	6.20%	0.197	0.248	0.306	0.379	11.86%
	MAE	0.337	0.342	0.351	0.355	4.22%	0.253	0.303	0.341	0.404	14.84%

As shown in Table 3, the model using only Fourier attention performs the worst on complex time-series forecasting, with average MSE increases of 16.70% and 58.50% and average MAE increases of 12.35% and 45.94% on the Traffic and Weather datasets, respectively, indicating the necessity of decomposing time series and applying independent predictive measures to each component. Furthermore, replacing the MLP with Fourier attention in the trend component (DFCNformer-FA-DSFA) results in an increase in MSE across all cases, which underlines the poorer generalization competence of the Fourier attention model in dealing with trend data.

Additionally, the models without the instability factor in the seasonal attention component (DFCNformer-MLP-FA) and without the Dual-CONET module in the trend component (DFCNformer w/o Dual-CONET) show average MSE increases of 4.48% and 0.99% on Traffic, and 6.72% and 10.28% on Weather, respectively. This highlights the importance of introducing the stabilization factor in the seasonal component and the coefficient network in the trend component for handling non-stationary data. For the strongly seasonal Traffic dataset, the model without the instability factor structure experiences a greater increase in MSE compared to the model without the dual-coefficient network structure. Conversely, for the trend-dominated Weather dataset, the model without the dual-coefficient network structure sees a greater MSE increase than the model without the instability factor structure. This may be due to the enhanced role of the stabilization attention mechanism in capturing seasonal changes and abrupt patterns in highly seasonal datasets, while the dual-coefficient network better learns non-stationary trend patterns in strongly trend-oriented datasets.

DFCNformer w/o Norm. is based on DFCNformer-MLP-FA but excludes the stabilization and de-stabilization strategies. Figure 6 presents the prediction comparison for a forecasting horizon of 96 steps on the Traffic (Figure 6a–c) and Weather (Figure 6d–f) test datasets. From the subfigures, it can be observed that the prediction curves in the DFCNformer-MLP-FA (Figure 6a,d) model fit the actual curves more closely. This may be due to the fact that the stabilization and de-stabilization strategies effectively remove noise and short-term fluctuations in the data, making the error distribution more concentrated and stable, allowing the model to more accurately capture long-term trends and key patterns. After removing the de-stabilization structure (Figure 6b,e), the model's predictive performance significantly declines, with a large deviation from the actual curve. This is because the predictions are not promptly corrected, failing to effectively follow the

distribution of the input sequence. Subsequently, after removing the stabilization structure (Figure 6c,f), the model's predictions exhibit large errors and significant fluctuations in the data. This is due to the model's difficulty in effectively learning the relevant features of non-stationary sequences, leading to reduced generalization capability.

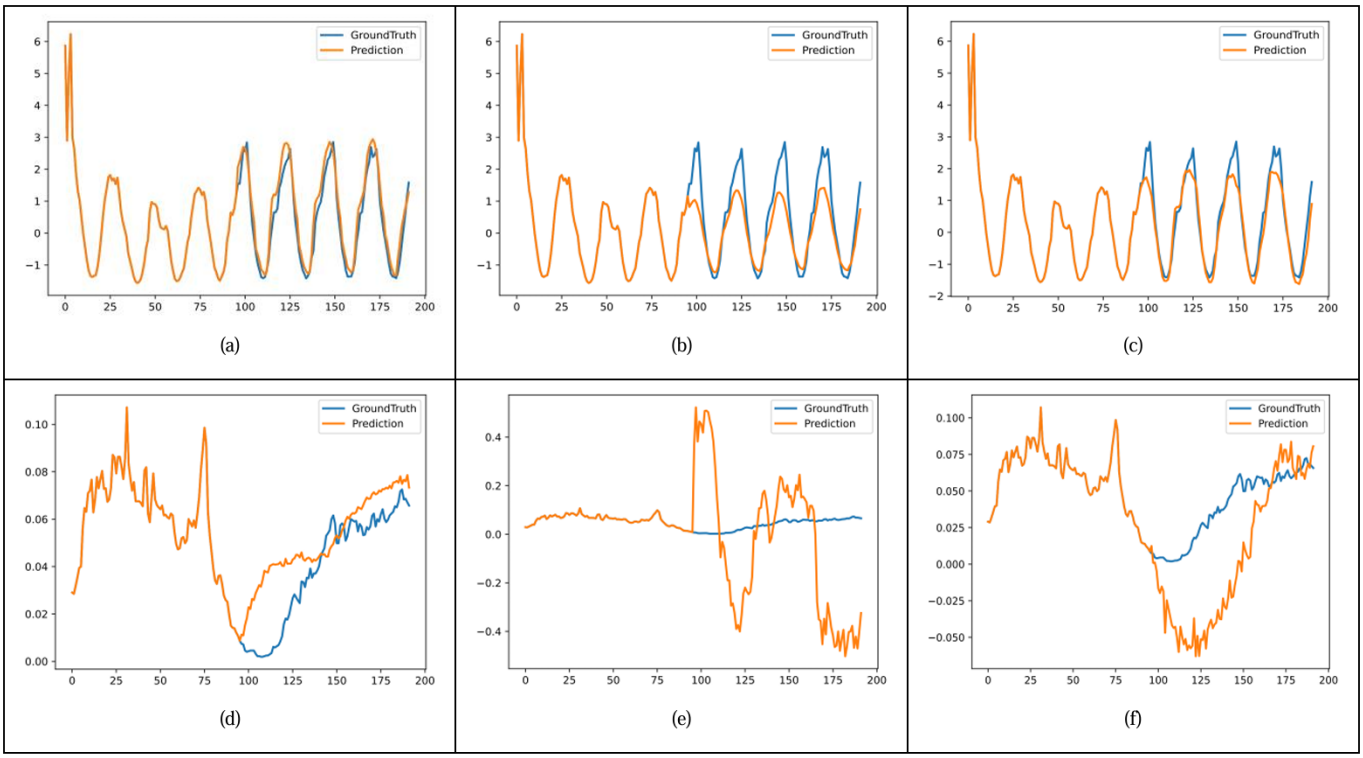

Figure 6. Effectiveness comparison of stabilization and de-stabilization strategies of 96 steps on Traffic and Weather datasets. (**a**–**c**) show the comparison for the Traffic dataset, while (**d**–**f**) display the results for the Weather dataset. (**a**,**b**) illustrate the prediction results of the DFCNformer-MLP-FA model, whereas (**b**,**e**) show the prediction results of the same model with the de-stabilization strategy removed. (**c**,**f**) represent the predictions of DFCNformer w/o Norm., where both the stabilization and de-stabilization strategies are excluded.

5. Conclusions

This paper introduces the DFCNformer model, which improves the forecasting performance of non-stationary time series by addressing their inherent non-stationarity. The model uses a stabilization strategy and decomposes the time series into seasonal and trend components. The seasonal component is handled using a de-stationary Fourier attention mechanism, while the trend component uses a dual-coefficient network and MLP to predict long-term trends. The results of the seasonal and trend components are then combined, and the final non-stationary time-series forecast is produced through a de-stabilization strategy. The experimental results on six public datasets demonstrate that DFCNformer outperforms other benchmark models in prediction accuracy, successfully mitigating errors caused by non-stationarity and exhibiting strong generalization capabilities.

Despite its strengths, the model's performance is sensitive to the quality and completeness of historical data. In cases of insufficient data, prediction accuracy may be compromised. Additionally, the model's computational cost could be improved. Future work will focus on enhancing the model's predictive efficiency in scenarios with limited or incomplete data, while also exploring simpler and more efficient architectures to reduce computational costs.

Author Contributions: Conceptualization, G.C.; methodology, Y.J.; formal analysis, Y.J. and Y.M.; writing—original draft preparation Y.J.; experiment Y.J. and Y.M.; project administration, G.C. and Y.J.; validation Y.J.; writing—review and editing, G.C. and Y.M.; resources G.C.; data curation Y.M. and Y.J.; supervision, G.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Ningbo Science and Technology Major Project (2024Z259).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in the paper can be found at the following link: ETTh2: https://doi.org/10.1609/aaai.v35i12.17325. Exchange: https://doi.org/10.1145/3209978.3210006. Traffic: http://pems.dot.ca.gov/(accessed on 14 May 2024). Weather: https://www.bgc-jena.mpg.de/wetter/ (accessed on 14 May 2024). ILI: https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html (accessed on 14 May 2024). Citypower: https://kaggle.com/datasets/fedesoriano/electric-power-consumption (accessed on 14 May 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, H.; Li, J.; Chang, L. Predicting Time Series Energy Consumption Based on Transformer and LSTM. In *International Conference on 6GN for Future Wireless Networks*; Springer Nature: Cham, Switzerland, 2023; pp. 299–314.
2. Chen, J.; Zheng, L.; Hu, Y.; Wang, W.; Zhang, H.; Hu, X. Traffic flow matrix-based graph neural network with attention mechanism for traffic flow prediction. *Inf. Fusion* **2024**, *104*, 102146. [CrossRef]
3. Chen, K.; Liu, Y.; Ji, T.; Yang, G.; Chen, Y.; Yang, C.; Zheng, Y. TEST-Net: Transformer-enhanced Spatio-temporal network for infectious disease prediction. *Multimedia Syst.* **2024**, *30*, 312. [CrossRef]
4. He, K.; Yang, Q.; Ji, L.; Pan, J.; Zou, Y. Financial time series forecasting with the deep learning ensemble model. *Mathematics* **2023**, *11*, 1054. [CrossRef]
5. Hui, G.; Chen, S.; He, Y.; Wang, H.; Gu, F. Machine learning-based production forecast for shale gas in unconventional reservoirs via integration of geological and operational factors. *J. Nat. Gas Sci. Eng.* **2021**, *94*, 104045. [CrossRef]
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
7. Bandara, K.; Bergmeir, C.; Hewamalage, H. LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1586–1599. [CrossRef]
8. Wen, Q.; Zhang, Z.; Li, Y.; Sun, L. Fast RobustSTL: Efficient and robust seasonal-trend decomposition for time series with complex patterns. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 2203–2213.
9. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22419–22430.
10. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. *Int. Conf. Mach. Learn.* **2022**, *162*, 27268–27286.
11. Kim, T.; Kim, J.; Tae, Y.; Park, C.; Choi, J.H.; Choo, J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In Proceedings of the International Conference on Learning Representations, Virtual, 25–29 April 2022.
12. Liu, Y.; Wu, H.; Wang, J.; Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9881–9893.
13. Wang, S.; Li, C.; Lim, A. A model for non-stationary time series and its applications in filtering and anomaly detection. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11. [CrossRef]
14. Ariyo, A.A.; Adewumi, A.O.; Ayo, C.K. Stock price prediction using the ARIMA model. In Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, UK, 26–28 March 2014; pp. 106–112.
15. Yu, R.; Zheng, S.; Anandkumar, A.; Yue, Y. Long-term forecasting using tensor-train rnns. In Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 Apri–3 May 2018.
16. Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. [CrossRef]
17. Abbasimehr, H.; Paki, R. Improving time series forecasting using LSTM and attention models. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 673–691. [CrossRef]

18. Casolaro, A.; Capone, V.; Iannuzzo, G.; Camastra, F. Deep learning for time series forecasting: Advances and open problems. *Information* **2023**, *14*, 598. [CrossRef]
19. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
20. Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; Xu, Q. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 5816–5828.
21. Wan, R.; Tian, C.; Zhang, W.; Deng, W.; Yang, F. A multivariate temporal convolutional attention network for time-series forecasting. *Electronics* **2022**, *11*, 1516. [CrossRef]
22. Ubal, C.; Di-Giorgi, G.; Contreras-Reyes, J.E.; Salas, R. Predicting the long-term dependencies in time series using recurrent artificial neural networks. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1340–1358. [CrossRef]
23. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are transformers effective for time series forecasting? *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 11121–11128. [CrossRef]
24. Kitaev, N.; Kaiser, Ł.; Levskaya, A. Reformer: The efficient transformer. *arXiv* **2020**, arXiv:2001.04451.
25. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 11106–11115. [CrossRef]
26. Zhang, Y.; Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In Proceedings of the Eleventh International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
27. Zhang, X.; Jin, X.; Gopalswamy, K.; Gupta, G.; Park, Y.; Shi, X.; Wang, H.; Maddix, D.C.; Wang, Y. First de-trend then attend: Rethinking attention for time-series forecasting. *arXiv* **2022**, arXiv:2212.08151.
28. Ogasawara, E.; Martinez, L.C.; de Oliveira, D.; Zimbrao, G.; Pappa, G.L.; Mattoso, M. Adaptive normalization: A novel data normalization approach for non-stationary time series. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010; pp. 1–8.
29. Lim, B.; Zohren, S. Time-series forecasting with deep learning: A survey. *Philos. Trans. R. Soc.* **2021**, *379*, 20200209. [CrossRef]
30. Koh, P.W.; Sagawa, S.; Marklund, H.; Xie, S.M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R.L.; Gao, I.; et al. Wilds: A benchmark of in-the-wild distribution shifts. *Int. Conf. Mach. Learn.* **2021**, *139*, 5637–5664.
31. Fan, W.; Wang, P.; Wang, D.; Wang, D.; Zhou, Y.; Fu, Y. Dish-ts: A general paradigm for alleviating distribution shift in time series forecasting. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*. 7522–7529. [CrossRef]
32. Liu, Z.; Cheng, M.; Li, Z.; Huang, Z.; Liu, Q.; Xie, Y.; Chen, E. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 14273–14292.
33. Oreshkin, B.N.; Carpov, D.; Chapados, N.; Bengio, Y. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv* **2019**, arXiv:1905.10437.
34. Passalis, N.; Tefas, A.; Kanniainen, J.; Gabbouj, M.; Iosifidis, A. Deep adaptive input normalization for time series forecasting. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3760–3765. [CrossRef]
35. Wan, J.; Xia, N.; Yin, Y.; Pan, X.; Hu, J.; Yi, J. TCDformer: A transformer framework for non-stationary time series forecasting based on trend and change-point detection. *Neural Netw.* **2024**, *173*, 106196. [CrossRef]
36. Yi, K.; Zhang, Q.; Cao, L.; Wang, S.; Long, G.; Hu, L.; He, H.; Niu, Z.; Fan, W.; Xiong, H. A survey on deep learning based time series analysis with frequency transformation. *arXiv* **2023**, arXiv:2302.02173.
37. Puech, T.; Boussard, M.; D'Amato, A.; Millerand, G. A fully automated periodicity detection in time series. In *Advanced Analytics and Learning on Temporal Data: 4th ECML PKDD Workshop, AALTD 2019, Würzburg, Germany, 20 September 2019*; Revised Selected Papers 4; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 43–54.
38. Tang, P.; Zhang, W. PDMLP: Patch-based Decomposed MLP for Long-Term Time Series Forecastin. *arXiv* **2024**, arXiv:2405.13575.
39. Lai, G.; Chang, W.C.; Yang, Y.; Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 95–104.
40. Traffic Dataset. Available online: http://pems.dot.ca.gov/ (accessed on 14 May 2024).
41. Weather Dataset. Available online: https://www.bgc-jena.mpg.de/wetter/ (accessed on 14 May 2024).
42. ILI Dataset. Available online: https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html (accessed on 14 May 2024).
43. Citypower Dataset. Available online: https://kaggle.com/datasets/fedesoriano/electric-power-consumption (accessed on 14 May 2024).