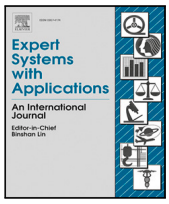
Expert Systems With Applications 262 (2025) 125588

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



ADMNet: An adaptive downsampling multi-frequency multi-channel network for long-term time series forecasting

Lele Yuan [a], Hua Wang [b], Fan Zhang [a,c,*]

[a] School of Computer Science and Technology, Shandong Technology and Business University, Yantai 264005, China
[b] School of Information and Electrical Engineering, Ludong University, Yantai 264005, China
[c] Shandong Future Intelligent Financial Engineering Laboratory, Yantai 264005, China

ARTICLE INFO

Keywords:
Long-term time series forecasting
Adaptive cyclic feature recognizer
Multi-frequency multi-channel network
Multi-period feature fusion

ABSTRACT

Long-term time series forecasting finds widespread applications in various domains such as energy, finance, and transportation. Decomposing time series into sub-sequences with distinct temporal relationships (periods) for analysis and modeling is an effective approach for long-term time series prediction. However, conventional decomposition techniques often rely on fixed-size parameter kernels for sliding averages, leading to inaccurate and unreasonable captures of the underlying periodicities in the time series. Additionally, it is essential to acknowledge that time series data often exhibits multi-periodic characteristics. In many time series prediction tasks, the value at a future time point is influenced by multiple periods within the time series. To accurately and flexibly capture the periodicities of time series, we propose an enhanced adaptive cyclic feature recognizer to automatically identify the periodic lengths for sliding average parameter kernel sizing. To fully exploit the multiple periodicities inherent in time series, we merge and smooth the sequences of features corresponding to the identified cycles, obtaining two frequency-blended cyclic feature fusion terms. Furthermore, we extract high-frequency random components to preserve finer details. Finally, we individually model the three cyclic feature fusion terms. In summary, we introduce the Adaptive Multi-Frequency Multi-Channel Network (ADMNet) designed to autonomously capture multi-periodic features. Experimental results show that, compared to the current state-of-the-art seven benchmark models, the proposed model achieves an average reduction in mean squared error (MSE) ranging from 6.19% to 22.75% across eight datasets, including ETT, Traffic, and Weather. This indicates that our model delivers superior predictive performance on multiple real-world datasets.

1. Introduction

Time series forecasting is a critical data analysis technique that involves analyzing and modeling historical data to predict future trends and behaviors. In recent years, models based on CNN (Convolutional Neural Network) (Livieris et al., 2020) and Transformer (Vaswani et al., 2017) have demonstrated outstanding predictive capabilities in time series modeling. Transformer-based models, in particular, often employ self-attention mechanisms, offering the advantage of addressing long-distance dependency issues and retaining information over extended ranges. This characteristic proves powerful in modeling scenarios involving large quantities of data with inherent relationships. Such models have found significant applications in various domains such as natural language processing (Nagy et al., 2023; Zhong et al., 2023), computer vision (Zhang, Chen, et al., 2023, 2024), and time series forecasting (Zhang, Guo, et al., 2023; Zhang, Wang, et al., 2024). However, it is essential to note that the computational nature of self-attention mechanisms introduces the challenge of quadratic time complexity. On the other hand, CNN-based models not only effectively capture local features in time series but also exhibit lower time complexity compared to Transformer-based models.

Time series data represents a unique sequential dataset that evolves over time. In recent years, methods focusing on the decomposition and re-modeling of time series have shown significant potential for improving prediction accuracy. Typically, time series decomposition involves breaking down the data into multiple categories based on temporal relationships, allowing the separate treatment of data with similar characteristics. SCINet (Liu, Zeng, et al., 2022) proposes a decomposition method based on odd and even elements, preserving temporal relationships to a large extent. Subsequently, odd and even sequences are individually processed using convolutional network downsampling to capture temporal features. The features of odd and even sequences

* Corresponding author at: School of Computer Science and Technology, Shandong Technology and Business University, Yantai 264005, China.
E-mail addresses: 2022410051@sdtbu.edu.cn (L. Yuan), 3410@ldu.edu.cn (H. Wang), zhangfan@sdtbu.edu.cn (F. Zhang).

https://doi.org/10.1016/j.eswa.2024.125588
Received 31 January 2024; Received in revised form 7 October 2024; Accepted 16 October 2024
Available online 23 October 2024

are then complemented through an interaction mechanism. This classification approach for time series allows the model to handle data with similar features separately, reducing model complexity while enhancing accuracy. Autoformer (Chen et al., 2021) introduces a progressive decomposition prediction architecture, employing a moving average decomposition into trend and periodic components in the sequence decomposition unit. The encoder then aggregates information on each periodic component through a similar process, with the trend and periodic components subsequently modeled separately in the decoder. FEDformer (Zhou et al., 2022) combines seasonal trend decomposition with Transformer, proposing a method of randomly selecting frequency domain information to eliminate redundant information obtained from Fourier transform representations. MICN (Wang et al., 2022) performs multi-scale decomposition on time series using parameters with different sizes to obtain trend and seasonal components. Subsequently, a combined convolutional network is employed to extract local features and global correlations for modeling the seasonal components. Both Transformer-based and CNN-based models, as described above, focus on modeling the mid-to-low-frequency range during the time series decomposition process, neglecting detailed information inherent in higher frequency components (as shown in Fig. 1). Additionally, the use of fixed-size parameter kernels for sliding averages in the decomposition process lacks flexibility and leads to low accuracy, thereby impacting the precision of model predictions.

A single time series often encompasses various latent cycles, exhibiting significant differences in periodicity among different time series. In the context of time series forecasting tasks, predicting the value of a variable at a specific time step is influenced by multiple cycles. For instance, the temperature at a future time point is not only related to the time of day (T_1) but is also influenced by various cycles such as seasons and months ($T_2, \ldots, T_n$). TimesNet (Wu et al., 2022) adeptly leverages the multi-cycle characteristics of time series, decomposing it into intra-cycle and inter-cycle variations. Creatively, it embeds one-dimensional time series data into a two-dimensional tensor, employing two-dimensional kernels for modeling. It is well-known that different time series exhibit different periodicities, and in most cases, complex time series possess multi-periodic characteristics. Therefore, accurately identifying the periods of a time series is crucial for seasonal-trend-period decomposition. Existing time series decomposition methods, such as Autoformer and FEDformer, decompose time series into trend, period, and seasonal components. However, when applying $AvgPool(\cdot)$ as shown in Eq. (1), the kernel size is fixed and manually set, which limits these models to capturing only a single periodic mode. When dealing with different time series, the kernel size must be set based on prior experience, which lacks adaptability and often leads to inaccuracies, thereby affecting the accuracy of the prediction results. MICN addresses the influence of multiple periods on prediction results by designing a multi-scale hybrid decomposition block, as shown in Eq. (2), which extracts multiple trend and periodic components. However, the kernel size in this model is still fixed, and it fails to accurately identify the periods. To overcome this limitation, we propose an enhanced adaptive periodic feature recognizer, which can automatically identify multiple periods in a time series. As shown in Eq. (6), we adaptively set the period length p_s as the kernel size for $AvgPool(\cdot)$. This not only allows for accurate recognition of different periods in the time series but also addresses the shortcomings of existing models with fixed kernel sizes, offering an innovative approach to time series decomposition.

$$\begin{aligned} X_t &= AvgPool(Padding(X)) \\ X_s &= X - X_t \end{aligned} \tag{1}$$

$$\begin{aligned} X_t &= mean(AvgPool(Padding(X))_{\text{kernel}_1}, \ldots, \\ &AvgPool(Padding(X))_{\text{kernel}_n}) \\ X_s &= X - X_t \end{aligned} \tag{2}$$

Traditional time series forecasting models often struggle to capture the latent multi-periodicity within time series due to their reliance on fixed-size parameter kernels, which limits their performance. Furthermore, focusing on a single frequency band typically fails to fully represent the complexity of time series data. To address these limitations, this paper introduces ADMNet, which captures multi-periodic features across different frequency bands in time series through an enhanced adaptive period feature recognizer. In addition, to handle unexpected noise in the time dimension while integrating the latent multi-periodic information of the time series, multiple extracted periodic features are fused and denoised. This process results in two low- and mid-frequency periodic feature fusion components and one high-frequency stochastic component. The low-frequency periodic feature fusion component is obtained by selecting the S frequencies with the largest amplitudes through Fourier transform, using the corresponding period lengths of these frequencies as the kernel sizes for average pooling to process the original time series. The resulting sequences of different periods are then smoothed through fusion to obtain the low-frequency periodic feature fusion component, which retains the lower frequency components of the time series and reflects long-term periodic fluctuations, such as seasonal changes or annual trends. The mid-frequency component is obtained by subtracting the low-frequency component, which is derived only through average pooling, from the original time series, and then smoothing and fusing the resulting sequence. The mid-frequency periodic feature fusion component represents the main periodic variations within the time series, such as monthly or quarterly fluctuations. Finally, the residuals obtained by subtracting both the mid-frequency and low-frequency periodic feature fusion components from the original time series constitute the high-frequency stochastic component. This high-frequency stochastic component captures the rapidly changing elements or noise within the original series, which typically lack clear periodicity and manifest as high-frequency fluctuations or randomness. In time series forecasting, capturing the complexity of the data often requires more than just single-frequency or single-period features. Thus, the fusion of low-, mid-, and high-frequency periodic features helps enhance the model's predictive power. As illustrated in Fig. 1, high-frequency components exhibit larger fluctuations within a unit time and possess stronger periodicity. To capture the detailed features and interdependencies among various cycle subsequences, we introduce a composite convolution module. Finally, information merging among different cycles is achieved through a Merge operation. The low-frequency fused item exhibits smoother fluctuations, indicating a higher correlation between future time points and historical values. Consequently, we employ a Multilayer Perceptron (MLP) network for prediction. In summary, the main contributions of this paper are as follows:

- We introduce an enhanced adaptive cycle feature recognizer to dynamically identify the cycles in time series, thereby determining the parameter kernel size. This enables the extraction of feature sequences associated with different latent cycles.
- To achieve the fusion of multi-cycle features, we merge and smooth the extracted feature sequences from different cycles, resulting in two fused items representing mid-to-low-frequency periodic features and a high-frequency random item.
- We propose a model framework named ADMNet, which dynamically recognizes the various cycles in time series, adaptively explores their latent cycle features, and models multi-frequency information separately. Ultimately, this leads to the aggregation of information for enhanced analysis.

2. Related work

2.1. Time series forecasting

Time series data is typically arranged in chronological order and exhibits temporal dependencies. Time series forecasting utilizes historical data to predict future values, helping decision-makers formulate

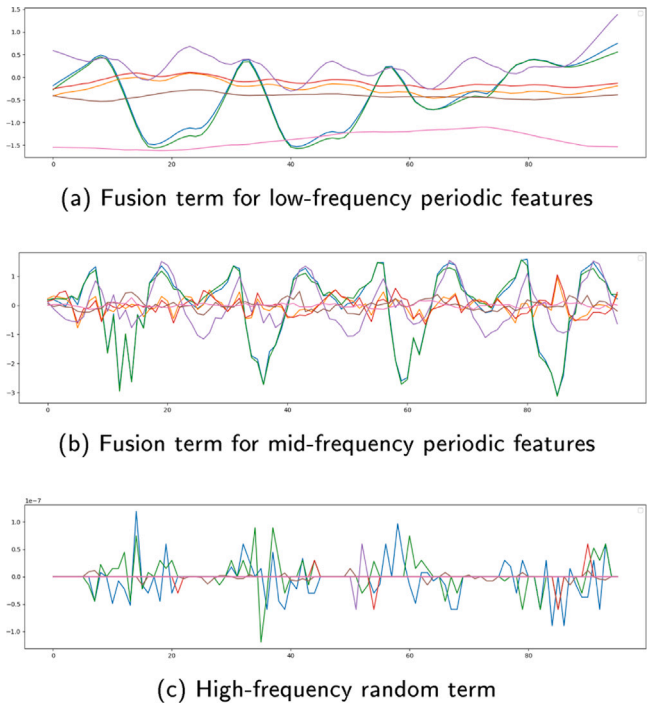


Fig. 1. Decomposition of multivariate time series into three frequency bands.

strategies. This field has a wide range of applications, such as traffic forecasting (Khaled et al., 2022), weather forecasting (Huang et al., 2024), oil temperature prediction (Huang & Liu, 2024), influenza monitoring and analysis, geological monitoring (Bi et al., 2021), and finance (Liu, Guo, et al., 2022). With increasing data volumes, traditional methods such as moving average (Satrio et al., 2021) and ARIMA (Sharma et al., 2020) often perform poorly when dealing with complex time series, leading to the development of machine learning (Cervantes et al., 2020; Quan et al., 2022; Sheykhmousa et al., 2020) and deep learning methods (Bi et al., 2021; Chen et al., 2023). In recent advancements in time series forecasting, several state-of-the-art models have been proposed. For example, Wang et al. (2023) introduces a method that combines global modeling and local feature extraction, where global attention is computed through a frequency-domain parity correction block, and multi-scale convolutions extract local features within periods. A dual-branch structure is used to model seasonality and trends separately, improving both forecasting performance and model interpretability. Wu et al. (2024) employs adaptive learning and a niche-based backtracking search algorithm to automatically optimize LSTM parameters. Abbasimehr et al. (2024) decomposes chaotic time series into multiple sub-series at different frequencies, fits a model to each sub-series, and aggregates the results to generate the final prediction. Geng et al. (2024) proposes a multi-attention network with redundant information filtering, capturing temporal dependencies in multivariate time series through multi-level feature fusion and spatial-semantic attention mechanisms, thereby improving the accuracy and interpretability of multi-horizon forecasting.

Many transformer-based models, such as Autoformer (Chen et al., 2021), propose a self-correlation mechanism for trend-period decomposition, which is embedded within the model's internal computations rather than being used solely as a preprocessing step. This mechanism allows the model to progressively extract long-term trend information from time series and optimize intermediate prediction results through decomposition and recursive adjustment. Although Autoformer can separate trend and seasonal components through trend-seasonality decomposition, this mechanism only handles a single dominant period by default. This single-decomposition strategy means that when a time series contains multiple periods (*e.g.*, daily, weekly, monthly cycles), Autoformer may fail to accurately capture all periodic features, extracting only the most prominent one while neglecting others. FEDformer (Zhou et al., 2022) enhances long-range time series forecasting by introducing a frequency enhancement module and a seasonal-trend decomposition mechanism. Its innovation lies in combining Fourier and wavelet transforms to extract global features of time series in the frequency domain, while a random mode selection strategy reduces computational complexity, lowering the complexity of the attention mechanism from quadratic to linear. The seasonal-trend decomposition allows the model to separate long-term trends and periodic features, improving both prediction accuracy and efficiency. However, this model has limitations in handling multi-periodic time series, primarily because the decomposition mechanism only captures a single periodic mode, lacking adaptability to complex multi-periodic variations. Additionally, the mechanism relies on fixed seasonality and trend assumptions, which may reduce the accuracy of the decomposition when dealing with frequently changing or nonlinear time series features. PatchTST (Nie et al., 2022) improves time series forecasting performance through patch segmentation and self-supervised learning, but its design lacks the capability for adaptive extraction of periodic features. Although patch segmentation effectively handles local information in time series, the model is not specifically optimized for periodic patterns and cannot identify complex multi-periodic features. In terms of adaptive periodic feature extraction, PatchTST does not deeply capture the periodic patterns at different frequencies in the time series, which may lead to reduced prediction accuracy in scenarios with multi-periodic data. Furthermore, while PatchTST significantly reduces computational complexity, it still relies on fixed patch sizes, limiting its ability to handle different periodic and frequency patterns. In contrast, ADMNet flexibly captures multi-periodic features through an adaptive periodic feature recognizer without relying on the inherent properties of time series or specific decomposition methods. Our model employs an adaptive feature recognition mechanism, effectively handling data with different periodicities and surpassing existing models in both accuracy and robustness.

2.2. Convolutional neural networks

CNN extracts features from input data through convolution operations, making it particularly suitable for capturing local patterns in time series data. Its parameter-sharing and local connection characteristics reduce the number of model parameters, lowering the risk of overfitting and improving computational efficiency. The advantages of CNN in time series forecasting include:

- Local feature extraction. Excellent at identifying periodic and trend features.
- Efficient parameter sharing mechanism. Capable of handling long sequences without increasing computational complexity.
- Translation invariance. Able to recognize recurring patterns in time series.

Many improved CNN models have achieved breakthroughs in fields such as speech recognition, traffic flow forecasting, and stock prediction (Liu, Ma, et al., 2022). WaveNet (Oord et al., 2016) uses dilated convolutional networks to increase the receptive field, improving the processing of speech signals, but its computational cost is high for long time series due to the complexity of its convolutional layers. ST-ResNet (Zhang et al., 2017) combines spatiotemporal convolution and residual networks, enhancing the fusion of spatiotemporal dependency features, though the fusion of different feature scales remains challenging. SCINet (Liu, Zeng, et al., 2022) handles multi-scale features of time series through sampling convolution operations, but its simple odd-even decomposition overlooks the handling of multi-periodic features, limiting its predictive ability for complex periodic sequences. MICN uses a multi-input convolutional network to jointly learn from

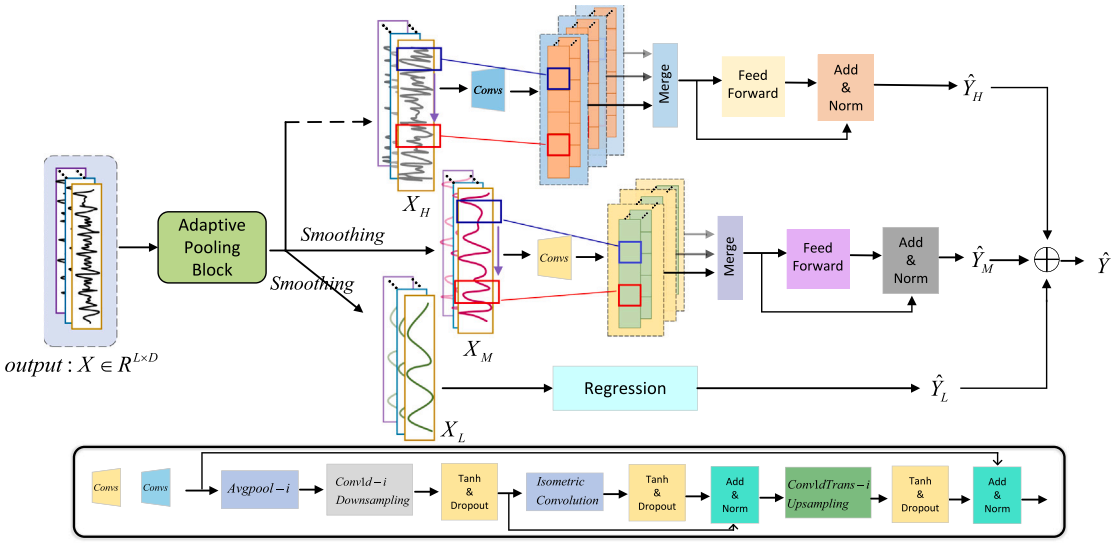


Fig. 2. Architecture of ADMNet. The high-frequency and mid-frequency components employ composite convolution for multi-level feature extraction, capturing intricate patterns embedded within them. The low-frequency components are subjected to prediction using a Multi-Layer Perceptron (MLP), and the final prediction result is obtained by summing the predictions from each channel.

multiple data sources, but its complex structure limits the model's interpretability. In contrast, ADMNet adopts a multi-scale decomposition and reconstruction strategy, flexibly handling various types of periodic data, thereby improving both the accuracy and robustness of predictions.

3. Methodology

This paper introduces a time series forecasting method named ADMNet, designed to flexibly capture different cycle features in time series for more effective decomposition. To address the limitation of existing models in inflexibly selecting parameter kernel sizes, we propose an enhanced adaptive cycle feature recognizer. This recognizer considers various cycle lengths from the original time series as parameter kernel sizes, performing downsampling to extract feature components corresponding to different cycles. To integrate information features from different cycles and mitigate the impact of noise, we conduct a fusion and smoothing process on the downsampling results, yielding low-frequency fusion components, mid-frequency fusion components, and high-frequency random components. We construct a multi-frequency, multi-channel network structure, where different channels extract features from different frequency components. The final prediction is accomplished through a feedforward network, with predictions from various frequency components aggregated to yield the ultimate prediction result.

3.1. Notations

The problem of multivariate time series forecasting can be defined as follows: predicting future values of a time series $\mathcal{X}_{L+1:L+O} = \{x_{L+1}, \ldots, x_{L+O}\} \in \mathbb{R}^{O\times D}$ of length O based on a given historical time series $\mathcal{X}_{1:L} = \{x_1, \ldots, x_L\} \in \mathbb{R}^{L\times D}$ of length L. Here, D represents the dimensionality of the time series, and when $D > 1$, it indicates a multivariate time series. x_t denotes the value at time step t.

3.2. ADMNet

Fig. 2 illustrates the overall architecture of ADMNet. The input time series $\mathcal{X} \in R^{L\times D}$ undergoes processing by the enhanced adaptive periodic feature recognizer (as shown in Fig. 3), yielding multiple feature components containing distinct periodic characteristics of the original time series. Following fusion and smoothing processing, two fused terms of periodic features (denoted as $\mathcal{X}_M$ and $\mathcal{X}_L$) corresponding to different frequency bands, along with a high-frequency random term (denoted as $\mathcal{X}_H$), are obtained. Distinct methods for feature extraction and result forecasting are applied to features in different frequency bands. Finally, the predictions from each frequency band are summed to yield the ultimate forecasting result.

3.2.1. Enhanced adaptive periodic feature recognizer

In many conventional decomposition methods, parameters for kernel size are typically fixed based on empirical considerations. However, different time series data often exhibit distinct periodic features, making it unreasonable to uniformly select the same kernel size for diverse time series data. To address this issue, as illustrated in Fig. 3, we utilize Fast Fourier Transform (FFT) to transform time series into the frequency domain for analysis. We employ a masking strategy to filter out the top S frequencies with the highest amplitudes. The main purpose of selecting S frequencies here is that significant periodicities in a time series often correspond to dominant frequency components with larger amplitudes. This is because recurring patterns in the time series appear as prominent amplitudes at specific frequencies in the frequency domain. Therefore, the primary goal of choosing S frequencies is to ensure that we capture the most significant periodic features within the time series. Based on the selected frequencies, we compute the potential periodic lengths of the time series, which are then used as the kernel size for downsampling. The resulting feature components encompass distinctive characteristics of various periods. Here, "adaptive" refers primarily to the ability of the adaptive periodic feature recognizer to selectively choose the period length of a time series as the kernel size for the convolution operation. The focus is on dynamically identifying the period length and then using it as the kernel size. Specifically, the process starts by using FFT to calculate the amplitude at different frequencies of the time series, followed by averaging the values over the D dimension to obtain the final amplitude, which corresponds to S frequencies:

$$\begin{aligned} Amp = Mean(&Amplitude(FFT(x))_1, \\ &Amplitude(FFT(x))_2, \\ &\cdots, \\ &Amplitude(FFT(x))_D) \end{aligned} \tag{3}$$

$$\{freq_1, \ldots, freq_s\} = \underset{freq_* \in \{1,\ldots,\lfloor \frac{T}{2} \rfloor\}}{\arg Tops}(Amp) \tag{4}$$

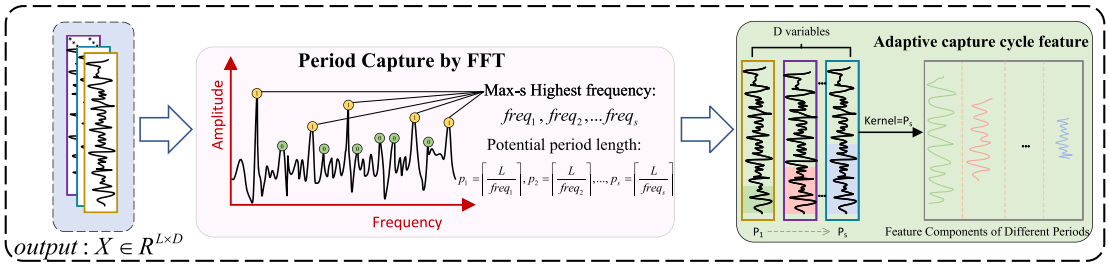


Fig. 3. Enhanced adaptive periodic feature recognizer.

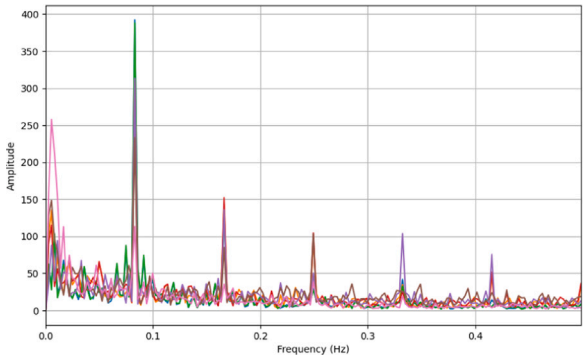


Fig. 4. Frequency amplitude distribution.

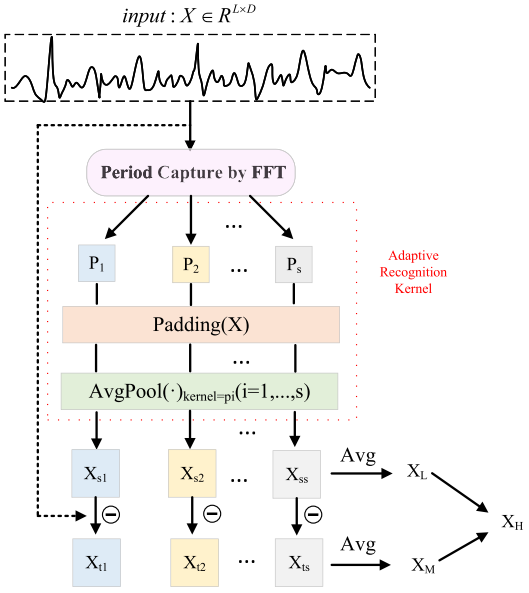


Fig. 5. Adaptive identification of kernel parameters and periodic feature fusion.

Where $Amplitude(\cdot)$ indicates the computation of the amplitude for frequency domain components. $freq_s$ denotes the frequencies corresponding to the top S selected amplitudes. The spectrum of most real-world time series data exhibits sparsity, meaning that the signal's energy is primarily concentrated in only a few frequencies, while the amplitudes of most other frequencies are small, potentially representing noise or minor periodic components. Therefore, a reasonable value for S should be small enough to capture the main periodic characteristics of the signal while filtering out noise. In our experiments, the S value was determined by observing the spectral distribution of the dataset. By sorting the frequency amplitudes in descending order, we identified the turning point where there was a significant change in amplitude to set the S value, as shown in Fig. 4.

The length of cycle $\{p_1, p_2, \ldots, p_s\}$ is calculated based on the selected frequencies as follows:

$$p_s = \left\lceil \frac{L}{freq_s} \right\rceil, s \in \{1, 2, \ldots, S\} \tag{5}$$

According to Eq. (5), we obtain $kernel = p_s, s \in \{1, 2, \ldots, S\}$. For the input multivariate time series (MTS) $\mathcal{X} \in R^{L \times D}$, applying $AvgPool(\cdot)_{kernel=p_s}$ yields S low-frequency terms (periodic feature sequences) $\mathcal{X}_S \in R^{L_{(S)} \times D}, L_{(S)} = \left\lfloor \frac{L}{p_S} \right\rfloor$ containing various periodic features. Subsequently, mid-frequency terms are derived from the low-frequency terms, as outlined in Eq. (6). This process achieves the objective of adaptively recognizing pooling window sizes to capture different periodic features of the time series. Fig. 5 provides a more intuitive illustration of this process.

$$\begin{aligned} \mathcal{X}_s &= AvgPool\left(Padding\left(\mathcal{X}\right)\right)_{kernel=p_s} \\ \mathcal{X}_t &= \mathcal{X} - \mathcal{X}_s \end{aligned} \tag{6}$$

3.2.2. Multi-frequency multi-channel network

To effectively leverage the diverse periodic features of the time series for more accurate forecastings, we perform fusion and smoothing on the obtained periodic feature sequences $\mathcal{X}_s$ from the enhanced adaptive periodic feature recognizer, resulting in three fused terms of periodic features denoted as $\mathcal{X}_H, \mathcal{X}_M, \mathcal{X}_L \in R^{L \times D}$ corresponding to different frequency bands. Subsequently, we model each fused term separately, establishing a multi-channel forecasting network. The detailed procedure of multi-frequency decomposition is described in Eq. (7). Based on the identified period lengths $(p_1, \ldots, p_s)$, these lengths are used as the kernel sizes for the AvgPool() operation. The input sequence $\mathcal{X}$ undergoes s iterations of AvgPool(), resulting in $\mathcal{X}_s$, which contains s periodic feature sequences. The $\mathcal{X}_t$ represents the extracted mid-frequency components, also containing s sequences. According to Eq. (7), both $\mathcal{X}_s$ and $\mathcal{X}_t$ undergo fusion and smoothing to obtain the low and mid-frequency periodic feature fusion components $\mathcal{X}_M$ and $\mathcal{X}_L$. The smoothing operation in the equations is based on averaging, which is the simplest and most direct denoising method while preserving the trend of the data. Although it seems straightforward, this smoothing process retains the periodic features across different frequency bands. This operation can eliminate faster fluctuations while retaining the main signal information, resulting in a more stable trend and periodic changes. After removing the low and mid-frequency components, the remaining portion is the high-frequency random component $\mathcal{X}_H$, which typically includes short-period fluctuations and other complex patterns. This concept is akin to a residual term, where the high-frequency, more random part is obtained by subtracting the smoothed low and mid-frequency components from the original signal. While $\mathcal{X}_H$ may contain noise, it may also encompass some short-term fluctuations or other complex patterns. Therefore, we do not simply regard $\mathcal{X}_H$ as noise; rather, we extract its underlying information through further modeling (convolution operations). In particular, in many practical applications, the short-term fluctuations contained in the high-frequency component are crucial for predictive outcomes. For example, in financial

time series, high-frequency components may reflect short-term market turbulence.

$$\begin{aligned}\mathcal{X}_M &= Avg_{i=1}^{S}\left(\mathcal{X}_t\right)\\ \mathcal{X}_L &= Avg_{i=1}^{S}\left(\mathcal{X}_s\right)\\ \mathcal{X}_H &= \mathcal{X} - \mathcal{X}_M - \mathcal{X}_L\end{aligned} \tag{7}$$

For the low-frequency periodic feature fusion term $\mathcal{X}_L$, which exhibits stationary fluctuations, we employ a Multi-Layer Perceptron (MLP) for prediction. As illustrated in Fig. 1, the mid-frequency periodic feature fusion term $\mathcal{X}_M$ and the high-frequency random term $\mathcal{X}_H$ contain more periodic information. Therefore, we utilize a composite convolution ($Convs$) for multi-level extraction of potential complex patterns. The one-dimensional convolution $Conv1d$ is employed to extract local features, followed by the use of a global convolution ($IsometricConv$) to capture its entire periodic features. To explore connections between different periods, multiple convolution kernels of different sizes are employed to capture more intricate underlying patterns. The network architecture for $\mathcal{X}_M$ and $\mathcal{X}_H$ is identical, but they use different convolution kernels.

Taking the modeling process of the mid-frequency feature fusion term $\mathcal{X}_M$ as an example, the embedding of the values of $\mathcal{X}_M$ involves a zero-padding approach, followed by the addition of positional embeddings and temporal embeddings. This process is as follows:

$$\mathcal{X}_M^{emb} = PE + TE + VE(Concat(\mathcal{X}_M, \mathcal{X}_{zero})) \tag{8}$$

Where $\mathcal{X}_M^{emb} \in R^{(L+O)\times D}$. PE represents positional embedding. VE represents value embedding. The temporal embeddings (TE) involve encoding numerical representations for attributes such as the year, month, and day of the time series, followed by normalization. After X_M^{emb} undergoes a simplification process through $AvgPool$, it utilizes $Conv1d$ to capture the fused local features of periodic patterns. This process is as follows:

$$\mathcal{Y}_M^{local,i} = Conv1d\left(AvgPool\left(Padding\left(\mathcal{X}_M^{emb}\right)\right)_{kernel=i}\right)_{kernel=i} \tag{9}$$

where $i \in \{\frac{L}{4}, \frac{L}{8}, \ldots\}$ represents the sizes of convolutional kernels in different branches. $\mathcal{Y}_M^{local,i} \in R^{\frac{(L+O)}{i}\times D}$ denotes the fused periodic local feature sequence obtained after down-sampling.

Additionally, we employ a convolutional kernel spanning the entire length of the feature sequence to capture global periodic features. To maintain the sequence length equal to the initial length, a transpose convolution ($Conv1dTrans$) is subsequently applied to upsample the results of the global convolution ($IsometricConv$). The specific implementation process is outlined as follows:

$$\begin{aligned}\mathcal{Y}'_M &= Dropout\left(Tanh\left(IsometricConv\left(Tanh\left(\mathcal{Y}_M^{local,i}\right)\right)\right)\right)\\ \mathcal{Y}'_M &= Norm\left(\mathcal{Y}_M^{local,i} + \mathcal{Y}'_M\right)\\ \mathcal{Y}_M &= Dropout\left(Tanh\left(Conv1dTrans\left(\mathcal{Y}'_M\right)\right)\right)\\ \mathcal{Y}_M &= Norm\left(\mathcal{X}_M^{emb} + \mathcal{Y}_M\right)\end{aligned} \tag{10}$$

where $\mathcal{Y}'_M \in R^{\frac{L+O}{i}\times D}$ represents the complete periodic feature sequence through $IsometricConv$. $\mathcal{Y}_M \in R^{(L+O)\times D}$ represents the output of a specific branch. Subsequently, we use two-dimensional convolution to merge the results from different branches. Finally, the forecasting of the mid-frequency fusion term is accomplished through a feedforward neural network, as shown in Eq. (11). We represent $\hat{\mathcal{Y}}_H$, $\hat{\mathcal{Y}}_M$, $\hat{\mathcal{Y}}_L$ as the forecasting results for the high-frequency random term, mid-frequency fusion term, and low-frequency feature fusion term, respectively. The ultimate forecasting result is the sum of these three components, denoted as $\hat{\mathcal{Y}} \in R^{O\times D}$.

$$\begin{aligned}\mathcal{Y}_M^{merge} &= Conv2d\left(\mathcal{Y}_M\right)\\ \hat{\mathcal{Y}}_M &= Norm\left(\mathcal{Y}_M^{merge} + FeedForward\left(\mathcal{Y}_M^{merge}\right)\right)\end{aligned} \tag{11}$$

$$\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_H + \hat{\mathcal{Y}}_M + \hat{\mathcal{Y}}_L \tag{12}$$

Algorithm 1 ADMNet

Input:
Time-series: $\mathbf{X} \in \mathbb{R}^{L\times D}$
Model parameters $\boldsymbol{\Theta}$
Frequency threshold **S**
Mask factor ***factor***
Batch size **B**
Output:
Time series prediction $\hat{\mathbf{Y}}$

repeat
  Each epoch
  for each mini-batch $\boldsymbol{X}^{(i)}$ from **X** **do**
    Transform to Frequency Domain: $Frequencies = FFT(X^{(i)})$
    Compute Amp and sort with Eq. (3) and (4)
    if $|freq| > factor$ **then**
      Retain $freq$
    else
      Delete
    end if
    Compute time series period and parameter kernel with Eq. (5) (6)
    Compute the multi-band periodic feature fusion term with Eq. (6) (7)
    Forecasting with Eq. (8) (9) (10) (11) (12)
  end for
until stopping criteria is met.

4. Experiment

In this section, we introduce the datasets, evaluation metrics, comparative models, comparative experiments and their visualization used in this study. We conduct extensive comparative experiments between our model and several state-of-the-art models across real-world datasets in domains such as electricity, traffic, exchange-rates, and weather. Additionally, we perform ablation experiments to validate the effectiveness of our components. We performed Fast Fourier Transform (FFT) analysis on each dataset to identify significant periodic features. This step helps us better adjust the model's hyperparameters, such as kernel size, to effectively capture the periodic patterns in the data.

4.1. Dataset description

To validate the effectiveness of ADMNet, we evaluated the model on nine real-world benchmark datasets:

- ETT: Comprising two hourly-level datasets and two 15-minute-level datasets, each dataset encompasses load characteristics of seven types of oil and power transformers from July 2016 to July 2018. Specifically, {ETTh1, ETTh2} are datasets partitioned at an hourly granularity, while {ETTm1, ETTm2} are datasets partitioned at a 15-minute granularity. The ETT dataset exhibits significant seasonal and periodic characteristics, making it suitable for evaluating a model's ability to handle complex data with both long-term and short-term variations.
- Exchange: Daily exchange rate records from eight different countries spanning from 1990 to 2016. Exchange rate data exhibits high volatility and noise characteristics.
- Electricity: Records the hourly electricity consumption of 370 customers from 2012 to 2014. Electricity load data is influenced not only by weather but also by daily usage patterns, making it suitable for evaluating a model's ability to capture complex multi-periodic characteristics.

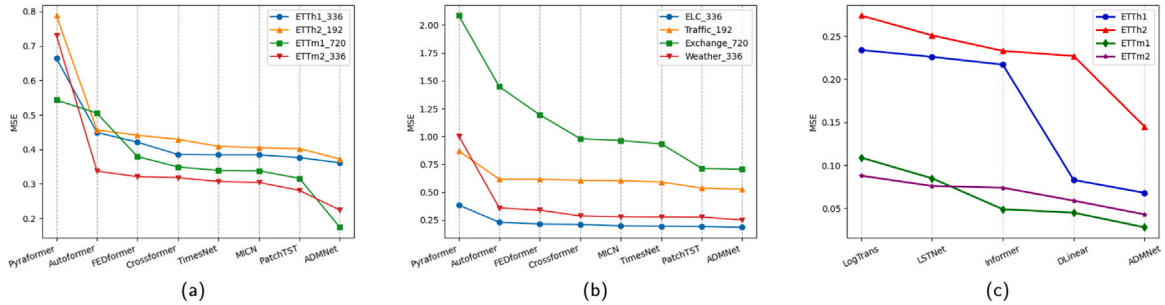


Fig. 6. Comparing the MSE of different models across eight datasets.

- Traffic Dataset: Describes road occupancy rates, comprising hourly data recorded by sensors on San Francisco highways from 2015 to 2016. Traffic flow data is typically influenced by factors such as time (*e.g.*, peak hours) and location, making it a suitable dataset for evaluating a model's ability to capture these time-varying characteristics.
- Weather: Includes 21 meteorological indicators such as temperature and humidity. Data was collected every 10 min throughout the year 2020 in Germany. Weather data exhibits clear seasonality, and the model needs to identify these periodic variations to provide accurate weather forecasts.
- ILI: The U.S. Centers for Disease Control and Prevention (CDC) reported the weekly proportion of flu-like illness cases to total patients from 2002 to 2021. Influenza incidence rates exhibit clear seasonality and periodicity, making them ideal for evaluating a model's performance in handling periodic time series data.

4.2. Evaluation metrics

Our model employs commonly used evaluation metrics for time series forecasting tasks: Mean Squared Error (MSE) and Mean Absolute Error (MAE) serve as indicators for assessing the predictive performance of the model. Their formulas are as follows:

$$MSE = \frac{1}{O}\sum_{i=1}^{O}(\hat{\mathcal{Y}}_i - \mathcal{Y}_i)^2 \tag{13}$$

$$MAE = \frac{1}{O}\sum_{i=1}^{O}\left|\hat{\mathcal{Y}}_i - \mathcal{Y}_i\right| \tag{14}$$

where, O represents the forecasting length of the time series, $\hat{\mathcal{Y}}_i$ denotes the predicted values by ADMNet, and $\mathcal{Y}_i$ represents the actual values.

4.3. The baseline models for comparison

We selected eleven state-of-the-art models, including Transformer-based models: PatchTST, Crossformer (Zhang & Yan, 2022), FEDformer (Zhou et al., 2022), Autoformer, Pyraformer (Liu, Yu, et al., 2021), Informer, LogTrans (Li et al., 2019). Non-Transformer-based models: DLinear (Zeng et al., 2023), MICN (Wang et al., 2022), TimesNet (Wu et al., 2022), LSTNet (Lai et al., 2018).

4.4. Hyper-parameter tuning

We utilized Fast Fourier Transform (FFT) to analyze the frequency distribution of various time series datasets to ensure that the kernel size could adapt to the periodic characteristics of different datasets. The hyperparameter tuning process of the ADMNet model employed a grid search strategy to ensure that the selected parameters could adapt to different datasets, allowing the model to achieve optimal performance in multi-period time series tasks. During the model training process, the sliding window for the ILI dataset was set to 36, while a sliding window size of 96 was used for the other datasets. The number of encoder layers was set to 2, the number of decoder layers was set to 1, and the hidden layer dimension of the model was set to 512. The final optimal parameters were selected through multiple experiments and iterations. Experiments were conducted on various key parameter combinations. The following is a detailed explanation of the process for selecting key parameters:

- isometric-kernel: Used to capture features at regular intervals within the signal. The purpose of this convolutional kernel is to extract periodic patterns and characteristics from the signal at equal time intervals. A grid search was performed for kernel sizes {10, 17, 24, 49}. Larger kernels can capture longer periodic features, while smaller kernels are more sensitive to short-term variations.
- conv-kernel: The main function of the downsampling convolutional kernel is to extract more representative features by reducing the resolution of the input data, while also reducing the computational load. The downsampling convolutional kernel helps the model focus on important information within the signal and reduces the impact of high-frequency noise. A grid search was performed for kernel sizes {10, 17, 24}. We aimed to find the most appropriate downsampling scale by testing different kernel sizes, balancing the retention of sufficient detail while eliminating unnecessary noise.
- batch-size: We performed a grid search between {32, 64, 128}, with 64 being selected for most datasets as it strikes a balance between computational resources and model performance. Smaller batch sizes (*e.g.*, 32) allow for more precise weight updates but increase training time. Larger batch sizes (*e.g.*, 128) can accelerate training but may reduce the model's generalization performance. learning rate: The search range was {1e−4, 5e−4, 1e−3, 5e−3}. When choosing the learning rate, we prioritized both the model's convergence speed and stability.
- S-freq: The S value was determined by observing the spectral distribution of the dataset. Frequency amplitudes were sorted in descending order, and the S value was set at the turning point where there was a significant change in amplitude. Based on the key periods shown in the frequency distribution plots of each dataset, we set the grid search range to {5, 6}, as this range encompasses the significant amplitudes for most datasets.

Table 2 lists the key parameters of the model. During model training, the sliding window for the ILI dataset was set to 36, while the window size for the other datasets was 96. We used 2 convolutional neural networks and employed the Adam optimizer, with all datasets trained for a maximum of 15 epochs. The entire training process was carried out on a single NVIDIA RTX 3060 Ti 8 GB GPU.

4.5. Data preprocessing

For a detailed description of the datasets, please refer to Table 1. The splits for the datasets ETTh1, ETTh2, ETTm1, and ETTm2 follow

Table 1
Dataset information. Huang et al. (2024).

Datasets	ETT{h1,h2}	ETT{m1,m2}	Exchange	Weather	Traffic	Electricity	ILI
Variants	7	7	8	21	862	321	7
Timesteps	17,420	69,680	7,588	52,696	17,544	26,340	966
Granularity	1 h	15 min	1 day	10 min	1 h	1 h	1 week

Table 2
Hyperparameter setting.

Parameter	Value	Explanation
B	32,64	batch size of train input data
epoch	15	train epochs
isometric_kernel	10,17,24,49	isometric convolution kernel size
conv_kernel	10,17,24	downsampling and upsampling convolution kernel size
S-freq	5,6	s frequency with the highest amplitude
e_layers	2	num of encoder layers
d_layers	1	num of decoder layers
lr	0.001	optimizer learning rate

the method described in Zhou et al. (2021), while the splits for the other datasets are based on a training/validation/testing ratio of 6/2/2. The preprocessing steps for the dataset can be broadly divided into the following steps:

- Load Data: This step involves reading the dataset from a specified path. Data Splitting: The dataset is split into training, validation, and testing sets in chronological order.
- Feature Selection: Based on the feature settings, either multivariate (M/MS) or univariate (S) feature data is selected.
- Data Normalization: The training data undergoes standardization, transforming it to have zero mean and unit variance.
- Time Feature Extraction: Time features such as year, month, day, and hour are extracted from the time column and encoded as numerical features.
- Data Slicing: The data is sliced for time series modeling purposes.
- Label Processing: Corresponding label time series are generated based on timestamps to provide temporal context for sequence prediction.
- Return Data: Finally, the required feature sequences, target sequences, and time labels are returned for use by the model.

4.6. Main results

(1) Multivariate long-term time series forecasting:

To validate the superior predictive performance of our model in long-term time series forecasting tasks compared to state-of-the-art models, we selected a wide range of forecasting lengths. Tables 3 and 4 provide the experimental comparison results between ADMNet and other ten advanced forecasting models. For ETTh1 and ETTh2, we chose forecasting lengths {24, 48, 96, 168, 192, 336, 720}. For ETTm1 and ETTm2, we selected forecasting lengths {24, 48, 96, 192, 288, 336, 720}. From Table 3, it can be observed that our model achieved the best predictive performance on most forecasting lengths for the majority of datasets. Particularly noteworthy is the improvement in predictive performance on the ETTm1 dataset, where ADMNet outperformed SOTA models by 44.93% (0.316→0.174) and 33.6% (0.363→0.241) for forecasting lengths 96 and 192, respectively. As shown in Table 4, on the ILI dataset, when the forecasting length is 24, ADMNet's predictive performance improved by 12.7% (3.052→2.664) compared to SOTA models. Overall, our model achieved the best predictive results for all forecasting lengths on ETTh2, ETTm1, ILI, and Weather datasets(see Fig. 6).

However, the Exchange dataset, lacking obvious periodicity, showed a less pronounced improvement in predictive performance since the adaptive pooling window recognizer struggled to capture its periodic length(see Fig. 9). Table 1 indicates that different time series exhibit distinct periodicities. While MICN employs fixed-sized convolutional kernels to capture time series with different periodicities, ADMNet employs a more flexible and effective decomposition approach, resulting in relatively poorer predictive performance for MICN compared to ADMNet.

TimesNet employs a hierarchical time block approach, primarily extracting high-order features from temporal patterns. However, TimesNet's handling of multi-periodic features is relatively simplistic, lacking a dedicated mechanism to address complex multi-scale periods. In contrast, ADMNet's adaptive periodic feature recognizer can flexibly capture multi-periodic variations and finely model the features of various frequency bands through a multi-frequency multi-channel network, resulting in better performance on multi-period tasks. Crossformer emphasizes a cross-dimensional attention mechanism that extracts features from multivariate time series by modeling relationships between different dimensions, but it does not excel in periodic processing. While Crossformer has advantages in cross-dimensional relationship modeling, ADMNet, through its periodic decoupling and multi-frequency channel design, can better capture multi-periodic features in time series and smooth high-frequency noise using frequency domain information. FEDformer utilizes frequency domain decomposition to separate different components of time series, but its decomposition method is fixed and cannot flexibly respond to the diverse periodic characteristics in the data. ADMNet's adaptive periodic recognizer can automatically identify multi-periodic features based on different data, providing more accurate decomposition while enhancing information interaction between different periods through periodic interaction, making it more stable in complex periodic tasks. MICN employs a multi-input convolutional network to extract features from multiple data sources, but its structure is complex and lacks a dedicated mechanism for periodic feature recognition, resulting in poorer interpretability. In contrast, ADMNet captures the key periodic features in time series more succinctly through adaptive periodic recognition while maintaining strong interpretability. Its frequency interaction module can also extract relationships between different frequencies, enhancing predictive performance. Autoformer handles trends and seasonality through an adaptive time series decomposition mechanism; however, it relies on fixed trend and seasonality partitions, which limits its ability to process complex multi-periodic signals. ADMNet can adaptively identify various periodic features and flexibly adjust the kernel size to capture both long-term and short-term characteristics. Its multi-frequency channel modeling better balances different frequency components, avoiding the limitations associated with fixed decomposition methods. Pyraformer adopts a sparse attention mechanism that performs well in terms of computational efficiency when handling long sequences, but its design mainly focuses on global relationship modeling, neglecting local periodic features. Building on global modeling, ADMNet enhances the fine processing of local periods and multi-frequency components, enabling it to better handle complex multi-period variations, particularly in scenarios where short-term and long-term features are mixed.

(2) Univariate long-term time series forecasting:

From the forecasting results in Table 5, it can be observed that ADMNet exhibits more accurate predictive performance across most forecast lengths compared to models employing seasonal trend decomposition, such as MICN, FEDformer, and Autoformer. On the four

Table 3
Forecasting results for multivariate time series. The input length L for all datasets is set to 96, and the forecasting length $O \in \{96, 192, 336, 720\}$. A lower MSE or MAE indicates better predictive performance.

Models		ADMNet		PatchTST		TimesNet		Crossformer		FEDformer		MICN		Autoformer		Pyraformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.375**	**0.413**	0.385	0.408	0.384	0.402	0.384	0.428	0.376	0.419	0.421	0.431	0.449	0.459	0.664	0.612
	192	**0.414**	**0.422**	0.431	0.432	0.436	0.429	0.438	0.452	0.420	0.448	0.474	0.487	0.500	0.482	0.790	0.681
	336	**0.432**	**0.421**	0.485	0.462	0.491	0.469	0.495	0.483	0.459	0.465	0.569	0.551	0.521	0.496	0.891	0.738
	720	0.501	0.484	**0.497**	**0.483**	0.521	0.500	0.522	0.501	0.506	0.507	0.770	0.672	0.514	0.512	0.963	0.782
ETTh2	96	**0.291**	**0.341**	0.343	0.376	0.340	0.374	0.347	0.391	0.358	0.397	0.299	0.364	0.346	0.388	0.645	0.597
	192	**0.375**	**0.406**	0.405	0.417	0.402	0.414	0.419	0.427	0.429	0.439	0.441	0.454	0.456	0.452	0.788	0.683
	336	**0.426**	**0.452**	0.448	0.453	0.452	0.452	0.449	0.465	0.496	0.487	0.654	0.567	0.482	0.486	0.907	0.747
	720	**0.432**	**0.468**	0.464	0.483	0.462	0.468	0.479	0.505	0.463	0.474	0.956	0.716	0.515	0.511	0.963	0.783
ETTm1	96	**0.174**	**0.276**	0.339	0.377	0.338	0.375	0.349	0.395	0.379	0.419	0.316	0.362	0.505	0.475	0.543	0.510
	192	**0.241**	**0.312**	0.376	0.392	0.374	0.387	0.405	0.411	0.426	0.441	0.363	0.390	0.553	0.496	0.557	0.537
	336	**0.356**	**0.402**	0.408	0.417	0.410	0.411	0.432	0.431	0.445	0.459	0.408	0.426	0.621	0.537	0.754	0.655
	720	**0.453**	**0.423**	0.499	0.461	0.478	0.450	0.487	0.463	0.543	0.490	0.481	0.476	0.671	0.561	0.908	0.724
ETTm2	96	**0.178**	**0.258**	0.192	0.273	0.187	0.267	0.208	0.292	0.203	0.287	0.179	0.275	0.255	0.339	0.435	0.507
	192	**0.224**	**0.312**	0.252	0.314	0.249	0.309	0.263	0.332	0.269	0.328	0.307	0.376	0.281	0.340	0.730	0.673
	336	**0.298**	0.405	0.318	0.357	0.321	**0.351**	0.337	0.369	0.325	0.366	0.325	0.388	0.339	0.372	1.201	0.845
	720	**0.407**	0.412	0.413	0.416	0.408	**0.403**	0.429	0.430	0.421	0.415	0.502	0.490	0.433	0.432	3.625	1.451
ELC	96	**0.158**	**0.265**	0.159	0.268	0.168	0.272	0.185	0.288	0.193	0.308	0.164	0.269	0.201	0.317	0.376	0.445
	192	0.178	0.285	**0.177**	**0.278**	0.184	0.289	0.201	0.295	0.201	0.315	0.177	0.285	0.222	0.334	0.378	0.447
	336	**0.185**	**0.296**	0.195	0.296	0.198	0.301	0.211	0.312	0.214	0.329	0.193	0.304	0.231	0.338	0.384	0.451
	720	0.213	0.321	0.215	**0.317**	0.221	0.322	0.223	0.335	0.246	0.355	**0.212**	0.321	0.254	0.361	0.389	0.452
Traffic	96	**0.511**	0.310	0.583	0.319	0.593	0.321	0.591	0.329	0.587	0.366	0.519	**0.309**	0.613	0.388	0.867	0.468
	192	**0.526**	**0.315**	0.591	0.331	0.617	0.336	0.607	0.345	0.604	0.373	0.537	0.315	0.616	0.382	0.869	0.467
	336	0.547	0.322	0.599	0.332	0.629	0.336	0.613	0.339	0.621	0.383	**0.534**	**0.313**	0.622	0.337	0.881	0.469
	720	**0.575**	**0.329**	0.601	0.341	0.640	0.350	0.620	0.348	0.626	0.382	0.577	0.330	0.660	0.408	0.896	0.473
Exchange	96	**0.101**	**0.233**	0.108	0.223	0.107	0.234	0.097	0.214	0.148	0.278	0.102	0.235	0.197	0.323	1.748	1.105
	192	0.178	0.320	0.197	0.316	0.226	0.344	0.190	0.310	0.271	0.380	**0.172**	**0.316**	0.300	0.369	1.874	1.151
	336	0.289	0.413	0.375	0.429	0.367	0.448	0.362	0.429	0.460	0.500	**0.281**	**0.407**	0.509	0.524	1.943	1.172
	720	**0.706**	**0.632**	0.934	0.773	0.964	0.746	0.980	0.783	1.195	0.841	0.714	0.658	1.447	0.941	2.085	1.206
Weather	96	**0.152**	**0.213**	0.171	0.230	0.172	0.220	0.191	0.251	0.217	0.296	0.161	0.229	0.266	0.336	0.622	0.556
	192	**0.209**	**0.259**	0.219	0.271	0.296	0.261	0.231	0.279	0.276	0.336	0.220	0.281	0.307	0.367	0.739	0.624
	336	**0.251**	**0.286**	0.277	0.321	0.280	0.306	0.287	0.332	0.339	0.380	0.278	0.331	0.359	0.395	1.004	0.753
	720	**0.312**	**0.342**	0.365	0.367	0.365	0.359	0.368	0.378	0.403	0.428	0.361	0.356	0.419	0.428	1.420	0.934

We utilized the original code from the paper, and to ensure experimental fairness, the input length and prediction length for all models are kept consistent.

Table 4
Forecasting results for multivariate time series. The input length L for all ETT datasets is set to 96, and the output length O for the ILI dataset is set to 36. The forecasting lengths $O \in \{24, 48, 168, 720\}$ for ETTh1, ETTh2 and $O \in \{24, 48, 96, 288\}$ for ETTm1, ETTm2.

Models		ADMNet		Dlinear		FEDformer		Autoformer		Informer		LSTNet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	24	**0.302**	**0.346**	0.309	0.351	0.312	0.391	0.384	0.425	0.577	0.549	1.293	0.901
	48	**0.315**	**0.357**	0.347	0.378	0.336	0.391	0.392	0.419	0.685	0.625	1.456	0.960
	168	**0.375**	**0.407**	0.428	0.423	0.418	0.443	0.490	0.481	0.931	0.752	1.997	1.214
	720	0.501	**0.484**	0.528	0.518	0.506	0.507	**0.498**	0.500	1.215	0.896	2.143	1.380
ETTh2	24	**0.165**	**0.239**	0.186	0.275	0.216	0.314	0.261	0.341	0.720	0.665	2.742	1.457
	48	**0.185**	**0.283**	0.227	0.305	0.267	0.345	0.312	0.373	1.457	1.001	3.567	1.687
	168	**0.380**	**0.423**	0.437	0.452	0.410	0.426	0.457	0.457	3.489	1.515	3.242	2.513
	720	**0.432**	**0.468**	0.773	0.631	0.463	0.474	0.474	0.484	3.467	1.473	4.625	3.709
ETTm1	24	**0.105**	**0.217**	0.249	0.312	0.282	0.349	0.383	0.403	0.323	0.369	1.968	1.170
	48	**0.131**	**0.237**	0.316	0.354	0.341	0.391	0.454	0.453	0.494	0.503	1.999	1.215
	96	**0.184**	**0.281**	0.345	0.373	0.378	0.418	0.481	0.463	0.678	0.614	2.762	1.542
	288	**0.363**	**0.371**	0.403	0.406	0.431	0.449	0.634	0.528	1.056	0.786	1.257	2.076
ETTm2	24	**0.101**	**0.186**	0.108	0.210	0.116	0.227	0.153	0.261	0.173	0.301	1.101	0.831
	48	**0.131**	**0.236**	0.143	0.245	0.148	0.255	0.178	0.280	0.303	0.409	2.619	1.393
	96	**0.174**	**0.268**	0.186	0.281	0.203	0.287	0.255	0.339	0.365	0.453	3.142	1.365
	288	0.308	0.367	**0.305**	**0.356**	0.301	0.352	0.342	0.378	1.047	0.804	2.856	1.329
ILI	24	**2.664**	**1.109**	3.052	1.244	3.228	1.260	3.483	1.287	5.764	1.677	6.235	1.951
	36	**2.582**	**1.032**	2.804	1.152	2.679	1.080	3.103	1.148	4.775	1.467	5.332	1.986
	48	**2.519**	**1.044**	2.829	1.161	2.622	1.078	2.669	1.085	4.763	1.469	6.352	2.041
	60	**2.761**	**1.103**	2.973	1.192	2.857	1.157	2.770	1.125	5.264	1.564	6.899	2.362

ETT datasets, ADMNet shows an average performance improvement of approximately 12% compared to the current state-of-the-art models. Analyzing the modeling approaches of each model, ADMNet introduces a high-frequency random term, retaining more detailed information about the time series data. Moreover, it utilizes a multi-frequency, multi-channel network for targeted feature extraction from various frequency components.

DLinear is a time series forecasting method based on linear models that primarily focuses on extracting simple linear trends. However, it has significant limitations when dealing with complex nonlinear

Table 5
Univariate long-term time series forecasting results on ETT full benchmark.

Models		ADMNet		MICN		FEDformer-f		FEDformer-w		Autoformer		Informer		LogTrans	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.058**	**0.174**	0.058	0.186	0.079	0.215	0.080	0.214	0.071	0.206	0.193	0.377	0.283	0.468
	192	**0.068**	**0.201**	0.079	0.210	0.104	0.245	0.105	0.256	0.114	0.262	0.217	0.395	0.234	0.409
	336	**0.075**	**0.213**	0.092	0.237	0.119	0.270	0.120	0.269	0.107	0.258	0.202	0.381	0.386	0.546
	720	**0.125**	0.304	0.138	0.298	0.142	0.299	0.127	**0.280**	0.126	0.283	0.183	0.355	0.475	0.629
ETTh2	96	0.129	0.274	0.155	0.300	**0.128**	**0.271**	0.156	0.306	0.153	0.306	0.213	0.373	0.217	0.379
	192	**0.145**	**0.299**	0.169	0.316	0.185	0.330	0.238	0.380	0.204	0.351	0.227	0.387	0.281	0.429
	336	**0.210**	**0.358**	0.238	0.384	0.231	0.378	0.271	0.412	0.246	0.389	0.242	0.401	0.293	0.437
	720	**0.263**	**0.415**	0.447	0.561	0.278	0.420	0.288	0.438	0.268	0.409	0.291	0.439	0.218	0.387
ETTm1	96	**0.028**	**0.128**	0.033	0.134	0.033	0.140	0.036	0.149	0.056	0.183	0.109	0.277	0.049	0.171
	192	**0.045**	**0.159**	0.048	0.164	0.058	0.186	0.069	0.206	0.081	0.216	0.151	0.310	0.157	0.317
	336	**0.052**	**0.181**	0.079	0.210	0.084	0.231	0.071	0.209	0.076	0.218	0.427	0.591	0.289	0.459
	720	**0.063**	**0.206**	0.096	0.233	0.102	0.250	0.105	0.248	0.110	0.267	0.438	0.586	0.430	0.579
ETTm2	96	**0.043**	**0.168**	0.059	0.176	0.067	0.198	0.063	0.189	0.065	0.189	0.088	0.225	0.075	0.208
	192	0.101	0.243	**0.100**	**0.234**	0.102	0.245	0.110	0.252	0.118	0.256	0.132	0.283	0.129	0.275
	336	**0.125**	**0.268**	0.153	0.301	0.130	0.279	0.147	0.301	0.154	0.305	0.180	0.336	0.154	0.302
	720	0.186	**0.310**	0.210	0.354	**0.178**	0.325	0.219	0.368	0.182	0.335	0.300	0.435	0.160	0.321

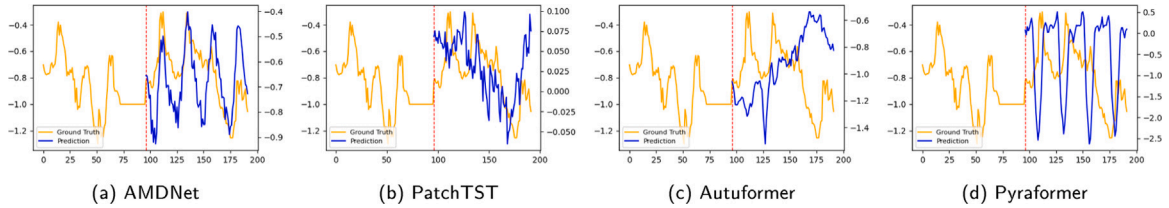


Fig. 7. Comparative performance visualization on the ETTh2 dataset with $L-96-O-192$.

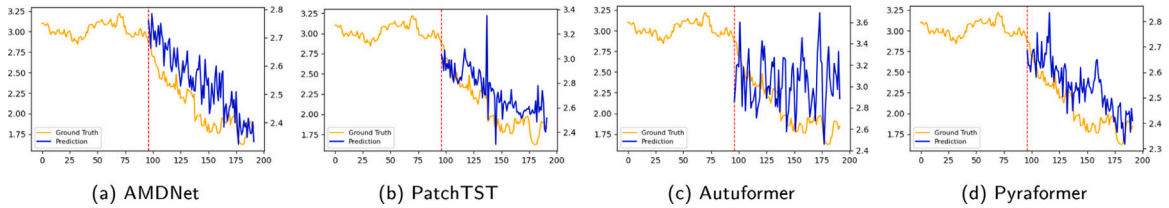


Fig. 8. Comparative performance visualization on the Weather dataset with $L-96-O-192$.

and multi-periodic features. In contrast, ADMNet can not only capture linear trends but also handle complex nonlinear multi-periodic signals through its multi-frequency channels and nonlinear periodic feature recognition, offering greater adaptability. Informer enhances the efficiency of processing long sequences through a sparse attention mechanism, but its attention mechanism is relatively weak in dealing with periodicity and high-frequency noise. ADMNet, with its adaptive periodic feature recognition and multi-frequency decomposition mechanisms, is more advantageous than Informer when handling multi-periodic signals and suppressing high-frequency noise. LSTNet is a classic model that combines convolutional and recurrent neural networks, excelling at capturing local and long-range dependencies in time series. However, it falls short in handling multi-periodicity. ADMNet, through its periodic decoupling, frequency interaction, and convolution module designs, is able to capture multi-periodic features and high-frequency noise more flexibly and accurately.

4.7. Analysis of time and space complexity

To assess the computational efficiency and cost of our model, we designed experiments evaluating time and space complexity through parameter count, training time, and MSE. To ensure fairness, all models had an input length of 96, output length of 192, batch size of 64, and epochs set to 100, with other parameters set to the default values provided in the paper.

Table 6
Analysis of complexity in different models.

Models	Complexity analysis		
	Params	Training (s)	MSE
N-BEATS	1,643,264	4,580	0.127
DeepAR	251,904	20,160	0.445
Informer	6,909,536	59,266	0.202
MICN	288,071	5,701	0.105
FEDformer	12,242,460	63,449	0.119
ADMNet	3,282,647	6,673	**0.075**

The results, as shown in Table 6. Due to the design of the adaptive periodic feature recognizer and the multi-frequency multi-channel network, our model has a higher number of parameters, which increases the time complexity. In terms of prediction accuracy, our model shows significant improvement over DeepAR with an 83.14% reduction in MSE, and an 18.47% reduction compared to MICN. Training time analysis reveals that the time complexity of our model is only exceeded by N-BEATS and MICN, with a 40.9% reduction in MSE compared to N-BEATS.

4.8. Visual analysis of model forecastings

To provide a clearer representation of the performance gap between ADMNet and other models, we overlay the real value curves with the

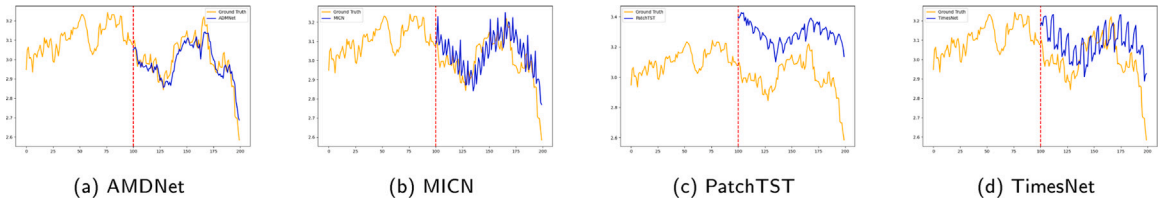


Fig. 9. Comparative performance visualization on the Exchange dataset with $L-192-O-96$.

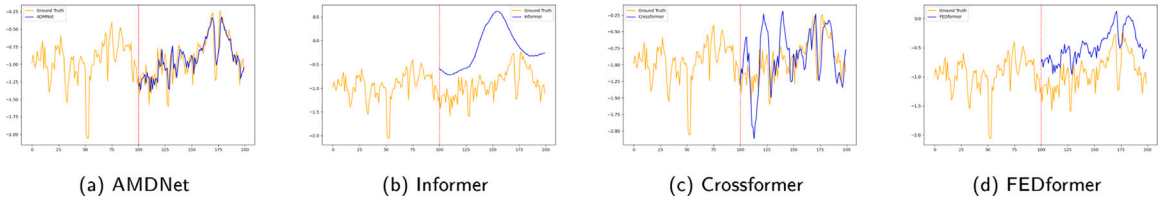


Fig. 10. Comparative performance visualization on the ETTm2 dataset with $L-192-O-96$.

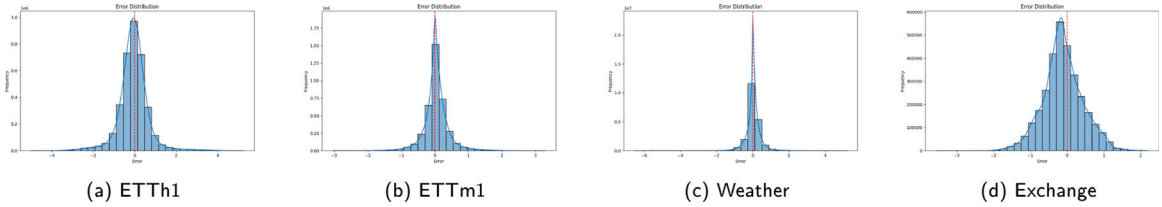


Fig. 11. Prediction error distribution plots of the model on ETTh1, ETTm1, Weather, and Exchange.

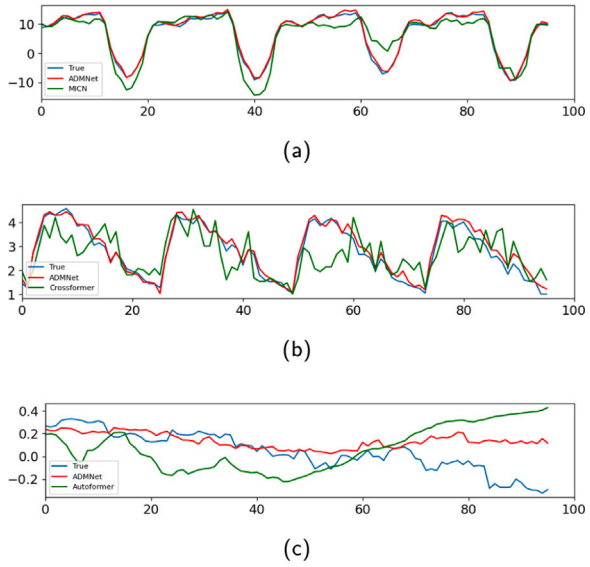


Fig. 12. Comparative visualization of forecasting performance between ADMNet and three benchmark models.

predicted curves on the same canvas. The fitting degree of the two curves is observed to assess the predictive performance of the models. As shown in Figs. 7 and 8, we conduct comparative experiments using the ETT and Weather datasets. Fig. 12 indicates that for data with pronounced periodicity, our model exhibits better fitting, indicating that ADMNet is more adept at improving predictive performance by capturing the periodicity of the data compared to other models.

To make our model more convincing, we added a comparison chart of the predicted and actual values between ADMNet and the baseline models on the ETTm2 dataset (Fig. 10), which ensures that our comparison now covers all multivariate baseline models. Additionally, we included error distribution plots for the model on four datasets, as shown in Fig. 11. The higher the frequency of data points close to the red dashed line, the more accurate the predictions made by our model, which further illustrates ADMNet's advantage in reducing prediction errors. Regarding the outliers, we believe that although ADMNet is designed to handle multi-periodicity, when there are multiple overlapping periods in the data, the model may not effectively identify the true periodic characteristics, which affects the prediction results. To address this limitation, we plan to incorporate a frequency interaction module in future work to model the relationships between different frequencies, thereby enhancing the model's ability to understand overlapping periods and handle more complex data (see Tables 7 and 8).

4.9. Ablation study

To validate the effectiveness of each component in improving the model's performance, we conducted experiments on the ELC dataset using MSE and MAE as evaluation metrics.

As shown in Table 9, we conducted three experiments to verify the effectiveness of the enhanced adaptive periodic feature recognizer and the multi-channel network architecture. Firstly, to validate the effectiveness of the enhanced adaptive periodic feature recognizer, we replaced it with average pooling using fixed-sized $kernel = i \in \left\{\frac{L}{4}, \frac{L}{8}, \ldots\right\}$ to pool the original time series. The forecasting was performed using the same multi-frequency multi-channel network as ADMNet. Secondly, to verify the effectiveness of the multi-channel network architecture, we retained the proposed enhanced adaptive periodic feature recognizer, removed the high-frequency random term, and adopted a dual-frequency dual-channel network architecture similar to other models. In Table 9, C-APR represents using fixed-sized convolutional kernels instead of the enhanced adaptive periodic feature recognizer while retaining the multi-frequency multi-channel network architecture. Two-channel indicates modeling only the mid-low-frequency cycle feature fusion item.

Table 7
Performance comparison on the Traffic dataset for different input lengths and forecasting lengths.

Input length		192				336				720			
Prediction length		96	192	336	720	96	192	336	720	96	192	336	720
Ours	MSE	**0.425**	**0.436**	**0.452**	**0.468**	**0.376**	**0.411**	**0.426**	**0.432**	**0.324**	**0.349**	**0.371**	**0.422**
	MAE	**0.293**	**0.302**	**0.301**	**0.302**	**0.261**	**0.276**	**0.284**	**0.299**	**0.246**	**0.252**	**0.265**	**0.284**
Dlinear	MSE	0.451	0.460	0.475	0.504	0.410	0.423	0.436	0.466	0.387	0.398	0.413	0.451
	MAE	0.303	0.308	0.316	0.335	0.282	0.287	0.296	0.315	0.274	0.279	0.287	0.309
FEDformer	MSE	0.589	0.611	0.617	0.624	0.629	0.637	0.633	0.641	0.641	0.608	0.609	0.663
	MAE	0.367	0.379	0.380	0.377	0.406	0.407	0.397	0.391	0.412	0.376	0.378	0.411
Autoformer	MSE	0.605	0.596	0.636	0.685	0.602	0.600	0.676	0.698	0.583	0.588	0.624	0.684
	MAE	0.377	0.368	0.391	0.426	0.379	0.368	0.418	0.439	0.381	0.392	0.411	0.430

Table 8
Performance comparison on the Weather dataset for different input lengths and forecasting lengths.

Input length		192				336				720			
Prediction length		96	192	336	720	96	192	336	720	96	192	336	720
Ours	MSE	**0.154**	**0.180**	**0.255**	**0.281**	**0.158**	**0.204**	**0.256**	**0.295**	**0.153**	**0.201**	**0.244**	**0.304**
	MAE	**0.198**	**0.264**	**0.314**	**0.349**	**0.201**	**0.248**	**0.281**	**0.332**	**0.209**	**0.256**	**0.286**	**0.318**
Dlinear	MSE	0.185	0.225	0.271	0.335	0.176	0.220	0.265	0.323	0.168	0.210	0.257	0.315
	MAE	0.245	0.282	0.321	0.373	0.237	0.282	0.319	0.362	0.226	0.265	0.307	0.353
FEDformer	MSE	0.232	0.278	0.348	0.388	0.256	0.277	0.389	0.408	0.292	0.311	0.371	0.417
	MAE	0.306	0.333	0.385	0.408	0.326	0.334	0.426	0.432	0.344	0.358	0.407	0.431
Autoformer	MSE	0.253	0.398	0.361	0.420	0.274	0.370	0.409	0.407	0.346	0.413	0.443	0.440
	MAE	0.332	0.455	0.398	0.433	0.353	0.430	0.448	0.421	0.407	0.437	0.466	0.454

Table 9
Ablation experiments on the Electricity dataset.

Prediction length	96		192		336		720	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ADMNet	**0.101**	**0.233**	**0.178**	**0.320**	**0.289**	**0.413**	**0.706**	**0.632**
C-APR	0.139	0.277	0.218	0.358	0.311	0.468	0.729	0.864
Two-channel	0.148	0.296	0.242	0.429	0.361	0.471	0.755	0.939

Table 10
Stock symbols in DJIA.

DJIA	AAPL	AMGN	AXP	BA	CAT	CRM
	CSCO	CVX	DIS	GS	HD	HON
	IBM	INTC	JNJ	JPM	KO	MCD
	MMM	MRK	MSFT	NKE	PG	TRV
	UNH	V	VZ	WBA	WMT	

Table 11
Prediction performance of four methods on DJIA.

Model	MSE	MAE
N-BEATS	10.96	2.01
CNN-LSTM	10.74	1.98
VML	7.05	1.55
ADMNet	**6.13**	**1.50**

4.10. Applications of research findings

Time series forecasting tasks have wide-ranging practical applications and significant impacts in areas such as energy load forecasting, financial market forecasting (*e.g.*, exchange rate prediction), weather, and traffic. To evaluate the applicability of our model in real-world scenarios, we selected the DJIA dataset, consisting of 29 U.S. company stocks, as an example for stock market prediction. The data was sourced from Yahoo Finance. The dataset information is shown in Table 10, covering a time span from January 2, 2018, to December 31, 2020, including 756 trading days. We split the dataset into training, validation, and test sets with a ratio of 4:1:1.

To verify the performance of ADMNet in stock prediction, we conducted comparative experiments with N-BEATS (Oreshkin et al., 2019), CNN-LSTM (Lu et al., 2020), and VML (Liu, Ma, et al., 2022). N-BEATS is a deep learning-based time series forecasting model that can automatically extract trend and periodic features from data without manually decomposing the time series. It is well-suited for complex, non-linear stock market data and has performed exceptionally well across multiple benchmark datasets. CNN-LSTM is a hybrid model combining Convolutional Neural Networks (CNN) and Long Short-Term Memory Networks (LSTM). CNN extracts local features, while LSTM captures long-term dependencies, making it suitable for handling both short-term fluctuations and long-term trends in stock prices. VML focuses on volatility prediction in stock markets, utilizing machine learning techniques (such as neural networks and SVM) to analyze volatility patterns, helping investors manage risk and make investment decisions. As shown in Table 11, our model outperformed all three models across all metrics, validating that ADMNet's multi-period recognition capability enhances the accuracy of financial market predictions, enabling investors to better forecast long-term trends and short-term fluctuations.

5. Conclusion

This paper proposes an Adaptive Down-sampling Multi-frequency Multi-channel Network (ADMNet) for effective utilization of the multiple periodicities present in time series data. The model introduces a module that adaptively captures the periodic lengths in time series, addressing the rigid and impractical selection of pooling window sizes in traditional time series decomposition methods. Additionally, considering that the future values of a time series are influenced by multiple periods, we innovatively introduce two different frequency bands of cyclic feature fusion components and a high-frequency random component containing more detailed features. These three components are modeled separately for forecasting. Experimental results demonstrate that ADMNet outperforms state-of-the-art methods on multiple datasets. Considering the complexity of time series data, ADMNet still has limitations in capturing overlapping periods. We plan to explore more complex frequency fusion strategies, such as introducing frequency interaction mechanisms to better capture overlapping periodic features, and combine it with other adaptive learning techniques to further improve the model's prediction accuracy and efficiency. Future research directions include: introducing more complex adaptive feature

extraction modules to enable the model to automatically adapt to different data characteristics; optimizing the model structure to improve computational efficiency and adapt to large-scale or high-dimensional datasets; exploring the application of ADMNet in other fields, such as biosignal analysis and demand forecasting; and combining ADMNet with other time series forecasting models to enhance the model's performance.

CRediT authorship contribution statement

Lele Yuan: Writing – original draft, Software, Formal analysis, Data curation, Visualization. **Hua Wang:** Validation, Investigation, Resources, Funding acquisition. **Fan Zhang:** Conceptualization, Methodology, Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the following: the National Natural Science Foundation of China under Grant No. 62272281, the Special Funds for Taishan Scholars Project, China (tsqn202306274), the Youth Innovation Technology Project of Higher School in Shandong Province, China under Grant No. 2023KJ212, and the NSFC Joint Fund with Zhejiang Integration of Informatization and Industrialization under Key Project, China (U22A2033).

Data availability

Data will be made available on request.

References

Abbasimehr, H., Behboodi, A., & Bahrini, A. (2024). A novel hybrid model to forecast seasonal and chaotic time series. *Expert Systems with Applications*, *239*, Article 122461.

Bi, X., Zhang, C., He, Y., Zhao, X., Sun, Y., & Ma, Y. (2021). Explainable time–frequency convolutional neural network for microseismic waveform classification. *Information Sciences*, *546*, 883–896.

Bi, X., Zhang, C., Wang, F., Liu, Z., Zhao, X., Yuan, Y., & Wang, G. (2021). An uncertainty-based neural network for explainable trajectory segmentation. *ACM Transactions on Intelligent Systems and Technology*, *13*(1), 1–18.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189–215.

Chen, M., Peng, H., Fu, J., & Ling, H. (2021). Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 12270–12280).

Chen, G., Wang, H., Liu, Y., Zhang, M., & Zhang, F. (2023). Resformer: Combine quadratic linear transformation with efficient sparse transformer for long-term series forecasting. *Intelligent Data Analysis*, *27*(6), 1557–1572.

Geng, X., He, X., Hu, M., Bi, M., Teng, X., & Wu, C. (2024). Multi-attention network with redundant information filtering for multi-horizon forecasting in multivariate time series. *Expert Systems with Applications*, *257*, Article 125062.

Huang, S., & Liu, Y. (2024). FL-Net: A multi-scale cross-decomposition network with frequency external attention for long-term time series forecasting. *Knowledge-Based Systems*, *288*, Article 111473.

Huang, S., Liu, Y., Zhang, F., Li, Y., Li, J., & Zhang, C. (2024). CrossWaveNet: A dual-channel network with deep cross-decomposition for long-term time series forecasting. *Expert Systems with Applications*, *238*, Article 121642.

Khaled, A., Elsir, A. M. T., & Shen, Y. (2022). TFGAN: Traffic forecasting using generative adversarial network with multi-graph convolutional network. *Knowledge-Based Systems*, *249*, Article 108990.

Lai, G., Chang, W. C., Yang, Y., & Liu, H. (2018). Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval* (pp. 95–104).

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y. X., & Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, *32*.

Liu, X., Guo, J., Wang, H., & Zhang, F. (2022). Prediction of stock market index based on ISSA-BP neural network. *Expert Systems with Applications*, *204*, Article 117604.

Liu, T., Ma, X., Li, S., Li, X., & Zhang, C. (2022). A stock price prediction method based on meta-learning and variational mode decomposition. *Knowledge-Based Systems*, *252*, Article 109324.

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., & Dustdar, S. (2021). Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*.

Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., & Xu, Q. (2022). Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, *35*, 5816–5828.

Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural computing and applications*, *32*, 17351–17360.

Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. *Complexity*, *2020*(1), Article 6622927.

Nagy, B., Hegedűs, I., Sándor, N., Egedi, B., Mehmood, H., Saravanan, K., Lóki, G., & Kiss, Á. (2023). Privacy-preserving federated learning and its application to natural language processing. *Knowledge-Based Systems*, *268*, Article 110475.

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2022). A time series is worth 64 words: Long-term forecasting with transformers. arXiv preprint arXiv:2211.14730.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437.

Quan, Q., Hao, Z., Xifeng, H., & Jingchun, L. (2022). Research on water temperature prediction based on improved support vector regression. *Neural Computing and Applications*, 1–10.

Satrio, C. B. A., Darmawan, W., Nadia, B. U., & Hanafiah, N. (2021). Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET. *Procedia Computer Science*, *179*, 524–532.

Sharma, R. R., Kumar, M., Maheshwari, S., & Ray, K. P. (2020). EVDHM-arima-based time series forecasting model and its application for COVID-19 cases. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–10.

Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P., & Homayouni, S. (2020). Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 6308–6325.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wang, H., Peng, J., Huang, F., Wang, J., Chen, J., & Xiao, Y. (2022). Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*.

Wang, M., Wang, H., & Zhang, F. (2023). FAMC-Net: Frequency domain parity correction attention and multi-scale dilated convolution for time series forecasting. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (pp. 2554–2563).

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., & Long, M. (2022). Timesnet: Temporal 2d-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186.

Wu, Y., Meng, X., Zhang, J., He, Y., Romo, J. A., Dong, Y., & Lu, D. (2024). Effective LSTMs with seasonal-trend decomposition and adaptive learning and niching-based backtracking search algorithm for time series forecasting. *Expert Systems with Applications*, *236*, Article 121202.

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting? *37*, In *Proceedings of the AAAI conference on artificial intelligence* (9), (pp. 11121–11128).

Zhang, F., Chen, G., Wang, H., Li, J., & Zhang, C. (2023). Multi-scale video super-resolution transformer with polynomial approximation. *IEEE Transactions on Circuits and Systems for Video Technology*.

Zhang, F., Chen, G., Wang, H., & Zhang, C. (2024). CF-DAN: Facial-expression recognition based on cross-fusion dual-attention network. *Computational Visual Media*, 1–16.

Zhang, F., Guo, T., & Wang, H. (2023). DFNet: Decomposition fusion model for long sequence time-series forecasting. *Knowledge-Based Systems*, *277*, Article 110794.

Zhang, W., Wang, H., & Zhang, F. (2024). Skip-timeformer: Skip-time interaction transformer for long sequence time-series forecasting. In *International joint conference on artificial intelligence* (pp. 5499–5507).

Zhang, Y., & Yan, J. (2022). Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.

Zhang, J., Zheng, Y., & Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. *31*, In *Proceedings of the AAAI conference on artificial intelligence*. (1).

Zhong, S., Scarinci, A., & Cicirello, A. (2023). Natural language processing for systems engineering: Automatic generation of systems modelling language diagrams. *Knowledge-Based Systems*, *259*, Article 110071.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning* (pp. 27268–27286). PMLR.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. *35*, In *Proceedings of the AAAI conference on artificial intelligence* (12), (pp. 11106–11115).