TS3Net: Triple Decomposition with Spectrum Gradient for Long-Term Time Series Analysis

Xiangkai Ma, Xiaobin Hong, Sanglu Lu, Wenzhong Li
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China
xiangkai.ma@smail.nju.edu.cn, xiaobinhong@smail.nju.edu.cn, sanglu@nju.edu.cn, lwz@nju.edu.cn,

***Abstract*—Time series analysis has a wide range of applications in the fields of weather forecasting, traffic management, fault detection, intelligent operation, etc. In the real world, time series typically consist of dynamic fluctuations and mixtures of periodicities, which bring challenges on modeling and analyzing their patterns. To overcome the complexities, a common approach is to decompose long-term time series into sub-components for easier analysis. Unlike conventional time series decomposition that decouples a series into the trend and seasonal parts, we proposed a novel triple decomposition method to decouple a long-term series into three components: trend-part, regular-part, and fluctuant-part. Notably, the third part is a particular component that represents the dynamic spectral fluctuation in time series with the formulation of spectrum gradient. Based on triple decomposition, we propose a novel task-general deep learning model called TS3Net for long-term series analysis. It introduces a temporal-frequency block (TF-Block) with a multi-branch structure to expand the time series into a *2D* temporal-frequency distribution. Subsequently, deep representation can be learned by a vision architecture that captures the dynamic variations from the complex multi-periodic series. The decomposed components are processed by TS3Net individually, and their results are integrated to form the final result for time series analysis. We conduct extensive experiment based on six open datasets to evaluate the proposed method in comparison with 10 baselines. Numerical results show that TS3Net significantly outperforms the state-of-the-art methods on both time series forecasting and imputation tasks. The source codes of TS3Net are publicly available on https://anonymous.4open.science/r/TS3Net-F20C.**

***Index Terms*—Time Series Forecast, Triple Decomposition, Dynamic Temporal-Spectral Variation, Spectrum Gradient, Multi-Periodic Series.**

I. Introduction

Long-term time series refers to a series of well-defined data items collected over months or years which has clear seasonal periodicity and time varying patterns. In recent years, long-term time series analysis has become a popular research topic [1]–[10] encompassing many downstream tasks such as forecasting [11]–[17], classification [18]–[21], and anomaly detection [22], [23], and showing great potential in many application fields including sensor-based industrial fault diagnosis [24], [25], power consumption detection for production equipment [26], weather forecasting [27], and imputation of missing data [28].

In the real world, long-term time series typically consists of dynamic trends and mixtures of periodicities. For example, electricity consumption in office buildings may have a periodicity pattern of weekday energy consumption being greater than weekend energy consumption. In addition, electricity consumption can show long-term trends and unpredictable fluctuation patterns with urban development. The power demand in a city may have a trend with the economic condition and a regular periodicity with the air temperature, but also shows dynamics variations due to social events. Based on the observations, long-term series reveals typical characteristics on *trend*, *regular periodicities*, and *dynamic frequency fluctuations*. The intermixing and overlapping of these factors make the modeling and analyzing of long-term series extremely challenging.

Recently long-term series analysis has become a popular topic in the deep learning community. Many works were proposed to use neural networks to model temporal dependency and frequency characteristics of time series, which include the CNN-based methods (e.g., TCN [29], SCINet [30], TimesNet [2] and MICN [1]), the RNN-based methods (e.g., LSSL [31] and LSTM [32]), and the Transformer-based methods (e.g., Autoformer [4], Non-stationary Transformer [5] and PatchTST [33]). While the existing methods typically decompose the time series into the *trend-part* and the *seasonal-part*, they emphasize on the regular periodicities in long term and neglect the subtle fluctuation of individual frequency components at different time points. *Little attention was paid to capture the dynamic fluctuant frequency features at different scale of the long-term series.*

To address this challenge, we propose a novel triple decomposition approach for long-term time series analysis. The basic idea is illustrated in Fig. 1. The upper left of the figure shows a time series with combination of long and short periodic dynamic variations. The time series is firstly decoupled into a *trend-part* and a *seasonal-part*. Subsequently, the seasonal-part is transformed into a *2D* temporal-frequency (TF) distribution with spectrum expansion, which shows its dynamic energy variations on multiple frequency sub-bands. Thirdly, *spectrum gradients* are computed from the TF distribution to capture the fluctuation patterns in the spectrum domain. Finally, based on the spectrum gradient, the seasonal-part is further decoupled into the *regular-part* and the *fluctuant-part*, which represent the stable spectrum component and the dynamic spectrum fluctuation in the time series respectively. More specifically, based on the concept of TF distribution space and spectral gradient, we can extract from the seasonal-part the regular-part which consists of dynamic variations happening periodically.

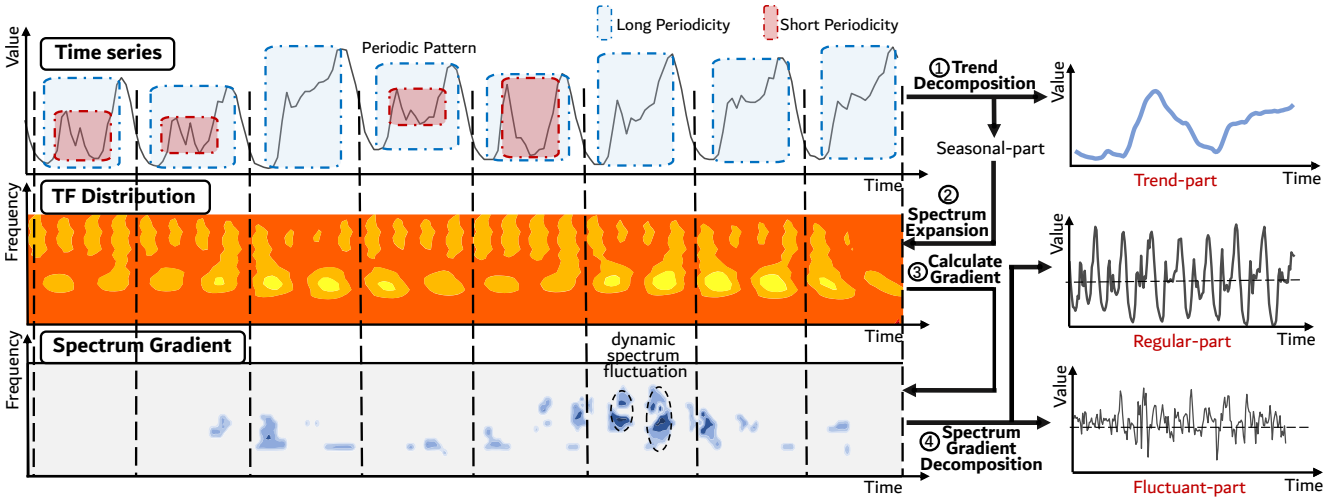


Fig. 1. Illistration of decomposing a long-term time series with mixture of periodicities into three components: trend-part, regular-part, and fluctuant-part. ① Using trend decomposition to decompose the time series into trend-part and seasonal-part. ② Using spectrum expantion to transform the seasonal-part into a *2D* temporal-frequency (TF) distribution. ③ Calculating the spectrum gradient from the TF distribution. ④ Using spectrum gradient to obtain the regular-part and fluctuant-part.

Where the regular-part contains a stable periodic pattern which is more predictable. Besides, the fluctuant-part is modeled independently similar to the trend-part, and serves as supplementary information to describe dynamic periodic patterns.

Based on triple decomposition, we go beyond the previous time series deep learning backbones and propose a novel task-general deep learning model called TS3Net for long-term time series analysis. A temporal-frequency block (TF Block) is designed to expand the time series into a *2D* tensor composed of continuous frequency sub-bands. By transforming time series into the temporal-frequency space, the stacked TF-Block module can break the bottleneck of representation capability in the original *1D* space and successfully capture the dynamic spectrum variations from the complex multi-periodic series. The regular-part, fluctuant-part and trend-part of the time series are modeled separately and processed by the TS3Net. The deep representations of the decoupled time series are captured by a vision architecture, followed by a prediction head trained individually. The prediction results of the three components are integrated together to form the final result for time series forecasting.

We conduct extensive experiment based on six open datasets to evaluate the proposed method. Numerical results show that TS3Net significantly outperforms the state-of-the-art methods on both time series forecasting and imputation tasks. The contributions of our work are as follows.

- Unlike conventional time series decomposition that decouples a series into trend and seasonal parts, we propose a novel triple decomposition method to decouple a long-term series into three components: trend-part, regular-part, and fluctuant-part. Where the fluctuant-part is a particular component to explicitly represent the dynamic spectral fluctuation in time series based on the proposed formulation of spectrum gradient in Section III-A.
- We propose a triple decomposition based deep learning model called TS3Net for long-term series analysis. It introduces a temporal-frequency block (TF-Block) with a multi-branch structure to convert the time series into a *2D* temporal-frequency distribution. Subsequently, deep representation are learned by a vision architecture to capture the dynamic spectrum variations from the complex series. The decomposed components are processed by TS3Net individually, and their results are integrated to form the final result for time series analysis.
- We conducted extensive experiments on 6 public datasets to evaluate the performance of the proposed method in comparison with 10 baselines. It shows that TS3Net significantly outperforms the SOTA methods on both time series forecasting and imputation tasks, and the performance improves by up to 31% compared to the SOTA method.

The rest of this paper is structured as follows. In Section II, we provide a summary of related works relevant to time series anlaysis. Section III presents the proposed model architecture, which includes the specific design of the Triple Decomposition and the TF-Block. In Section IV, we demonstrate the results of the numerical experiments in long-term time series forecasting and imputation benchmarks and conduct a comprehensive analysis to determine the effectiveness of the Triple-Decomposition scheme for time series forecasting. Finally, Section V concludes the works of this paper.

II. Related Work

We introduce the related works in terms of deep learning models, frequency analysis, and spatial-temporal forecasting for time series.

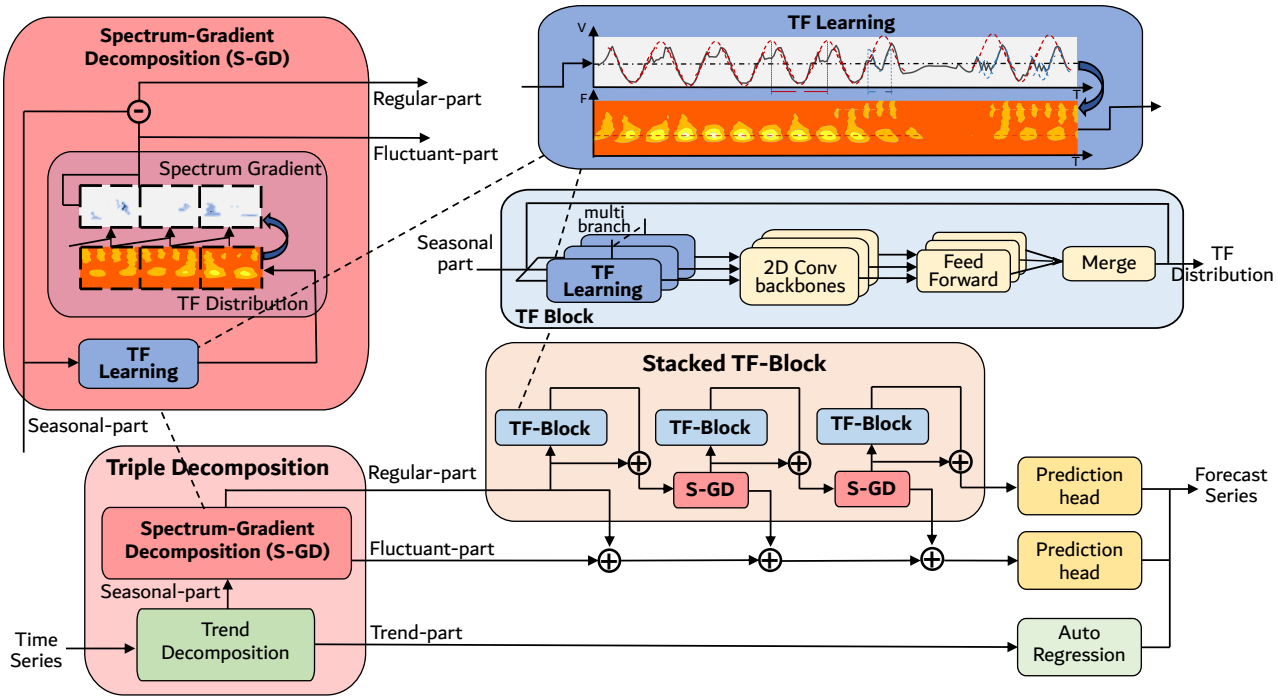


Fig. 2. The overall architecture of TS3Net. It consists of multiple TF-Blocks stacked in a residual way, with a Spectrum-Gradient Decomposition (S-GD) layer inserted between adjacent TF-Blocks, which are used to capture dynamic spectrum from different wavelet-generating functions.

A. Deep Learning Models for Time Series Analysis

There have been many studies applying deep learning models for time series analysis, such as RNN-based models [31], [34], MLP-based models [35]–[37], and TCN-based models [29]. More specifically, CNNs have achieved excellent performances in many time series areas [38], [39], thanks to the ability of their deep stacked convolution to extract global information from large-size input data, which is particularly suitable for long-term forecasting. Since both local features and global information play an important role in series modeling, MICN [1] considered both local and global contexts to achieve time series forecasting with linear computational complexity and memory cost.

Transformer [40] has shown great power in sequential prediction tasks. However, the self-attention mechanism is limited to quadratic complexity and struggles with long-term forecasting tasks. Therefore many researchers have focused on achieving linear complexity by improving the self-attentive. FEDformer [3] used Fourier enhanced blocks as an alternative to the self-attentive mechanism, which reduced the computational overhead to linear complexity without loss of accuracy. Autoformer [4] introduced an auto-correlation mechanism that discovered and aggregated the similarity of sub-seriess based on periodicity information. Dlinear [36] employed a channel-independent design and achieved satisfying forecasting performance using only the linear layer as the backbone. Besides, the PatchTST [33] model utilises a set of sub-series of univariate time series as tokens together with a channel-independent design to achieve the state-of-the-art among the current transformer-based models.

However, the CNNs and Transformers are modeled based on CV and NLP backgrounds respectively, which lack of consideration of the mixture of multi-periodicity and dynamic spectrum of long-term time series.

B. Frequency Analysis for Time Series

Wavelet Transform (WT) has been widely used in signal processing and analysis in many fields over the past two decades [38], [41]–[44]. Recently, frequency analysis based on Fast Fourier Transform (FFT) and Wavelet Transform (WT) have shown great potential in time series analysis. To model multi-periodicity time series, TimesNet [2] raveled out the complex temporal variations into the multiple intra-period and inter-period variations. In contrast to previous methods that keep low frequency components and throw away the high frequency ones, FEDformer [3] generated a set of mixed frequency components by Fourier analysis, and introduced the Fourier enhanced blocks as an alternative to the self-attentive mechanism.

However, the conventional spectral analysis only estimate the main frequency component of the series without taking into account the periodic variation of temporal-frequency distribution and spectral dynamics. Different from their works, this paper proposes a triple decomposition method by introducing spectrum gradients to explicitly capture the dynamic variations of the spectral patterns.

C. Spatial-Temporal Series Forecasting

Spatial-temporal series analysis occupies an important position in research related to time series and thus received more and more attention in recent years [11], [12], [14]. RNTrajRec [43] is a trajectory recovery framework based on a road network augmentation Transformer, which included a spatial-temporal transformer component to capture the spatial features of GPS points in the trajectory. The Self-supervised Spatial-Temporal Bottleneck Attentive Network (SSTBAN) [44] reduced the computational complexity while encoding global spatial-temporal dynamics to produce robust latent representations for long-term traffic forecasting.

While the previous works lack of consideration of involving patterns in spatial-temporal series, our proposed triple decomposition based method can effectively capture dynamic temporal patterns at multiple periodicities from the formulation of temporal-frequency distribution and spectral gradient.

III. Methodology

In this section, we introduce the long-term series analysis method with triple decomposition. Besides, the specific description of the important symbols involved in the method is shown in Table I.

A. TS3Net: Long-Term Series Analysis with Triple Decomposition

We propose a triple decomposition based time series analysis framework called TS3Net to capture the complex periodic patterns and spectrum dynamics from the temporal-frequency (TF) distribution of long-term series. The details are illustrated in Figure 2.

Firstly, the original time series is decoupled into the trend-part and seasonal-part by *trend decomposition*, and then the seasonal-part is decoupled into the regular-part and fluctuant-part by *spectrum gradient decomposition*. Where the regular-part involves the long-term periodicity of the time series and the fluctuant-part represents the spectrum dynamics calculated from the spectrum gradient.

Then we design a *stacked TF-Block* structure for learning the temporal-frequency distribution from long-term series. In each TF-Block, we apply spectrum expansion to expand the time series into a *2D* tensor composed of continuous frequency sub-bands. Specifically, the multiple frequency sub-bands and temporal variations are embedded into the columns and rows of the temporal-frequency distribution space respectively, which makes the spectrum dynamic variations to be easily modeled by the *2D* kernels. Subsequently, a multi-branch structure is designed to capture the spectrum characteristics with different Wavelet generation functions.

In the time series forecasting task, the decomposed components are processed by the stacked TF-blocks and the Autoregression layer separatively, and their results are combined to form the final prediction.

B. Triple Decomposition (TD)

The triple decomposition decouples a time series into the trend-part, regular-part, and fluctuant-part, which is introduced in the following.

1) Trend Decomposition: In fact, the trend-part can be observed in the low-frequency portion of 2D distribution, which can be explained by that trend-part is essentially a baseline drift phenomenon in the signal processing.

We adopt a recently popular decoupling approach [1], [3], [4] to decompose the original time series into the trend-part and seasonal-part.

Specifically, we use multiple averaging pooling layers with a set of scales to obtain the trend, and use padding operations to keep the length of the trend component constant. Therefore, the time series X_{1D} can be decoupled into a trend-part X_{trend} and a seasonal-part $X_{seasonal}$ with the following equation:

$$\begin{aligned} X_{trend} &= \mathrm{Avg}Pool\Big(Padding\big(X_{1D}\big)\Big), \\ X_{seasonal} &= X_{1D} - X_{trend}. \end{aligned} \tag{1}$$

2) Multi-Periodicity Patterns: For a time series with length T, we analyze the time series in the frequency domain by Fast Fourier Transform (FFT). The top k frequencies with large amplitude values are represented as follows:

$$\{f_1, \cdots, f_k\} = \underset{f_* \in \{1, \cdots, [\frac{T}{2}]\}}{\arg \mathrm{Topk}} \left(\mathrm{Amp}(\mathrm{FFT}(\mathbf{X}_{1:\mathrm{T}}))\right), \tag{2}$$

where $\mathrm{FFT}(\cdot)$ and $\mathrm{Amp}(\cdot)$ denote the FFT and the calculation of amplitude values. Finally, we compute the k latent period lengths p_i by $p_i = \left\lceil \frac{T}{f_i} \right\rceil$, where $i \in \{1, \cdots, k\}$ and k is hyper-parameter indicating the number of potential periodic patterns. For simplicity, we next describe each model component in detail from the perspective of a single branch i=1, and denote p_1 as T_f to represent the length of the periodic sub-series in this branch. Note that in practice we use the top-k periodicities with large amplitude values for time series analysis.

3) Spectrum-Gradient Decomposition (S-GD): As shown in Figure 1, temporal variation involves multiple sub-bands of the spectrum simultaneously in the TF distribution. We proposed a TF learning layer to explicitly capture variations tendency of sub-bands under continuous frequency to form a comprehensive representation of the spectrum.

In this work, the Complex Gaussian Wavelet was chosen as the wavelet generating function, which is frequently used in the statistical signal processing applications to study amplitudes and phases. The complex Gaussian function is defined as:

$$\psi(t) = C_p e^{-it} e^{-t^2}, \tag{3}$$

where C_p is a normalization factor that guarantees that the wavelet has unit energy; e^{-t^2} is a Gaussian envelope; and e^{-it} represents a complex sinusoid.

To analyze the frequency components of the time series at each time point, the time center of the time series is determined by the translation factor β. Besides, α is the scale factor that involves dilation and compression of the wavelet generating

TABLE I
THE SPECIFIC DESCRIPTION OF THE SYMBOLS INVOLVED IN THE PAPER.

Symbol formula	Definition
T	Lookback Length
C	Dimension of Multivariate Times Series
λ	The number of spectral sub-bands in the temporal-frequency distribution
$X_{1D} = \{x_1, x_2, ..., x_T\} \in \mathbb{R}^{T \times C}$	Input time series at dimension d
$S = \{s_1, s_2, ..., s_\lambda\}, s_i = (2 \cdot \lambda)/i, i \in [1, 2, ..., \lambda]$	Scale factors of Continuous Wavelet Transform
F_c	Central frequency of wavelet generating function ψ
$F_i = F_c/s_i$	Frequency of wavelet function ψ_i generated by scale factor s_i
$WT(\cdot)$	Wavelet Transformation
$IWT(\cdot)$	Inverse Wavelet Transform
$\psi(t)$	wavelet generating function with frequency F_c
$\psi_i(t)$	wavelet function with frequency F_i
$TF_i = Amp(WT(X_{1D}, \psi_i)) \in \mathbb{R}^{T \times C}$	Temporal-frequency coefficient of X_{1D} at frequency F_i
$X_{2D} = \{TF_1, ..., TF_\lambda\}^\top \in \mathbb{R}^{\lambda \times T \times C}$	TF tensor
$\Delta_{2D} \in \mathbb{R}^{\lambda \times T \times C}$	Spectrum gradients from the TF distribution
$\Delta_{1D} = IWT(\Delta_{2D}) \in \mathbb{R}^{T \times C}$	Spectrum vector from the TF distribution

function. Assume $\psi_{\alpha,\beta}(t)$ is the dilation and translation of the wavelet generating function $\psi(t)$, which is written as follows:

$$\psi_{\alpha,\beta}(t) = \frac{1}{\sqrt{|\alpha|}} \psi\left(\frac{t-\beta}{\alpha}\right). \tag{4}$$

Based on the localization in both time (β) and frequency (α) in Equ. 4, the wavelet transform calculates the local amplitude $TF_{\alpha,\beta}$ in temporal-frequency space, which reflects the energy contained in the frequency component f_α of the time series at timestamp β. The computation is as follows:

$$TF_{\alpha,\beta} = \int_{-\infty}^{\infty} x(t) \cdot \psi^*_{\alpha,\beta}(t) dt = \left\langle x, \psi^*_{\alpha,\beta} \right\rangle, \tag{5}$$

where $*$ symbolizes the complex conjugate, and $\langle \cdot, \cdot \rangle$ represents the inner product operation.

Assume the original *1D* time series X_{1D} has length T and C recorded variate. To transform the time series we choose a set of continuous scale factors:

$$S = \{s_1, ..., s_\lambda\}, s_i = \frac{2 \cdot \lambda}{i}, F_i = \frac{F_c}{s_i}, i \in [1, ..., \lambda], \tag{6}$$

where F_c is the central frequency of the wavelet generating function, and λ is a hyper-parameter representing the number of sub-bands.

The wavelet $\psi_i = \{f_1, .., f_{L_i}\}$ with frequency F_i can be uniformly sampled from ψ with frequency F_c, where L and L_i denote the length of ψ and ψ_i respectively, and L_i can be computed by L and s_i as $L_i = L/s_i$. Thus, we calculate the temporal-frequency coefficient TF_i of $X_{1D} = \{x_1, .., x_T\}$ at frequency F_i as follows:

$$\begin{aligned} TF_{i,j} &= -\sqrt{s_i} \cdot \sum_{k=0}^{j} \left(x_k \cdot f_{\frac{L_i}{2}+j-k} \right), \\ TF_i &= Amp\big(WT(X_{1D}, \psi_i)\big) = \big\{TF_{i,1}, ..., TF_{i,T}\big\}, \end{aligned} \tag{7}$$

where $WT(\cdot)$ is the wavelet transformation; $Amp(\cdot)$ denotes the amplitude values; and TF_i represents the vector of amplitude of frequency F_i; T is the length of the input series X_{1D}.

We repeat the process for i from 1 to λ, and apply successive λ frequencies corresponding to ψ_i, we concatenate them into a *2D* tensor X_{2D} as:

$$X_{2D} = \{TF_1, .., TF_\lambda\}^\top \in \mathbb{R}^{\lambda \times T \times C}. \tag{8}$$

The spectrum gradient is formulated as the dynamic changes in the spectral domain, which intuitively reflects the irregular fluctuation of time series. To learn the dynamic spectrum patterns, we calculate spectrum gradients from the TF distribution and further decompose the seasonal component into regular-part and fluctuant-part.

The TF learning layer takes $X_{seasonal}$ as input and output X_{2D}. Subsequently, $X_{2D} \in \mathbb{R}^{\lambda \times T \times C}$ is split into u components along the time axis without overlap, where the i-th component $S^i_{2D} \in \mathbb{R}^{\lambda \times T_f \times C}$ indicates the spectrum of the length-T_f sub-series, where T_f and $u = T/T_f$ denote the length and number of sub-series, respectively. The spectrum gradient Δ^i_{2D} is defined as the difference between the spectrum S^{i-1}_{2D} and S^i_{2D} in the ith sub-series, which is derived by:

$$\begin{aligned} &\Delta^i_{2D} = S^i_{2D} - S^{i-1}_{2D}, S^0_{2D} = 0, \\ &\Delta_{2D} = \{\Delta^1_{2D}, .., \Delta^u_{2D}\} \in \mathbb{R}^{\lambda \times T \times C}, \\ &\Delta_{1D} = IWT\big(\Delta_{2D}\big) \in \mathbb{R}^{T \times C}, \end{aligned} \tag{9}$$

where $IWT(\cdot)$ denotes the Inverse Wavelet Transform.

Based on the above equation, the seasonal component can be decoupled into a regular-part $X_{regular}$ and a fluctuant-part $X_{fluctuant}$ as:

$$\begin{aligned} X_{regular} &= X_{seasonal} - \Delta_{1D}, \\ X_{fluctuant} &= \Delta_{2D}. \end{aligned} \tag{10}$$

In summary, the overall S-GD operation can be formulated as:

$$\textit{S-GD}(X_{\textit{seasonal}}) = \left[X_{\textit{regular}}, X_{\textit{fluctuant}}\right]. \tag{11}$$

C. Temporal Frequency Block (TF-Block)

As shown in Figure 2, we organize the TF-Block in a residual way to learn from the *2D* temporal-frequency distribution. The output of the embedding layer X_{1D}^0 is fed into the stacked TF-blocks for feature extraction. For the l-th TF-Block (in practice, we set the default value of the stacking number for TF-Block to 3), the process is formalized as:

$$\begin{aligned} \left[X_r^{l-1}, X_f^{l-1}\right] &= \textit{S-GD}\left(X_{1D}^{l-1}\right), \\ X_{1D}^{l} &= \textit{TF-Block}\left(X_r^{l-1}\right) + X_r^{l-1}. \end{aligned} \tag{12}$$

The operations in each TF-block involve three successive parts. Firstly, it calculates the spectrum component in $X_r^{l\text{-}1}$ by $\textit{WT}(\cdot)$ and $\textit{Amp}(\cdot)$, which has the converted *2D* TF-Distribution from the *1D* time series. Secondly, it captures the spectrum information from the *2D* tensors using a convolutional network. Thirdly, it calculates the weighted summation of the *1D* representation generated by a vision architecture from different wavelet basis functions. The operations are referred to as TF Learning Layer, FeedForward Layer, and Weight-learned Merge Layer respectively. And the *TF-Block* is formalized as follows:

$$\begin{aligned} X_r^{l-1} &= \textit{TripleDecomposition}(X_{1D}^{l-1}), \\ X_{2D}^{l,i} &= \left\{ \textit{Amp}\left(WT\left(X_r^{l-1}, \psi_{i,j}\right)\right) \right\}^{\top}, \\ & \qquad i \in \{1, ..., m\}, j \in \{1, ..., \lambda\}, \\ \overline{X}_{2D}^{l,i} &= \textit{ConvBackbone}\left(X_{2D}^{l,i}\right), i \in \{1, ..., m\}, \\ \overline{X}_{1D}^{l,i} &= \textit{FeedForward}\left(\overline{X}_{2D}^{l,i}\right), i \in \{1, ..., m\}, \\ X_{1D}^{l} &= X_r^{l-1} + \textit{Merge}\left(\overline{X}_{1D}^{l,1}, ..., \overline{X}_{1D}^{l,m}\right), \end{aligned} \tag{13}$$

where $X_{2D}^{l,i} \in \mathbb{R}^{\lambda \times T \times d_{\textit{model}}}$ is the *2D* tensor transformed by the i-th mother wavelet function ψ which generates $\{\psi_{i,1}, ..., \psi_{i,\lambda}\}$. Besides, we designed m branches to capture different underlying patterns. After the transformation, we process the *2D* tensor by the inception block [45] as $\textit{ConvBackbone}(\cdot)$, which is one of the most well-acknowledged vision backbones involving a multi-scale *2D* kernel. Then we transform the learned *2D* representation $\overline{X}_{2D}^{l.i}$ back to *1D* space $\overline{X}_{1D}^{l,i} \in \mathbb{R}^{T \times d_{\textit{model}}}$ by FeedForward Layer. Finally, we concatenate all *1D* representations $\left\{\overline{X}_{1D}^{l,1}, ..., \overline{X}_{1D}^{l,m}\right\}$ and the seasonal-part $X_r^{l\text{-}1}$ to form the output of the TF-Block.

D. Overall Forecasting Process

After triple decomposition, we process the three subcomponents individually, and combine their output to form the final prediction result.

Algorithm 1: Master Training Stage of TS3Net

Input: Lookback series $X_{1D} = \{x_1, x_2, ..., x_T\} \in \mathbb{R}^{T \times C}$.
Output: Forecasting series $Y \in \mathbb{R}^{T_{\textit{pred}} \times C}$.

```
Randomly initialize the parameter set θ_1;
for iteration = 1,2,3, ... do
    Decompose originial time series into three parts: X_regular,
      X_fluctuant and X_trend by (1)-(10);
    Learning deep representation from X_regular by Stacked
      TF-Block, and that subsequent X_regular will pass
      through the S-GD component before being fed into the
      TF-Block, as follows (12). And the implementation of
      TF-Block are shown in (13);
    Getting Y_regular from the output (X_r^N) of Stacked
      TF-Block, by the Prediction-Head (based on MLP), as
      follows (14);
    Collect the output (X_f^{l-1}) of the S-GD component in TD
      and Stacked TF-Block together as input to the
      Prediction-Head, and outputs Y_fluctuant, as follows (15);
    Get the prediction of trend-part X_trend as Y_trend by (16);
    Get the final prediction result by combining the output
      of all three components, as follows (17);
    Get the forecasting loss L_mse between prediction and
      ground truth of future time series and update
      parameters θ_1 according to the gradient of L_mse;
return θ_1
```

a) Regular-part: The regular-part is fed into the stacked TF-blocks for feature extraction. The TS3Net adopts a multilayer structure with recursive decomposition to enable high-level feature representation. In each layer, the input tensor first undergoes the S-GD, and the refined regular-part $X_r^{l\text{-}1}$ obtained from the S-GD (as computed by Eq. 12) is used as the input of the TF-Block. The extracted high-level feature X_r^N (the output of the *N*-th TF-Block) is fed into a prediction head (a multilayer perceptron), which outputs the prediction result:

$$Y_{\textit{regular}} = \textit{Prediction}\left(X_r^N\right). \tag{14}$$

b) Fluctuant-part: Similar to the process of regular-part, the fluctuant-part is also fed into the stacked TF-blocks for feature learning. In each layer, the S-GD cursively builds the $X_f^{l\text{-}1}$ computed by Eq. 12. We combine the $X_f^{l\text{-}1}$ from each layer with $X_{\textit{fluctuant}}$ as high-level feature presentation, which is fed into a multi-layer perceptron (MLP) to form the prediction results. Specifically, the process is formalized as:

$$\begin{aligned} X_f^{\textit{sum}} &= IWT\left(X_{\textit{fluctuant}} + \sum_l X_f^{l-1}\right), \\ Y_{\textit{fluctuant}} &= \textit{Prediction}\left(X_f^{\textit{sum}}\right). \end{aligned} \tag{15}$$

c) Trend-part: Since the trend-part is a low-frequency component without obvious periodicity, we design an Autoregression layer based on multi-layer perceptron (MLP) to forecast the future value. For the trend-part $X_{trend} \in \mathbb{R}^{T \times C}$ obtained from the decomposition layer, the output is:

$$Y_{\textit{trend}} = \textit{Autoregression}\left(X_{\textit{trend}}\right) \in \mathbb{R}^{T_{\textit{pred}} \times C}. \tag{16}$$

TABLE II
DESCRIPTION OF DATASETS. THE DATASET SIZE IS ORGANIZED IN (TRAIN, VALIDATION, TEST).

Tasks	Dataset	Dim	Series Length	Dataset Size	Information (Frequency)
Forecasting (Long-term)	ETTm1, ETTm2	7	{96, 192, 336, 720}	(34465, 11521, 11521)	Electricity (15 mins)
	ETTh1, ETTh2	7	{96, 192, 336, 720}	(8545, 2881, 2881)	Electricity (15 mins)
	Electricity	321	{96, 192, 336, 720}	(18317, 2633, 5261)	Electricity (Hourly)
	Traffic	862	{96, 192, 336, 720}	(12185, 1757, 3509)	Transportation (Hourly)
	Weather	21	{96, 192, 336, 720}	(36792, 5271, 10540)	Weather (10 mins)
	Exchange	8	{96, 192, 336, 720}	(5120, 665, 1422)	Exchange rate (Daily)
	ILI	7	{24, 36, 48, 60}	(617, 74, 170)	Illness (Weekly)
Imputation	ETTm1, ETTm2	7	96	(34465, 11521, 11521)	Electricity (15 mins)
	ETTh1, ETTh2	7	96	(8545, 2881, 2881)	Electricity (15 mins)
	Electricity	321	96	(18317, 2633, 5261)	Electricity (15 mins)
	Weather	21	96	(36792, 5271, 10540)	Weather (10 mins)

TABLE III
EXPERIMENT CONFIGURATION OF TS3NET. ALL EXPERIMENTS USE THE ADAM [46] OPTIMIZER WITH THE DEFAULT HYPERPARAMETER CONFIGURATION FOR (β_1, β_2) AS $(0.9, 0.999)$.

Tasks / Configurations	Model Hyper-parameter				Training Process			
	λ	Layers	d_{min} †	d_{max} †	LR*	Loss	Batch Size	Epochs
Long-term Forecasting	100	2	32	512	10^{-4}	MSE	32	10
Imputation	100	2	64	128	10^{-3}	MSE	16	10

† $d_{\text{model}} = \min\{\max\{2^{\lceil \log C \rceil}, d_{\text{min}}\}, d_{\text{max}}\}$, where C is input series dimension.
∗ LR means the initial learning rate.

Combining the output of all three components, we have the final prediction result as follows.

$$Y = Y_{trend} + Y_{regular} + Y_{fluctuant}. \tag{17}$$

The detailed process of the master training stage is in Algorithm 1.

IV. EXPERIMENT

A. Benchmarks

To evaluate the performace of the proposed method, we extensively experiment on mainstream time series analysis tasks including long-term forecasting and imputation (i.e., predicting the missing data in a time series). The long-term forecasting task is evaluated with six popular real-world datasets, including: (1) **ETT (ETTh1, ETTh2, ETTm1 and ETTm2)**[1] [47] contains six power load features and readings of oil temperature used for monitoring electricity transformers between July 2016 and July 2018. ETT involves four subsets. ETTm1 and ETTm2 are recorded at 15-minute intervals, while ETTh1 and ETTh2 are recorded hourly. (2) **Exchange**[2] [48] records daily exchange rates of eight different countries ranging from 1990 to 2016. (3) **Weather**[3] contains 21 meteorological indicators, such as temperature, humidity, and precipitation, which were recorded every 10 minutes in the year of 2020. (4) **Electricity**[4] comprises hourly power consumption of 321 clients from 2012 to 2014. (5) **Traffic**[5] reports the number of vehicles loaded on 862 roads at each moment in time. (6) **ILI**[6] includes weekly recorded data on the number of patients with seven influenza-like illnesses between 2002 and 2021. The imputation task is evaluated with datasets from the electricity and weather scenarios, including ETT and Weather. To compare the performance under different proportions of missing data, we randomly mask the time points with a ratio of $\{12.5\%, 25\%, 37.5\%, 50\%\}$. Table II summarizes the dataset descriptions.

B. Baselines

We compare the proposed TS3NET model with the well-acknowledged and advanced models for long-term time series forecasting and imputation, which include the CNN-based Model: **TimesNet** [2] and **MICN** [1]; the MLP-based models: **LightTS** [37] and **DLinear** [36]; the Transformer-based models: **Informer** [47], **Pyraformer** [49], **Autoformer** [4], **FEDformer** [3], **Stationary Transformer** [5] and **PatchTST** [33]. Overall, more than ten baselines are included for a comprehensive comparison.

To ensure a fair comparison, we follow the experimental settings of TimesNet [2]. Specifically, in all forecasting and imputation tasks, we set the input length to 96. Note that the PatchTST [33] paper reported their results with input size 336

[1]https://github.com/zhouhaoyi/Informer2020
[2]https://github.com/laiguokun/multivariate-time-series-data
[3]https://www.bgc-jena.mpg.de/wetter/
[4]https://archive.ics.uci.edu/dataset/321/electricity
[5]http://pems.dot.ca.gov
[6]https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html

TABLE IV

COMPARISON OF PERFORMANCE ON LONG-TERM FORECASTING TASK. THE PAST SERIES LENGTH IS SET TO 36 FOR ILI AND 96 FOR THE OTHERS. ALL THE RESULTS CONSISTED OF 4 DIFFERENT PREDICTION LENGTHS, THAT IS $\{24, 36, 48, 60\}$ FOR ILI AND $\{96, 192, 336, 720\}$ FOR THE OTHERS.

Models		**TS3Net (Ours)**		PatchTST [33]		TimesNet [2]		MICN [1]		LightTS [37]		DLinear [36]		FEDformer [3]		Stationary [5]		Autoformer [4]		Pyraformer [49]		Informer [47]	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.324	**0.362**	0.322	0.364	0.338	0.375	**0.316**	0.362	0.374	0.400	0.345	0.372	0.379	0.419	0.386	0.398	0.505	0.475	0.543	0.510	0.672	0.571
	192	**0.358**	**0.380**	0.368	0.389	0.374	0.387	0.363	0.390	0.400	0.407	0.380	0.389	0.426	0.441	0.459	0.444	0.553	0.496	0.557	0.537	0.795	0.669
	336	**0.404**	**0.407**	0.412	0.408	0.410	0.411	0.408	0.426	0.438	0.438	0.413	0.413	0.445	0.459	0.495	0.464	0.621	0.537	0.754	0.655	1.212	0.871
	720	**0.458**	**0.436**	0.469	0.445	0.478	0.450	0.459	0.464	0.527	0.502	0.474	0.453	0.543	0.490	0.585	0.516	0.671	0.561	0.908	0.724	1.166	0.823
	Avg	**0.386**	**0.396**	0.393	0.402	0.400	0.406	0.387	0.411	0.435	0.437	0.403	0.407	0.448	0.452	0.481	0.456	0.588	0.517	0.691	0.607	0.961	0.734
ETTm2	96	**0.169**	**0.253**	0.178	0.260	0.187	0.267	0.179	0.275	0.209	0.308	0.193	0.292	0.203	0.287	0.192	0.274	0.255	0.339	0.435	0.507	0.365	0.453
	192	**0.239**	**0.297**	0.251	0.303	0.249	0.309	0.262	0.326	0.311	0.382	0.284	0.362	0.269	0.328	0.280	0.339	0.281	0.340	0.730	0.673	0.533	0.563
	336	**0.298**	**0.335**	0.314	0.346	0.321	0.351	0.305	0.353	0.442	0.466	0.369	0.427	0.325	0.366	0.334	0.361	0.339	0.372	1.201	0.845	1.363	0.887
	720	0.399	**0.393**	0.403	0.401	0.408	0.403	**0.389**	0.407	0.675	0.587	0.554	0.522	0.421	0.415	0.417	0.413	0.433	0.432	3.625	1.451	3.379	1.338
	Avg	**0.276**	**0.319**	0.287	0.328	0.291	0.333	0.284	0.340	0.409	0.436	0.350	0.401	0.305	0.349	0.306	0.347	0.327	0.371	1.498	0.869	1.410	0.810
ETTh1	96	0.387	0.417	0.378	0.400	0.384	0.402	0.398	0.427	0.424	0.432	0.386	**0.400**	**0.376**	0.419	0.513	0.491	0.449	0.459	0.664	0.612	0.865	0.713
	192	0.436	0.436	0.440	0.429	0.436	**0.429**	0.430	0.453	0.475	0.462	0.437	0.432	**0.420**	0.448	0.534	0.504	0.500	0.482	0.790	0.681	1.008	0.792
	336	0.460	**0.449**	0.472	0.455	0.491	0.469	**0.440**	0.460	0.518	0.488	0.481	0.459	0.459	0.465	0.588	0.535	0.521	0.496	0.891	0.738	1.107	0.809
	720	**0.464**	**0.465**	0.475	0.478	0.521	0.500	0.491	0.509	0.547	0.533	0.519	0.516	0.506	0.507	0.643	0.616	0.514	0.512	0.963	0.782	1.181	0.865
	Avg	**0.436**	**0.441**	0.441	0.441	0.458	0.450	0.440	0.462	0.491	0.479	0.456	0.452	0.440	0.460	0.570	0.537	0.496	0.487	0.827	0.703	1.040	0.795
ETTh2	96	**0.290**	**0.339**	0.297	0.341	0.340	0.374	0.299	0.364	0.397	0.437	0.333	0.387	0.358	0.397	0.476	0.458	0.346	0.388	0.645	0.597	3.755	1.525
	192	**0.374**	**0.391**	0.385	0.394	0.402	0.414	0.422	0.441	0.520	0.504	0.477	0.476	0.429	0.439	0.512	0.493	0.456	0.452	0.788	0.683	5.602	1.931
	336	**0.419**	0.432	0.434	**0.428**	0.452	0.452	0.447	0.474	0.626	0.559	0.594	0.541	0.496	0.487	0.552	0.551	0.482	0.486	0.907	0.747	4.721	1.835
	720	**0.429**	**0.445**	0.432	0.451	0.462	0.468	0.442	0.467	0.863	0.672	0.831	0.657	0.463	0.474	0.562	0.560	0.515	0.511	0.963	0.783	3.647	1.625
	Avg	**0.378**	**0.401**	0.387	0.404	0.414	0.427	0.403	0.437	0.602	0.543	0.559	0.515	0.437	0.449	0.526	0.516	0.450	0.459	0.826	0.703	4.431	1.729
Electricity	96	**0.153**	**0.247**	0.177	0.267	0.168	0.272	0.164	0.269	0.207	0.307	0.197	0.282	0.193	0.308	0.169	0.273	0.201	0.317	0.386	0.449	0.274	0.368
	192	**0.168**	**0.263**	0.184	0.274	0.184	0.289	0.177	0.285	0.213	0.316	0.196	0.285	0.201	0.315	0.182	0.286	0.222	0.334	0.378	0.443	0.296	0.386
	336	**0.180**	**0.272**	0.200	0.290	0.198	0.300	0.193	0.304	0.230	0.333	0.209	0.301	0.214	0.329	0.200	0.304	0.231	0.338	0.376	0.443	0.300	0.394
	720	**0.201**	**0.291**	0.242	0.324	0.220	0.320	0.212	0.321	0.265	0.360	0.245	0.333	0.246	0.355	0.222	0.321	0.254	0.361	0.376	0.445	0.373	0.439
	Avg	**0.176**	**0.268**	0.201	0.289	0.192	0.295	0.187	0.295	0.229	0.329	0.212	0.300	0.214	0.327	0.193	0.296	0.227	0.338	0.379	0.445	0.311	0.397
Traffic	96	**0.424**	**0.267**	0.436	0.279	0.593	0.321	0.519	0.309	0.615	0.391	0.650	0.396	0.587	0.366	0.612	0.338	0.613	0.388	0.867	0.468	0.719	0.391
	192	**0.445**	**0.265**	0.447	0.283	0.617	0.336	0.537	0.315	0.601	0.382	0.598	0.370	0.604	0.373	0.613	0.340	0.616	0.382	0.869	0.467	0.696	0.379
	336	**0.447**	**0.270**	0.459	0.285	0.629	0.336	0.534	0.313	0.613	0.386	0.605	0.373	0.621	0.383	0.618	0.328	0.622	0.337	0.881	0.469	0.777	0.420
	720	0.479	0.295	**0.472**	**0.291**	0.640	0.350	0.577	0.325	0.658	0.407	0.645	0.394	0.626	0.382	0.653	0.355	0.660	0.408	0.896	0.473	0.864	0.472
	Avg	**0.449**	**0.274**	0.453	0.285	0.620	0.336	0.542	0.316	0.622	0.392	0.625	0.383	0.610	0.376	0.624	0.340	0.628	0.379	0.878	0.469	0.764	0.416
Weather	96	0.172	**0.210**	0.179	0.219	0.172	0.220	**0.161**	0.229	0.182	0.242	0.196	0.255	0.217	0.296	0.173	0.223	0.266	0.336	0.622	0.556	0.300	0.384
	192	0.231	**0.257**	0.225	0.259	**0.219**	0.261	0.220	0.281	0.227	0.287	0.237	0.296	0.276	0.336	0.245	0.285	0.307	0.367	0.739	0.624	0.598	0.544
	336	0.288	0.303	0.280	**0.298**	0.280	0.306	**0.278**	0.331	0.282	0.334	0.283	0.335	0.339	0.380	0.321	0.338	0.359	0.395	1.004	0.753	0.578	0.523
	720	0.358	**0.348**	0.355	0.357	0.365	0.359	**0.311**	0.356	0.352	0.386	0.345	0.381	0.403	0.428	0.414	0.410	0.419	0.428	1.420	0.934	1.059	0.741
	Avg	0.262	**0.280**	0.267	0.283	0.259	0.287	**0.243**	0.299	0.261	0.312	0.265	0.317	0.309	0.360	0.288	0.314	0.338	0.382	0.946	0.717	0.634	0.548
Exchange	96	**0.085**	**0.204**	0.091	0.212	0.107	0.234	0.102	0.235	0.116	0.262	0.088	0.218	0.148	0.278	0.111	0.237	0.197	0.323	1.748	1.105	0.847	0.752
	192	0.181	**0.301**	0.179	0.301	0.226	0.344	**0.172**	0.316	0.215	0.359	0.176	0.315	0.271	0.380	0.219	0.335	0.300	0.369	1.874	1.151	1.204	0.895
	336	0.334	0.418	0.328	0.416	0.367	0.448	**0.272**	**0.407**	0.377	0.466	0.313	0.427	0.460	0.500	0.421	0.476	0.509	0.524	1.943	1.172	1.672	1.036
	720	0.887	0.713	0.830	0.686	0.964	0.746	**0.714**	**0.658**	0.831	0.699	0.839	0.695	1.195	0.841	1.092	0.769	1.447	0.941	2.085	1.206	2.478	1.310
	Avg	0.371	0.409	0.357	0.404	0.416	0.443	**0.315**	**0.404**	0.385	0.447	0.354	0.414	0.519	0.500	0.461	0.454	0.613	0.539	1.913	1.159	1.550	0.998
ILI	24	**1.863**	**0.890**	2.558	0.906	2.317	0.934	2.684	1.112	8.313	2.144	2.398	1.040	3.228	1.260	2.294	0.945	3.483	1.287	7.394	2.012	5.764	1.677
	36	1.988	0.916	2.290	0.897	1.972	0.920	2.507	1.013	6.631	1.902	2.646	1.088	2.679	1.080	**1.825**	**0.848**	3.103	1.148	7.551	2.031	4.755	1.467
	48	**1.757**	**0.823**	2.195	0.905	2.238	0.940	2.423	1.012	7.299	1.982	2.614	1.086	2.622	1.078	2.010	0.900	2.669	1.085	7.662	2.057	4.763	1.469
	60	**1.885**	**0.872**	1.954	0.889	2.027	0.928	2.653	1.085	7.283	1.985	2.804	1.146	2.857	1.157	2.178	0.963	2.770	1.125	7.931	2.100	5.264	1.564
	Avg	**1.873**	**0.875**	2.249	0.899	2.139	0.931	2.567	1.056	7.382	2.003	2.616	1.090	2.847	1.144	2.077	0.914	3.006	1.161	7.635	2.050	5.137	1.544
1^{st} Count		**66**		4		2		13		0		1		2		2		0		0		0	

TABLE V
COMPARISON OF PERFORMANCE ON IMPUTATION TASK. IN THE EXPERIMENT, WE RANDOMLY MASK $\{12.5\%, 25\%, 37.5\%, 50\%\}$ TIME POINTS IN LENGTH-96 TIME SERIES AND REPORT ALL RESULTS FROM THE 4 DIFFERENT MASK RATIOS.

Models		**TS3Net (Ours)**		PatchTST [33]		TimesNet [2]		MICN [1]		LightTS [37]		DLinear [36]		FEDformer [3]		Stationary [5]		Autoformer [4]		Pyraformer [49]		Informer [47]	
Mask Ratio		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	12.5%	**0.016**	**0.084**	0.045	0.135	0.019	0.092	0.039	0.136	0.075	0.180	0.058	0.162	0.035	0.135	0.026	0.107	0.034	0.124	0.670	0.541	0.047	0.155
	25%	**0.020**	**0.092**	0.048	0.141	0.023	0.101	0.059	0.169	0.093	0.206	0.080	0.193	0.052	0.166	0.032	0.119	0.046	0.144	0.689	0.553	0.063	0.180
	37.5%	**0.024**	**0.099**	0.053	0.148	0.029	0.111	0.080	0.198	0.113	0.231	0.103	0.219	0.069	0.191	0.039	0.131	0.057	0.161	0.737	0.581	0.079	0.200
	50%	**0.030**	**0.110**	0.060	0.157	0.036	0.124	0.102	0.221	0.134	0.255	0.132	0.248	0.089	0.218	0.047	0.145	0.067	0.174	0.770	0.605	0.093	0.218
	Avg	**0.023**	**0.096**	0.052	0.145	0.027	0.107	0.070	0.181	0.104	0.218	0.093	0.206	0.062	0.177	0.036	0.126	0.051	0.150	0.717	0.570	0.071	0.188
ETTm2	12.5%	**0.017**	**0.076**	0.026	0.093	0.018	0.080	0.061	0.166	0.034	0.127	0.062	0.166	0.056	0.159	0.021	0.088	0.023	0.092	0.394	0.470	0.133	0.270
	25%	**0.019**	**0.080**	0.029	0.101	0.020	0.085	0.112	0.220	0.042	0.143	0.085	0.196	0.080	0.195	0.024	0.096	0.026	0.101	0.421	0.482	0.135	0.272
	37.5%	**0.021**	**0.087**	0.032	0.106	0.023	0.091	0.163	0.274	0.051	0.159	0.106	0.222	0.110	0.231	0.027	0.103	0.030	0.108	0.478	0.521	0.155	0.293
	50%	**0.024**	**0.091**	0.035	0.112	0.026	0.098	0.253	0.337	0.059	0.174	0.131	0.247	0.156	0.276	0.030	0.108	0.035	0.119	0.568	0.560	0.200	0.333
	Avg	**0.020**	**0.084**	0.031	0.103	0.022	0.088	0.147	0.249	0.046	0.151	0.096	0.208	0.101	0.215	0.026	0.099	0.029	0.105	0.465	0.508	0.156	0.292
ETTh1	12.5%	**0.039**	**0.138**	0.097	0.203	0.057	0.159	0.072	0.192	0.240	0.345	0.151	0.267	0.070	0.190	0.060	0.165	0.074	0.182	0.857	0.609	0.114	0.234
	25%	**0.053**	**0.156**	0.116	0.224	0.069	0.178	0.105	0.231	0.265	0.364	0.180	0.292	0.106	0.236	0.080	0.189	0.090	0.203	0.829	0.672	0.140	0.262
	37.5%	**0.068**	**0.174**	0.136	0.242	0.084	0.196	0.139	0.267	0.296	0.382	0.215	0.318	0.124	0.258	0.102	0.212	0.109	0.222	0.830	0.675	0.174	0.293
	50%	**0.087**	**0.197**	0.159	0.261	0.102	0.215	0.184	0.309	0.334	0.404	0.257	0.347	0.165	0.299	0.133	0.240	0.137	0.248	0.854	0.691	0.215	0.325
	Avg	**0.062**	**0.166**	0.127	0.233	0.078	0.187	0.125	0.249	0.284	0.373	0.201	0.306	0.117	0.246	0.094	0.201	0.103	0.214	0.842	0.682	0.161	0.279
ETTh2	12.5%	**0.033**	**0.116**	0.057	0.151	0.040	0.130	0.101	0.221	0.101	0.231	0.100	0.216	0.095	0.212	0.042	0.133	0.044	0.138	0.976	0.754	0.305	0.431
	25%	**0.037**	**0.122**	0.063	0.159	0.046	0.141	0.153	0.273	0.115	0.246	0.127	0.247	0.137	0.258	0.049	0.147	0.050	0.149	1.037	0.774	0.322	0.444
	37.5%	**0.042**	**0.131**	0.068	0.167	0.052	0.151	0.230	0.334	0.126	0.257	0.158	0.276	0.187	0.304	0.056	0.158	0.060	0.163	1.107	0.800	0.353	0.462
	50%	**0.049**	**0.141**	0.074	0.176	0.060	0.162	0.340	0.409	0.136	0.268	0.183	0.299	0.232	0.341	0.065	0.170	0.068	0.173	1.193	0.838	0.369	0.472
	Avg	**0.040**	**0.127**	0.066	0.163	0.049	0.146	0.206	0.309	0.119	0.250	0.142	0.259	0.163	0.279	0.053	0.152	0.055	0.156	1.079	0.792	0.337	0.452
Weather	12.5%	**0.024**	**0.039**	0.029	0.049	0.025	0.045	0.034	0.085	0.047	0.101	0.039	0.084	0.041	0.107	0.027	0.051	0.026	0.047	0.140	0.220	0.037	0.093
	25%	**0.026**	**0.043**	0.032	0.055	0.029	0.052	0.047	0.115	0.052	0.111	0.048	0.103	0.064	0.163	0.029	0.056	0.030	0.054	0.147	0.229	0.042	0.100
	37.5%	**0.028**	**0.046**	0.035	0.059	0.031	0.057	0.062	0.141	0.058	0.121	0.057	0.117	0.107	0.229	0.033	0.062	0.032	0.060	0.156	0.240	0.049	0.111
	50%	**0.030**	**0.047**	0.038	0.063	0.034	0.062	0.080	0.167	0.065	0.133	0.066	0.134	0.183	0.312	0.037	0.068	0.037	0.067	0.164	0.249	0.053	0.114
	Avg	**0.027**	**0.044**	0.034	0.057	0.030	0.054	0.056	0.127	0.055	0.117	0.052	0.110	0.099	0.203	0.032	0.059	0.031	0.057	0.152	0.235	0.045	0.104
1^{st} Count		**48**		0		0		0		0		0		0		0		0		0		0	

and 512. We re-test the results of PatchTST with the input length setting to 96.

C. Implementation Details

We provide the experiment configurations in Table III. All experiments are repeated three times, implemented in PyTorch [50] and conducted on a single Tesla V100 SXM2 32GB GPU.

Our method is trained with the L2 loss, using the ADAM optimizer with an initial learning rate of 10^{-3}, and Batch size is set to 32. The training process is early stopped after three epochs (patience=3) if there is no loss degradation on the valid set. The hyper-parameter λ is set to 100, and the hyper-parameter sensitivity analysis can be found in Section IV-H. For a fairer comparison, we fix the input length to 96 for all datasets (except 36 for ILI). By default, TS3Net contains 2 TF-Blocks.

All the baselines that we reproduced are implemented based on the configurations of the original paper or their official code. For a fair comparison, we design the same input embedding and final prediction layer for all base models. It is worth noting that the results of PatchTST reported in our paper are different from that of the original paper, due to the reason that our results are obtained with Lookback window equal to 96 while the original experimental results of PatchTST are obtained with Lookback window equal to 336.

Similar to the literature, we adopt the mean square error (MSE) and mean absolute error (MAE) to evaluate the long-term forecasting and imputation tasks.

D. Long-Term Forecasting Results

As shown in Table IV, TS3Net performs the best on 8 out of 9 datasets in long-term forecasting. Concretely, compared to the second-place model MICN, TS3Net achieves 11% relative error reduction on ETTh2; 7% relative reduction on ETTm2; 10% relative reduction on Traffic; and 11% relative reduction on Electricity. Note that TS3Net even provides remarkable improvements with 31% averaged MSE reduction on the ILI dataset, although it only learns from a short Lookback window (series length is set to 36 in the ILI dataset). Besides, we can also find that TS3Net makes further improvements as the length of prediction increases, showing its competitiveness in

TABLE VI

ABLATIONS ON MODEL ARCHITECTURE. WE REMOVE THE TD ARCHITECTURE AND THE TF-BLOCK FROM TS3NET SEQUENTIALLY TO DEMONSTRATE THE MODEL'S PERFORMANCE.

Datasets		ETTm1					Electricity					Traffic					Exchange				
Prediction Length		96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg
w/o TD	MSE	0.370	0.409	0.461	0.525	0.441	0.172	0.189	0.206	0.226	0.198	0.570	0.581	0.614	0.617	0.596	0.115	0.227	0.417	1.044	0.451
	MAE	0.391	0.410	0.439	0.471	0.428	0.265	0.282	0.287	0.310	0.286	0.324	0.337	0.358	0.362	0.345	0.250	0.349	0.479	0.790	0.467
w/o TF-Block	MSE	0.350	0.387	0.436	0.490	0.416	0.160	0.176	0.194	0.218	0.187	0.525	0.543	0.550	0.590	0.552	0.108	0.212	0.372	0.946	0.409
	MAE	0.378	0.397	0.425	0.455	0.414	0.267	0.280	0.298	0.315	0.290	0.317	0.320	0.331	0.340	0.327	0.228	0.323	0.445	0.749	0.436
w/o Both	MSE	0.384	0.424	0.479	0.552	0.459	0.195	0.214	0.233	0.260	0.226	0.594	0.624	0.634	0.637	0.622	0.134	0.250	0.435	1.048	0.467
	MAE	0.415	0.436	0.461	0.499	0.453	0.303	0.331	0.334	0.365	0.333	0.353	0.359	0.371	0.375	0.365	0.259	0.360	0.482	0.786	0.472
TS3Net	MSE	**0.324**	**0.358**	**0.404**	**0.458**	**0.386**	**0.153**	**0.168**	**0.180**	**0.201**	**0.176**	**0.464**	**0.495**	**0.507**	**0.536**	**0.501**	**0.085**	**0.181**	**0.334**	**0.887**	**0.372**
	MAE	**0.362**	**0.380**	**0.407**	**0.436**	**0.396**	**0.247**	**0.263**	**0.272**	**0.291**	**0.268**	**0.287**	**0.305**	**0.318**	**0.335**	**0.311**	**0.204**	**0.301**	**0.418**	**0.713**	**0.409**

TABLE VII

PERFORMANCE COMPARISON BETWEEN THE PROPOSED "TRIPLE DECOMPOSITION" AND THE CONVENTIONAL "TREND-SEASONAL DECOMPOSITION". IN THE TABLE, TSD-CNN AND TSD-TRANS ARE THE TREND-SEASONAL DECOMPOSITION MODELS USING CNN AND TRANSFORMER AS THE BACKBONE RESPECTIVELY.

Model		TSD-CNN					TSD-Trans					***TS3Net***				
PredictionLength		96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg
ETTm1	MSE	0.364	0.396	0.408	**0.451**	0.405	0.346	0.385	0.405	0.488	0.406	**0.324**	**0.358**	**0.404**	0.458	**0.386**
	MAE	0.384	0.407	0.410	0.439	0.410	0.378	0.399	0.411	0.453	0.410	**0.362**	**0.380**	**0.407**	**0.436**	**0.396**
ETTm2	MSE	0.184	0.257	0.325	0.392	0.290	0.190	0.255	0.305	**0.392**	0.286	**0.169**	**0.239**	**0.298**	0.399	**0.276**
	MAE	0.269	0.314	0.356	0.395	0.334	0.275	0.313	0.344	0.395	0.332	**0.253**	**0.297**	**0.335**	**0.393**	**0.320**
Exchange	MSE	0.096	0.189	0.382	0.946	0.403	0.096	0.190	0.341	0.913	0.385	**0.085**	**0.181**	**0.334**	**0.887**	**0.372**
	MAE	0.217	0.311	0.444	0.734	0.427	0.217	0.311	0.421	0.719	0.417	**0.204**	**0.301**	**0.418**	**0.713**	**0.409**

TABLE VIII

ROBUSTNESS ANALYSIS OF LONG-TERM FORECASTING. THE PARAMETER $\boldsymbol{\rho}$ INDICATES DIFFERENT PROPORTIONS OF NOISE INJECTION. WE IMPLEMENT IT ON THREE DATASETS: ETTH1, ETTH2 AND EXCHANGE.

Datasets		ETTh1					ETTh2					Exchange				
PredictionLength		96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg
$\boldsymbol{\rho} = 0\%$	MSE	0.397	0.436	0.468	0.468	0.442	0.290	0.374	0.419	0.429	0.378	0.085	0.181	0.334	0.887	0.371
(TS3Net)	MAE	0.417	0.436	0.449	0.465	0.441	0.339	0.391	0.432	0.445	0.401	0.204	0.301	0.418	0.713	0.409
$\boldsymbol{\rho} = 1\%$	MSE	0.398	0.438	0.469	0.467	0.443	0.294	0.377	0.423	0.447	0.385	0.091	0.189	0.365	0.1092	0.434
	MAE	0.418	0.438	0.451	0.466	0.443	0.344	0.396	0.437	0.459	0.409	0.213	0.309	0.435	0.793	0.438
$\boldsymbol{\rho} = 5\%$	MSE	0.400	0.439	0.470	0.471	0.445	0.301	0.384	0.430	0.464	0.395	0.107	0.205	0.379	1.404	0.524
	MAE	0.420	0.440	0.452	0.469	0.445	0.353	0.404	0.447	0.478	0.421	0.234	0.331	0.446	0.908	0.480
$\boldsymbol{\rho} = 10\%$	MSE	0.403	0.442	0.471	0.473	0.447	0.308	0.391	0.437	0.478	0.404	0.116	0.213	0.375	1.598	0.576
	MAE	0.423	0.443	0.454	0.471	0.448	0.361	0.412	0.453	0.490	0.429	0.246	0.334	0.448	0.971	0.499

long-term time-series forecasting. All above results show that TS3Net can cope well with a variety of time-series forecasting tasks in real-world applications.

E. Imputation Results

As shown in Table V, our proposed TS3Net exhibits the best performance on all datasets in the imputation task, which verifies the model capacity in capturing temporal variation from extremely complicated time series. Especially, compared to the second-place model (i.e., TimesNet), TS3Net achieves 15% relative reduction under all mask settings on the ETTm1 dataset; achieves 21% relative reduction under all mask settings on the ETTh1 dataset; and achieves 10% relative reduction under all mask settings on the Weather dataset.

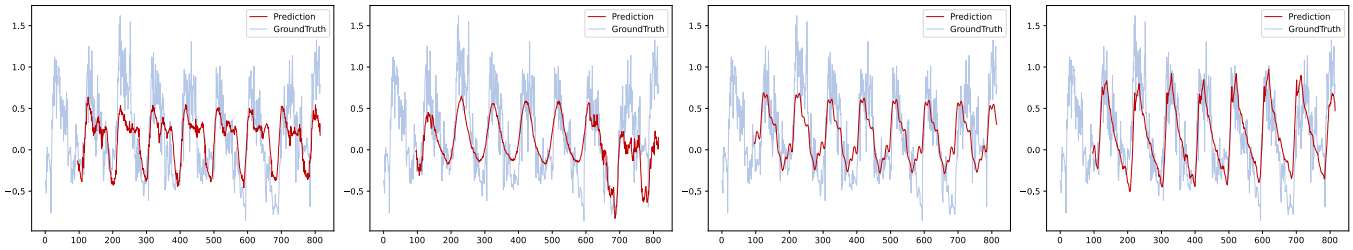
F. Ablation Study

To elaborate on the properties of our TS3Net, we conduct detailed ablations on the model's architecture. As shown in Table VI, we find that removing the Triple Decomposition architecture in TS3Net will cause significant performance degradation. These results may be because the dynamic spectrum fluctuation will affect the learning of the low-frequency distribution in the spectrum. Besides, in this paper, to adequately capture the dynamic spectrum pattern of the time series, we expand the time series into a *2D* tensor composed of continuous frequency components. Here we compare our design with the case of converting *1D* time series to *2D* tensor by replicating and concatenating only. From Table VI, we can find that the performance of *Removing TD* degenerates 14.3%;

TABLE IX

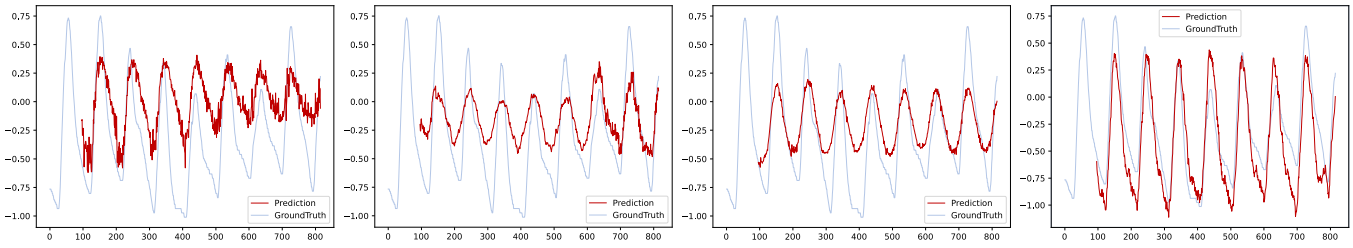
HYPER-PARAMETER SENSITIVITY OF LONG-TERM FORECASTING. THE HYPER-PARAMETER λ INDICATES THE NUMBER OF SPECTRAL SUB-BANDS IN THE TEMPORAL-FREQUENCY DISTRIBUTION. WE IMPLEMENT THE HYPER-PARAMETER EXPERIMENTS ON THREE DATASETS: ETTH1, ETTH2 AND EXCHANGE. GIVING CONSIDERATION TO BOTH EFFICIENCY AND PERFORMANCE, WE SET $\lambda = 100$ FOR LONG-TERM FORECASTING AND IMPUTATION.

Datasets		ETTh1					ETTh2					Exchange				
PredictionLength		96	192	336	720	Avg	96	192	336	720	Avg	96	192	336	720	Avg
$\lambda = 50$	MSE	0.397	0.436	0.470	0.466	0.442	0.292	0.376	0.422	0.453	0.386	0.088	0.182	0.343	0.956	0.392
	MAE	0.418	0.436	0.450	0.466	0.443	0.341	0.394	0.434	0.460	0.407	0.208	0.303	0.423	0.743	0.419
$\lambda = 100$ (default*)	MSE	0.397	0.436	0.468	0.468	0.442	0.290	0.374	0.419	0.429	0.378	0.085	0.181	0.334	0.887	0.371
	MAE	0.417	0.436	0.449	0.465	0.441	0.339	0.391	0.432	0.445	0.401	0.204	0.301	0.418	0.713	0.409
$\lambda = 150$	MSE	0.395	0.437	0.469	0.475	0.444	0.292	0.376	0.421	0.441	0.383	0.088	0.182	0.352	0.913	0.384
	MAE	0.416	0.438	0.451	0.470	0.444	0.341	0.394	0.433	0.453	0.405	0.208	0.303	0.427	0.719	0.414
$\lambda = 200$	MSE	0.395	0.438	0.466	0.477	0.444	0.292	0.377	0.422	0.445	0.384	0.087	0.181	0.346	0.901	0.379
	MAE	0.416	0.439	0.450	0.471	0.444	0.341	0.394	0.434	0.455	0.406	0.208	0.303	0.424	0.714	0.412



(a) TimesNet (b) MICN (c) DLinear (d) **TS3Net(ours)**

Fig. 3. Visualization on electricity transformer A's MULL (Middle UseLess Load) in the ETTm1 dataset with the predict-720 setting.



(a) TimesNet (b) MICN (c) DLinear (d) **TS3Net(ours)**

Fig. 4. Visualization on normalized OT (Oil Temperature) in the ETTm2 dataset with the predict-720 setting.

Removing TF-Block degenerates 7.8%, and *Removing Both* degenerates 27.5% on the Exchange dataset. Similar results are found in other datasets, which indicate the advantages of our design.

In addition, to further verify the effectiveness of the proposed "Triple Decomposition", we compare it with the conventional "Trend-Seasonal Decomposition" and show the results in Table VII. In the table, TSD-CNN uses the trend-seasonal decomposition and maintains the same backbone as TS3Net. TSD-Tran uses the trend-seasonal decomposition and a vanilla Transformer as the backbone. According to the table, TS3Net achieves 8% relative average error reduction on Exchange compared to the TSD-CNN; 5% relative reduction on ETTm1 compared to the TSD-Trans. Overall, TS3Net achieved the best performance on 13 over 15 compared results, which shows the superiority of the proposed triple decomposition.

G. Robustness Analysis

To demonstrate the robustness of the proposed TS3Net, we conduct a synthetic noise interference experiment as follows. Firstly, the proportion $\boldsymbol{\rho}$ of the input data was randomly selected to add noise following the distribution characteristics of the original signal. TS3Net will be trained on the noise-introduced datasets, and the metrics on the validation set were calculated. As shown in Table VIII, the MSE and MAE only increased slightly as the level of introduced noise increased (the performance degradation is less than 2% in ETTh1), which indicates that TS3Net shows good robustness in dealing with data with abnormal fluctuations.

H. Hyper-Parameter Sensitivity

As presented in Eq 6, the hyper-parameter λ is the number of sub-bands, which is used to adequately capture the variation of the time series in the low frequencies. We provide the sensitivity analysis for the hyper-parameter λ in Table IX. we can see that our proposed TS3Net presents stable performances under different choices of λ in all four tasks. One of the explanations is that the energy of the real-world time series is mainly concentrated on the low-frequency sub-band, and the high-frequency part is related to noise. Therefore, the proposed

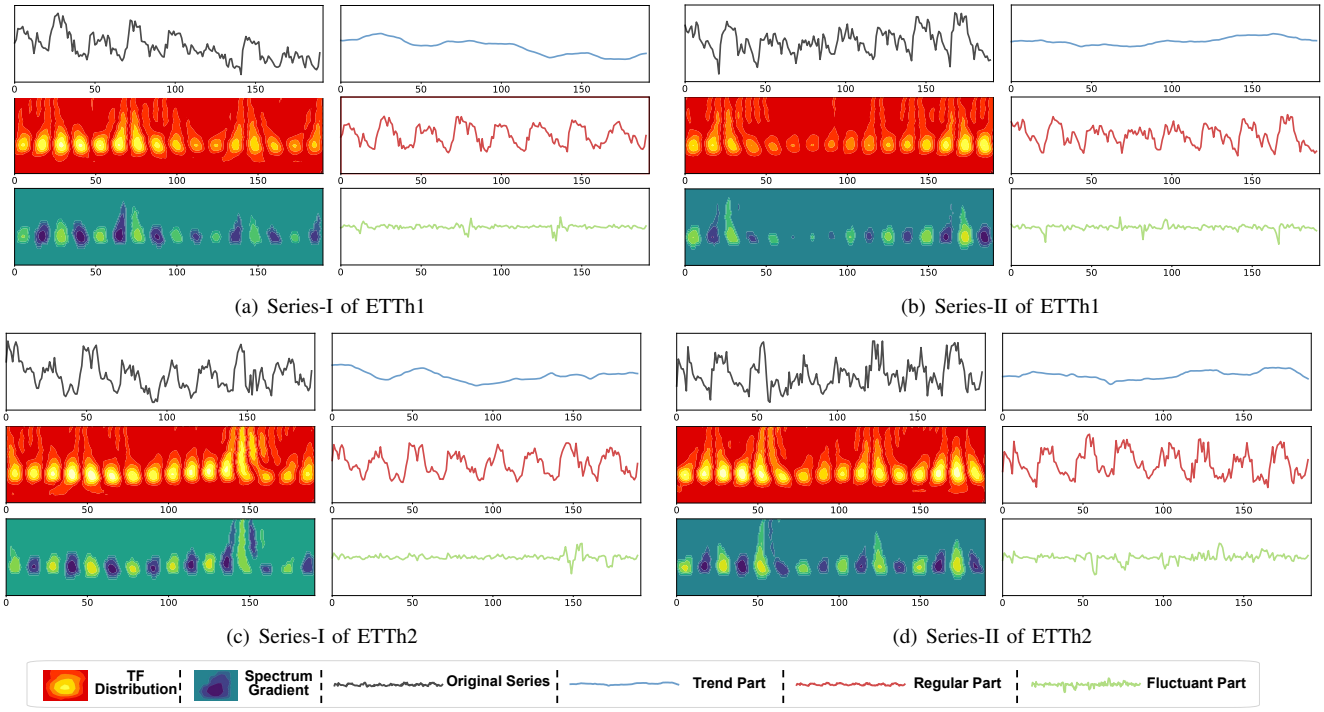


Fig. 5. Visualization of Triple Decomposition and three parts obtained (trend-, regular- and fluctuant-part) on ETTh1 and ETTh2, where the upper left subplot represents the original series of length 192, the warm-colored heat map is the TF distribution, and the cool-colored heat map demonstrates the spectral gradient. In addition, the right side shows trend, regular and fluctuant-part corresponding to the blue, red, and green curves.

TS3Net does not need to determine the main frequency component of the time series since it is not sensitive to the hyper-parameter λ.

Using table IX, we can verify the model's robustness with respect to the hyper-parameter λ. Different values of λ yield slightly different results. Concretely, when λ is set to 50, TS3Net performs worse because of the inadequate capacity to capture and separate complex multi-frequency sub-band patterns of the time series, especially in the low-frequency sub-bands. When λ continues to increase, the performance improves at the beginning, and stabilizes after reaching some threshold. TS3Net achieves almost the same performance when λ is equal to 100, 150, or 200. We set λ to 100 in this paper by default.

I. Visualization

To provide a visualized comparison among different models, we provide showcases of the long-term forecasting task, including the Electricity and ETTm2 datasets, as shown in Figure 3 and Figure 4. We visualize the prediction results for multiple variables in each dataset. Based on the figure, the predicted results of TS3Net (red curve) are very close to that of the original time series (blue curve), which shows the effectiveness of using TS3Net for time series prediction.

We further visualize the results of triple decomposition of several time series on the ETTh1 and ETTh2 datasets, which are shown in Figure 5. The left part of each figure shows the original time series, the TF-distribution, and the spectrum gradient derived with the proposed approach. The right part of each figure illustrates the results of triple decomposition of the original time series: the trend-part (blue curve) reflects the baseline value of a long-term series; the regular-part (red curve) reflects the stable periodicity of the curve; and the fluctuant-part (green curve) reflects the multi-periodicity dynamics of the time series. Specifically, in the regular-part, the sub-series corresponding to different periods show similar fluctuation shapes, which implies that this downstream vision network (CNNs) learns stable temporal patterns for the prediction task, and we believe that this is an important reason for the effectiveness of the triple decomposition and the TF-Block.

V. Conclusion

This paper proposed a task-general deep learning model called TS3Net for long-term series analysis. Based on the formulation of spectrum gradient from the frequency space, TS3Net adopts a triple decomposition method to decouple a long-term series into three components: trend-part, regular-part, and fluctuant-part. It introduces a temporal-frequency block (TF-Block) with multi-branch structure to learn deep representation of time series to capture the dynamic spectrum variations from the complex multi-periodic series. It processes the three decomposed components individually and integrates their results to form the final result for time series analysis. Extensive experiments based on six open datasets with ten baselines showed that TS3Net significantly outperformed the state-of-the-art methods on a variety of long-term time series analysis tasks.

REFERENCES

[1] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao, "MICN: Multi-scale local and global context modeling for long-term series forecasting," in *ICLR*, 2023.

[2] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *ICLR*, 2023.

[3] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *ICML*, 2022.

[4] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting," in *NeurIPS*, 2021.

[5] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Rethinking the stationarity in time series forecasting," in *NeurIPS*, 2022.

[6] Z. Wang, X. Xu, W. Zhang, G. Trajcevski, T. Zhong, and F. Zhou, "Learning latent seasonal-trend representations for time series forecasting," in *NeurIPS*, 2022.

[7] T. Zhou, Z. Ma, xue wang, Q. Wen, L. Sun, T. Yao, W. Yin, and R. Jin, "Film: Frequency improved legendre memory model for long-term time series forecasting," in *NeurIPS*, 2022.

[8] A. Seyfi, J.-F. Rajotte, and R. T. Ng, "Generating multivariate time series with common source coordinated GAN (COSCI-GAN)," in *NeurIPS*, 2022.

[9] Y. Li, X. Lu, Y. Wang, and D. Dou, "Generative time series forecasting with diffusion, denoise, and disentanglement," in *NeurIPS*, 2022.

[10] J. Jeon, J. KIM, H. Song, S. Cho, and N. Park, "Gt-gan: General purpose time series synthesis with generative adversarial networks," in *NeurIPS*, 2022.

[11] Y. Cui, S. Li, W. Deng, Z. Zhang, J. Zhao, K. Zheng, and X. Zhou, "Roi-demand traffic prediction: A pre-train, query and fine-tune framework," *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1340–1352, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260171606

[12] Y. Li, X. Lu, H. Xiong, J. Tang, J. Su, B. Jin, and D. Dou, "Towards long-term time-series forecasting: Feature, pattern, and distribution," *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1611–1624, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:255440671

[13] G. Li, X. Wang, G. S. Njoo, S. Zhong, S. H. G. Chan, C.-C. Hung, and W.-C. Peng, "A data-driven spatial-temporal graph neural network for docked bike prediction," *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 713–726, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:249010849

[14] R.-G. Cirstea, B. Yang, C. Guo, T. Kieu, and S. Pan, "Towards spatio- temporal aware traffic time series forecasting," *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2900–2913, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247778412

[15] R.-G. Cirstea, T. Kieu, C. Guo, B. Yang, and S. J. Pan, "Enhancenet: Plugin neural networks for enhancing correlated time series forecasting," *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 1739–1750, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235617661

[16] Z. Wang, T. Xia, R. Jiang, X. Liu, K.-S. Kim, X. Song, and R. Shibasaki, "Forecasting ambulance demand with profiled human mobility via heterogeneous multi-graph neural networks," *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 1751–1762, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235616020

[17] A. Saadallah, M. Tavakol, and K. Morik, "An actor-critic ensemble aggregation model for time-series forecasting*," *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 2255–2260, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235615720

[18] T. Mu, H. Wang, S. Zheng, Z. Liang, C. Wang, X. Shao, and Z. Liang, "Tsc-automl: Meta-learning for automatic time series classification algorithm selection," *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1032–1044, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260172370

[19] G. Li, B. Choi, J. Xu, S. S. Bhowmick, D. N. yin Mah, and G. L. Wong, "Ips: Instance profile for shapelet discovery for time series classification," *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 1781–1793, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251300190

[20] A. Yamaguchi, K. Ueno, and H. Kashima, "Learning evolvable time-series shapelets," *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 793–805, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251292042

[21] G. Li, B. Choi, J. Xu, S. S. Bhowmick, K.-P. Chun, and G. L. Wong, "Efficient shapelet discovery for time series classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, pp. 1149–1163, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:219463012

[22] T. Kieu, B. Yang, C. Guo, R.-G. Cirstea, Y. Zhao, Y. heng Song, and C. S. Jensen, "Anomaly detection in time series with robust variational quasi-recurrent autoencoders," *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 1342–1354, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251293217

[23] X. Chen, L. Deng, F. Huang, C. Zhang, Z. Zhang, Y. Zhao, and K. Zheng, "Daemon: Unsupervised anomaly detection and interpretation for multivariate time series," *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 2225–2230, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235616579

[24] X. Ma, P. Wang, B. Zhang, and M. Sun, "A multirate sensor information fusion strategy for multitask fault diagnosis based on convolutional neural network," *J. Sensors*, vol. 2021, pp. 9 952 450:1–9 952 450:17, 2021.

[25] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik, "Self-supervised contrastive pre-training for time series via time-frequency consistency," in *NeurIPS*, 2022.

[26] Z. Wang, Y. Zhou, R. Wang, T.-Y. Lin, A. Shah, and S. N. Lim, "Few-shot fast-adaptive anomaly detection," in *NeurIPS*, 2022.

[27] M. G. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. H. Leufen, A. Mozaffari, and S. Stadtler, "Can deep learning beat numerical weather prediction?" *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 379, 2021.

[28] N. Karmitsa, S. Taheri, A. M. Bagirov, and P. Mäkinen, "Missing value imputation via clusterwise linear regression," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, pp. 1889–1901, 2022.

[29] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," in *NeurIPS*, 2019.

[30] M. Liu, A. Zeng, M.-H. Chen, Z. Xu, Q. Lai, L. Ma, and Q. Xu, "Scinet: Time series modeling and forecasting with sample convolution and interaction," in *NeurIPS*, 2021.

[31] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," in *ICLR*, 2022.

[32] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W.-K. Wong, and W. chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NeurIPS*, 2015.

[33] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *ICLR*, 2023.

[34] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *SIGIR*, 2018.

[35] C. Challu, K. G. Olivares, B. N. Oreshkin, F. Garza, M. Mergenthaler, and A. Dubrawski, "N-hits: Neural hierarchical interpolation for time series forecasting," *arXiv preprint arXiv:2201.12886*, 2022.

[36] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *AAAI*, 2023.

[37] T. Zhang, Y. Zhang, W. Cao, J. Bian, X. Yi, S. Zheng, and J. Li, "Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures," *ArXiv*, vol. abs/2207.01186, 2022.

[38] S. E. Finder, Y. Zohav, M. Ashkenazi, and E. Treister, "Wavelet feature maps compression for image-to-image cnns," *ArXiv*, vol. abs/2205.12268, 2022.

[39] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context," *ArXiv*, vol. abs/2005.03191, 2020.

[40] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[41] Y. Fang, Y. Qin, H. Luo, F. Zhao, B. Xu, L. Zeng, and C. Wang, "When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks," *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 517–529,

2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260171424

[42] Q. Zhang, D. Guo, X. Zhao, L. Yuan, and L. Luo, "Discovering frequency bursting patterns in temporal graphs," *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 599–611, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260172197

[43] Y. Chen, H. Zhang, W. Sun, and B. Zheng, "Rntrajrec: Road network enhanced trajectory recovery with spatial-temporal transformer," *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 829–842, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254017729

[44] S. Guo, Y. Lin, L. Gong, C. Wang, Z. Zhou, Z. Shen, Y. Huang, and H. Wan, "Self-supervised spatial-temporal bottleneck attentive network for efficient long-term traffic forecasting," *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1585–1596, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260171784

[45] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," *CVPR*, 2022.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106

[47] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *AAAI*, 2021.

[48] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:4922476

[49] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *ICLR*, 2021.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.