Energy 314 (2025) 133964

Contents lists available at ScienceDirect

Energy

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

High-precision energy consumption forecasting for large office building using a signal decomposition-based deep learning approach

Chao-fan Wang [a,1], Kui-xing Liu [b,1], Jieyang Peng [c], Xiang Li [d], Xiu-feng Liu [e], Jia-wan Zhang [a], Zhi-bin Niu [a,*]

[a] College of Intelligence and Computing, Tianjin University, China
[b] School of Architecture, Tianjin University, China
[c] Department of Electronic Engineering Tsinghua University, Beijing, China
[d] China Iron and Steel Research Institute Group, Beijing, China
[e] Technical University of Denmark, Denmark

ARTICLE INFO

Keywords:
Energy consumption forecasting
Building energy efficiency
Energy data analytics
Deep learning

ABSTRACT

Accurate long-term energy consumption forecasting is crucial for efficient energy management in large office buildings. Recent research highlights that deep learning approaches, including RNN, LSTM, and transformer-based models, are at the forefront of promising advancements. They are unified in obtaining more discriminative representations. The challenges lie in the complexity of data influenced by diverse factors such as weather, building characteristics, and occupant behavior, etc., and the need to accurately model the intricate patterns of time-series periodicity and trends. In this paper, we introduce `SPAformer`, an innovative end-to-end deep learning model adept at unraveling and forecasting the intricate components of energy consumption data. It is motivated by the hypothesis that decomposing energy consumption into detailed functional categories and isolating trends and periodic components can significantly enhance forecasting accuracy. In response, we propose spectra-patch attention (SPA) mechanism, which combines time and frequency signals, to better capture the repeating patterns in lengthy data sequences. We have evaluated our approach on a real-world granular dataset from a large commercial office building and demonstrated SPAformer's superior performance. By achieving a 12% improvement in prediction accuracy over state of the art attention-based models, SPAformer marks a significant stride in energy forecasting. This work contributes to better-informed decision making about energy saving strategies, emphasizing the model's usefulness in the ongoing planning and fine-tuning of building energy systems.

1. Introduction

The imperative for sustainable urban development has placed a spotlight on the energy consumption of large office buildings, which are significant contributors to urban energy demand [1]. In China, for instance, nearly 28% of the nation's total energy consumption is attributed to buildings, a figure that is on an upward trajectory due to the swift completion of new buildings and the ongoing improvement in living standards [2]. This situation underscores the critical need for China, as well as other nations, to prioritize the reduction of energy consumption in buildings and enhance energy efficiency measures [3]. Accurately forecasting long-term energy use within these structures is crucial for effective energy management and conservation. However, the complexity of energy patterns and the influence of various unpredictable factors make long-term forecasting a formidable challenge.

Previous research in energy consumption forecasting has primarily focused on short-term [4] and medium-term predictions [5], leveraging a range of methodologies from traditional statistical models to more recent machine learning techniques [6,7]. Statistics-based models can effectively model the short-term patterns of sequences and achieve the accuracy of short-term forecasts. Limited by the reasonable selection of autoregressive models and the high requirement for domain knowledge, statistical modeling is not suitable for mining medium and long-term patterns [8]. Deep learning methods, including RNN, LSTM and attention mechanism-based models, have been successively proposed to

* Corresponding author.
E-mail addresses: wcf@tju.edu.cn (C.-f. Wang), liukuixing1@sina.com (K.-x. Liu), pengjieyang1991@gmail.com (J. Peng), xli@berkeley.edu (X. Li), xiuli@dtu.dk (X.-f. Liu), jwzhang@tju.edu.cn (J.-w. Zhang), zniu@tju.edu.cn (Z.-b. Niu).
[1] Contributed equally.

https://doi.org/10.1016/j.energy.2024.133964
Received 27 April 2024; Received in revised form 14 October 2024; Accepted 23 November 2024
Available online 24 December 2024

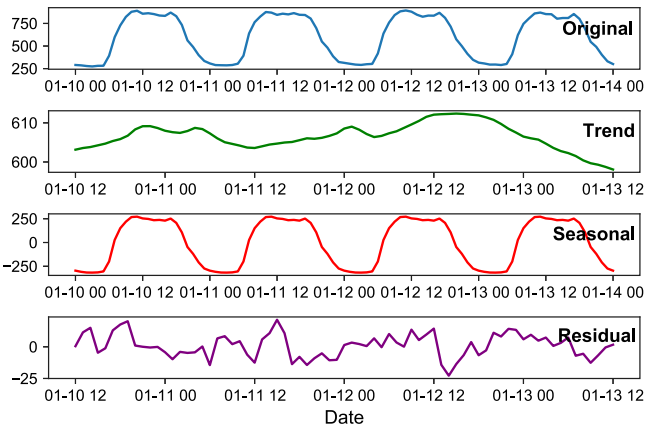

Fig. 1. Periodic-trend series decomposition.

solve the sequence prediction problem, which brings hope to the prediction of medium and long-term energy consumption [9]. In particular, the attention mechanism, with its natural advantage in processing sequence data, has been widely used in various time series prediction tasks [10], including energy consumption prediction [11]. It is important to note that compared to universal time series, building energy consumption data are affected by a variety of factors such as weather, building characteristics, and occupant behavior. This makes the patterns contained in building energy consumption data more complex. **Previous models face the challenge of insufficient representation capabilities when dealing with the prediction of building energy consumption, and are often difficult to cope with the dual challenges of trend decomposition and periodicity detection of long sequence energy consumption data.**

The primary challenge in long-term energy consumption forecasting lies in **the intricate interplay between trend components and periodic fluctuations**, which are difficult to disentangle and predict over extended period. As shown in Fig. 1, energy data is intertwined by trends, cycles, and noise. Modeling based on observations captures spurious correlations of unpredictable noise. This spurious correlation is exacerbated when different representations learned by the model become entangled [12]. In addition, most existing models lack the granularity to accurately reflect the diverse categories of energy usage and their unique patterns. It cannot accurately identify the direction of building energy consumption [13]. Furthermore, the application of attention mechanisms has been limited in their **ability to simultaneously process and integrate time-domain and frequency-domain information**, which is essential for capturing the multifaceted nature of energy consumption dynamics. Transformer variants mostly capture the long-term dependence of energy consumption through time domain information [14]. Although FEDformer [15] fuses the frequency signal into the transformer model through the frequency domain enhancement block, the lack of time domain information reduces the representation ability of short periods and local trends. This gap in the research landscape underscores the need for an innovative approach that can adeptly handle these complexities, paving the way for more accurate and reliable long-term energy forecasting methodologies.

To address the above challenges, we propose an end-to-end signal decomposition-based energy consumption prediction model for large office buildings, namely `SPAformer`.

The model decomposes the raw long-series energy consumption data into period and trend components via a sequence decomposition module. This decomposition is pivotal as it aligns with our understanding that energy consumption patterns are inherently composed of these two elements, each requiring a distinct approach for accurate forecasting. As the main component of long-term series, fluctuations in trend components are easier to learn by the model. Therefore, we use a simple Multilayer Perceptron (MLP) to predict the trend term to prevent overfitting. This choice is justified by the MLP's proven efficiency in capturing linear relationships, making it an ideal fit for trend prediction where complexity does not necessarily equate to accuracy.

For the prediction of periodic items, we consider that the dataset is not particularly large, which suggests that the risk of overfitting is acceptable. Therefore, the encoder–decoder architecture is used for prediction of periodic components. This architecture's ability to model complex dependencies between time steps makes it especially suitable for capturing the nuanced patterns within periodic data. Experimental results also show that this architecture performs better than direct prediction, providing empirical evidence for its effectiveness.

In order to capture the complex periodic patterns of long-distance energy consumption sequences, we propose the **Spectral-Patch Attention (SPA)** mechanism to fully exploit the ability of time-domain and frequency-domain signals to capture periodic patterns. This mechanism is crucial for understanding the multifaceted nature of energy consumption data, where different frequency-domain signals reflect different characteristics of time-series data. The reasoning behind focusing on low-frequency components for long-period predictions is that they are more influential in shaping the overall trend, a hypothesis supported by our preliminary analyses. Therefore, in SPA, we propose a multi-scale frequency multi-head self-attention module to capture both long and short periods. This module's design is inspired by the notion that accurately forecasting energy consumption necessitates a nuanced understanding of its temporal dynamics across various scales.

In addition, inspired by PathTST [16], we introduce the patch self-attention mechanism into the SPA module to consider time-domain signals. This inclusion is justified by the need to better capture local dependencies and non-periodic features, which are often overshadowed by broader trends but are crucial for high-resolution forecasts. The patch self-attention mechanism represents our commitment to a holistic approach, ensuring that both global and local patterns are given due consideration in our model.

In summary, the challenges of accurately predicting energy consumption in large office buildings are multifaceted, involving complex patterns of energy use that are influenced by both predictable and unpredictable variables, how to disentangle trends, cycles, and noise in energy data, and how to simultaneously process and integrate time-domain and frequency-domain information to cope with the dual challenges of trend decomposition and periodicity detection in long sequence energy consumption data. Recognizing the gap in existing methodologies for handling long-series data with both trend and periodic variations, we successively proposed the sequence decomposition module and the spectral-patch attention (SPA) mechanism. In addition, we proposed to integrate them in an encoder–decoder architecture. This study makes an important contribution to the field of energy consumption prediction for large office buildings, focusing on the novel method of SPAformer model encapsulation.

In short, we summarize the main contributions of this work as follows:

- Our technical innovation lies in the development of SPAformer, a state-of-the-art model that integrates several novel components for enhanced forecasting accuracy. At its core, SPAformer utilizes a signal decomposition-based approach, separating energy consumption data into trend and periodic components. This is complemented by the introduction of the spectral-patch attention (SPA) mechanism, which adeptly captures complex periodic patterns through a nuanced analysis of time-domain and frequency-domain signals. This dual focus on decomposition and attention mechanisms represents a significant leap forward in predictive modeling.
- We have subjected SPAformer to rigorous evaluation, utilizing a robust dataset to benchmark its performance against existing models. This evaluation not only demonstrates SPAformer's superior capability in accurately forecasting energy consumption

but also validates the effectiveness of our novel technical contributions. Through extensive testing, including comparisons to baseline models and assessments under various scenarios, our work provides a solid foundation for the practical application and future development of energy consumption forecasting models.

- Our model achieves $\mathcal{O}(L \log L)$ complexity. In addition, SPAformer achieves state-of-the-art results in both memory consumption and time efficiency on real devices. At the same time, we further evaluated the model on 6 popular benchmark datasets, verifying the strong generalization ability and training stability of SPAformer. We used multiple indicators such as MSE, MAE and KS test to evaluate the performance of the model. Through comprehensive and detailed evaluation, our work provides support for the application and development of a wide range of long-time series prediction models.

2. Related work

The domain of energy consumption forecasting, particularly for large office buildings, has witnessed a variety of methodological advancements aimed at enhancing prediction accuracy and reliability. This body of work spans from traditional statistical methods to more recent machine learning and deep learning approaches, each contributing unique insights and tools for tackling the complex dynamics of energy usage. The evolution of these methodologies reflects an increasing emphasis on handling high-dimensional data, capturing temporal dependencies, and addressing the challenges of non-linear patterns in energy consumption. Notably, the integration of attention mechanisms and the decomposition of time series into interpretable components have emerged as significant themes. These techniques aim to improve model performance by providing nuanced analysis capabilities and facilitating the understanding of underlying consumption patterns. Additionally, comparative studies have underscored the importance of thorough evaluation frameworks, emphasizing the need for models to be tested across diverse settings and conditions to validate their generalizability and effectiveness.

2.1. Time-series decomposition techniques

Time-series decomposition is an important strategy for analyzing building sub-item energy consumption. The sequence is usually decomposed into three components, including trend term, period term and residual [17]. The trend term reflects the overall trend of the time-series without considering seasonality and irregularity [18]. The period term reflects the periodicity of the time-series. It is usually associated with time intervals such as daily, weekly, monthly, yearly, etc. The residual reflects the part of the time-series that is not explained by trend and periodicity, such as noise, mutations, etc. For long-term sequence tasks, sequence decomposition is performed before predicting future sequences. Then different modeling methods are used according to the different attributes of each component.

Our survey found that in the field of building, a large number of previous studies have used sequence decomposition strategies to predict energy consumption. D.Liu et al. [19] used a joint algorithm of SVR and empirical mode decomposition (EMD) to predict building energy consumption. This strategy can effectively enhance the model's ability to capture sequence patterns while reducing the complexity of sequence prediction. C.Zhou et al. [20] proposed a new multiple decomposition integration method based on residual compensation, which decomposes energy consumption into trends and residual. This study uses a linear regression model to predict the trend sub-series, and a triple exponential smoothing model to evaluate the low-frequency approximation residual sub-series. Finally, the sum of sub-sequence predictions is used as the total energy consumption prediction result.

In recent studies, researchers have realized the important role of sequence decomposition in transformer prediction of time-series. Autoformer [21] first introduced a seasonal-trend decomposition architecture in transformer, which uses a simple moving average method. The results show that the introduction of the decomposition architecture significantly improves the model performance by 50%–80%. CoST [12] proposes to learn different feature representations through sequence decomposition, which applies contrastive learning methods to learn period and trend representations. SSDNet [22] combines the transformer architecture with state space models to provide probabilistic and interpretable forecasts, including trend and seasonality components and previous time steps important for the prediction.

However, they did not fully consider the respective characteristics of cycles and trends. These studies mixed the two predictions in the encoder–decoder, which did not isolate the periodicity and trend prediction well, resulting in the entanglement of the two. Research [23] shows that (I) For the prediction of periodic components, frequency-domain attention models are more sample efficient than time-domain attention models, as softmax with exponential terms correctly amplifies the dominant frequency pattern in Fourier space. (II) For trend data, attention models often exhibit poor generalization, as attention models naturally interpolate rather than infer context.

2.2. Predictive modeling in time-series analysis

According to the prediction range, energy consumption prediction tasks can be divided into short-term, medium-term and long-term prediction. However, long sequences usually contain more complex patterns than short sequences, and previous research mainly focused on short-term or medium-term predictions of energy consumption, such as energy consumption in the next day or week. If we can achieve accurate predictions of energy consumption for the next few months or even longer, policymakers can formulate more effective energy management and energy-saving policies.

2.2.1. Overview of long time-series prediction challenges

Commercial office building energy consumption, as a natural time-series data, contains building thermal inertia, periodicity and time lag [24]. The periodicity of energy consumption is usually a complex pattern of mixed short and long term. The short-term periodicity patterns include daily, weekly cycles, etc [25]. For example, building chillers will typically operate at night and shut down during the day due to the effects of a stepped electricity tariff [26]. Commercial office buildings, which are staffed by mostly office workers, have a significant difference in energy consumption between weekdays and weekends. Weekday energy consumption is greater than the weekend, so there is a clear weekly cycle [27]. Affected by the different climates in the four seasons in the north, there is a seasonal cycle in building energy consumption. For example, cold stations consume more electricity in the summer, while hot stations consume more in the winter [28]. Therefore, decomposing the series and predicting complex periodic patterns individually is crucial to improve office building energy consumption predictions. Therefore, improving the model's ability to capture both short and long periods of long sequences is critical to improving prediction accuracy.

The survey found that in order to capture the complex periodic patterns in the sequence, previous studies on building energy consumption prediction considered different information domains, which were divided into three categories: (I) time-domain (II) frequency-domain (III) hybrid time-frequency.

2.2.2. Comparative review of time-domain prediction models

In the time domain, we can recognize the growth or decline of trends, periodic changes, etc. Time domain information is important

to capture the dynamic characteristics of the sequence, which provides direct information about how the signal changes over time. Most of the previous studies predict building energy consumption from a time domain perspective. Rahman et al. [29] developed and optimized deep recurrent neural network (RNN) models from a time domain perspective, aiming to predict medium- and long-term (≥ 1 week) electricity load in hourly units. The paper also analyzed the relative performance of the model for different types of electricity consumption patterns and used deep NN to perform imputation on an electricity consumption dataset containing segments of missing values. Peng et al. [30] proposed a spatial–temporal feature extraction framework that integrates spatial and temporal information to capture the consistency of energy consumption data. This method achieves high-precision demand prediction at the user level by considering the correlation of energy consumption patterns among different users. Li et al. [31] used wavelet transform to filter the raw data and separate the daily periodic patterns and residuals, and then the predictions of the date patterns and residuals were combined to obtain the final prediction.

Time domain information of time series is also widely used in transformer variants. Time domain information of time series is also widely used in transformer variants. For example, PatchTST [16] provided a solution for multivariate time series forecasting in the time domain: (I) It divides the time-series into sub-sequence-level patches as the input labels of the Transformer, which better preserves local semantic information. (II) Channel independence: each channel contains a univariate time-series, and all channels share the same embedding and weights to predict multivariate time-series data.

2.2.3. Assessment of frequency-domain prediction approaches

In the frequency domain, we decompose the different frequency components of the signal by converting the time-domain signal into a frequency-domain representation via Fourier transform or wavelet transform. This helps the model to detect hidden periodic patterns in the series, such as daily, weekly and monthly periodicity in building energy consumption data. Yan et al. [32] proposed an ultra-short-term photovoltaic power prediction model based on optimal frequency domain decomposition and deep learning. They decomposed photovoltaic power into low-frequency and high-frequency components, and utilized convolutional neural network (CNN) to predict them, and then obtained the final prediction results through additive reconstruction.

Recently, some researchers have found that combining the frequency domain with the attention mechanism can further improve the accuracy of long-term sequence prediction. FEDformer [15] proposed a Fourier frequency domain enhancement block to capture important structures in time series through frequency domain mapping. ETSformer [33] selects top-K largest amplitude modes as frequency domain attention, and combines it with exponential smoothing attention to replace the transformer's self-attention mechanism.

2.2.4. Hybrid time-frequency prediction methods

Although time-domain analysis is able to capture the dynamic characteristics of the sequence, it cannot fully capture the frequency characteristics for some highly periodic time series because it ignores the information in the frequency domain. On the contrary, focusing only on the frequency domain information will ignore the dynamic changes in the time domain. Therefore, the joint time-frequency approach is proposed to fill the gap between the two. Zhang et al. [34] proposed to decompose the electricity load into several components. Then a hybrid model based on improved empirical mode decomposition (IEMD), autoregressive integrated moving average (ARIMA) and wavelet neural network (WNN) was designed. The model was then optimized by the fruit fly optimization algorithm (FOA) to improve the prediction accuracy. Mounir et al. [35] proposed a power forecasting method based on EMD and bidirectional LSTM, in which EMD separates the time series into components of different resolutions, and LSTM predicts each component separately.

However, models based on attention mechanisms are mostly studied from a single perspective of time domain or frequency domain. This cannot fully capture the multi-scale features of long time series. In addition, the frequency-domain attention mechanism ignores the time-domain information, which is important for capturing the local and dynamic features of the sequence. Therefore, we propose a time-frequency joint method and consider the impact of multi-scale frequency domain information on long-term series prediction in the frequency domain.

2.3. Advances in multivariate energy prediction

Building sub-item energy consumption data, as a multivariate time-series, is predicted by many methods that can be categorized into three groups: (I) physical modeling (II) statistical and shallow learning methods (III) deep learning methods. Our research focuses on deep learning methods.

2.3.1. Statistical and shallow learning methods based multivariate modeling for energy prediction

In the past decades, there have been a large number of studies using statistical methods to predict building energy consumption, such as autoregressive integrated moving average (ARIMA) [36] and support vector regression (SVR) [37]. Yuan et al. [38] used two univariate models, the ARIMA model and the GM (1,1) model, to predict China's primary energy consumption. ARIMA-ANFIS [39] uses three models to predict annual energy consumption in Iran, applying the AdaBoost (adaptive boosting) data diversification model to address data deficiencies. However, most of such models are limited to linear univariate time-series and cannot solve the MTS problem well. In order to predict MTS data, vector autoregressive (VAR) models based on autoregressive were proposed. Later vector autoregressive moving average (VARMA) model, which combines VAR and moving average, was proposed. For example, GM(1,1)-VAR(1) model [40], which combines a VAR and a gray model, is proposed for forecasting residential electricity demand in Cameroon. Although VAR and VARMA are widely used in multivariate time-series forecasting tasks, both are linear regression models that cannot capture the nonlinear relationships in time-series data. Based on this, models based on kernel methods [41], ensembles [42], Gaussian processes [43] have been used for multivariate time-series forecasting. Although these models can express nonlinear relationships to a certain extent, the settings of kernel functions and related parameters need to be based on domain knowledge, so they cannot adapt to MTS data under different tasks.

2.3.2. Emerging deep learning techniques in multivariate prediction

With the development of deep learning, many neural network models have been proposed and applied to multivariate building energy consumption prediction tasks. A large number of studies have shown that deep learning-based methods perform better than statistical methods. Mainstream models addressing multivariate time-series prediction can be broadly categorized into three groups: **(I)** Hybrid models based on CNN and RNN, where CNN capture cross-dimensional dependencies and RNN capture cross-temporal dependencies. **(II)** Using Graph Neural Network (GNN) to capture cross-dimensional dependencies and using time-series convolution (e.g., RNN, LSTM, etc.) to capture cross-time dependencies. **(III)** Transformer based on attention mechanism captures both cross-dimensional and cross-time dependencies.

Hybrid Models based on CNN and RNN for MTS Forecasting. CNN is widely used in the image domain due to their strong local perception and multi-layer abstraction capabilities. In addition, the convolution operation in CNN can learn a set of dimension-specific features from each channel and then capture cross-dimensional dependencies by fusing the outputs of these channels. This multi-channel processing can better utilize the correlations between different dimensions in the input data. As a result, CNN is also widely used to

capture cross-dimensional dependencies in building energy consumption data. RNN with its recurrent connection structure makes it memory capable to remember past information and use it for current computation [44]. Therefore, RNN is widely used to process sequential data such as machine translation, speech recognition, time-series prediction, etc. However, RNN faces gradient vanishing and gradient explosion problems, and its short-term memory makes it difficult to capture long-term dependencies. Therefore, LSTM that incorporate gate structures has been proposed to solve these problems. Due to the respective strengths of CNN and LSTM, many studies have fused them to improve the model to enhance multivariate time-series prediction accuracy. A large number of previous studies have applied such methods to energy consumption prediction in the building field [45,46]. For example, Kim et al. [45] proposed a CNN–LSTM hybrid network model to effectively predict housing energy consumption. Experiments show that a neural network combining CNN and LSTM can extract complex energy consumption features. CNN layers can extract features among multiple variables that affect energy consumption, while LSTM layers are suitable for modeling temporal information of irregular trends in time-series components.

GNN for MTS Forecasting. GNN has achieved great success in dealing with spatial dependencies between entities in a network. Another spatio-temporal graph neural network (STGNN) based on GNN has been proposed and offers great advantages in dealing with multivariate time-series data. This form of neural network was originally proposed to solve the traffic prediction problem [47,48]. Its input is a multivariate time-series with an external graph structure, which describes the relationships between variables in the multivariate time-series. In STGNN, dimensional dependencies between nodes are captured by graph convolution, while temporal dependencies between historical states can be captured by RNN. STGNN is widely used in transportation, finance, medicine, and other fields. In the field of buildings, the main object of previous work is the spatial dependence between different buildings [49,50], but there are fewer studies on multivariate time-series data for a single building. It is worth noting that many GNN-based MTS prediction methods assume that the predicted value of a single variable is affected by all other variables, ignoring the causal relationship between variables. To address this issue, CauGNN [51] introduces neural Granger causality to GNN and uses convolutional neural network filters with different perceptual scales for time-series feature extraction.

Transformers for MTS Forecasting. Last few years, many variants of transformer have been proposed to significantly improve the performance of various tasks, such as natural language processing [52], computer vision [53], speech recognition [54], etc. Transformers have powerful modeling capabilities for long-term dependencies and correlations in sequence data. Recently, many transformer-based models have been proposed for MTS prediction and have shown great performance [55–57]. Preformer [56] introduces a novel and efficient multi-scale segmentation correlation mechanism, which divides the time series into several segments and uses segmentation-based correlation attention to replace point-based attention.

The quadratic complexity of the attention mechanism leads to high model training costs, which prevents its direct application to long series prediction. Many models have been proposed to reduce the time complexity. For example, Informer [55] uses the low-rank property of the self-attention matrix to reduce complexity, and it selects Q with high similarity based on the similarity between Q and K. In addition, timestamp information is an important component of multivariate time series, such as time of day, day of week, day of month, month of year, etc. G.Zerveas et al. [57] introduce temporal information via a learnable embedding layer that encodes timestamps into positional encodings, while the layer learns the embedding vectors for each location along with other model parameters.

In order to model periodic patterns in time-series data at different time scales, a hierarchical structure has recently been introduced into transformer. Pyraformer [58] devised an attention mechanism based on a tree structure, where fine-grained nodes correspond to the original sequences and coarse-grained nodes denote the low-resolution sequences. Pyraformer proposes an inter-scalar tree structure to capture features at different resolutions, while in-scale neighboring connections simulate temporal dependencies at different scales. In addition to the ability to integrate information at different multi-resolutions, hierarchical architectures have the advantage of being computationally efficient, especially for long time sequences. Informer [55] takes into account sequence patterns at different resolutions by adding a max pooling layer with a stride of 2 between attention blocks to downsample the sequence to half of it. However, to the best of our knowledge, there are no transformer-based studies that consider multi-scale periodic patterns of building energy consumption sequences from a frequency domain perspective.

3. Dataset and problem formulation

3.1. Dataset

To address the problem of forecasting energy demand with high precision, we collected a real-world high-resolution energy demand dataset from a typical large commercial office complex in Beijing, China. The modeling drawings of the office buildings are shown in (Fig. 2a). The building has 18 floors, with a standard floor area of more than 3400 square meters. There are 849 parking spaces, and nearly half of them are equipped with charging piles. At the same time, the building is equipped with air conditioning systems, VAV variable air volume systems, PM2.5 elimination systems, etc. (Fig. 2b) is the VAV sensor layout on the 16th floor of the building.

The commercial office building is equipped with a state-of-the-art sensor system that captures real-time data, ensuring the integrity and high resolution of our dataset. This dataset encompasses three years of hourly energy consumption data, from July 1, 2020, to July 1, 2023, yielding a total of 13,140 data points. Energy usage within the building is categorized into 12 distinct areas (Fig. 3), including energy consumption for the heating and cooling stations, air conditioning terminals, ventilation systems, elevators, water pumps, water features, kitchens, information centers, electric vehicle charging stations, cinemas, lighting and sockets, and corridor emergency lighting. Detailed statistical information of this dataset can be found in Table 1.

This dataset is instrumental in validating our approach, providing a representative cross-section of energy usage patterns commonly observed in such settings. Its size and scope are particularly well-suited for evaluating the performance of sophisticated deep learning models, which require substantial amounts of detailed data to capture and learn from the complex, nonlinear interplay of factors affecting energy demand. By leveraging this typical yet high-resolution, reliable, and extensive dataset, our research stands to make significant strides in the accurate prediction of energy usage, offering potential for substantial advancements in energy management practices for large commercial buildings.

3.2. Problem formulation

In multivariate time series forecasting of building energy consumption, we aim to predict the future time series $\hat{\boldsymbol{X}} = X_{L+1:L+\tau} \in \mathbb{R}^{\tau \times d_x}$ given the historical series $\boldsymbol{X} = X_{1:L} \in \mathbb{R}^{L \times d_x}$, i.e.,

$$\hat{\boldsymbol{X}} = f(\boldsymbol{X}). \tag{1}$$

where τ, L are the number of future and past time steps, respectively, and $f(\cdot)$ denotes the prediction mapping function. d_x is the number of dimensions, and $d_x = 12$ in this work represents the building energy consumption of 12 categories.

Additionally, the input signal can be combined with timestamp information, such as year, month, day, etc. In this work, instead of

Table 1
Statistical information on the energy consumption of the 12 categories in 2022. The table sorts all categories of energy consumption from large to small according to the mean value.

Energy consumption category	Maximum values (kWh)	Minimum values (kWh)	Mean value (kWh)	Standard deviation (kWh)
Lighting sockets	942.49	105.33	518.35	226.99
Heating and cooling stations	2299.11	24.14	354.10	612.39
Information centers	445.98	156.56	354.05	32.94
Air conditioner terminals	586.71	16.80	192.78	151.70
Ventilation	303.05	45.93	151.65	48.61
Kitchens	289.37	8.36	105.34	80.76
Elevators	80.24	4.44	32.24	18.69
Cinemas	79.10	3.12	26.89	21.70
Corridor emergency lighting	18.47	6.80	14.02	1.04
Charging piles	108.57	0.03	12.39	18.08
Water landscapes	34.88	4.09	12.23	4.18
Water pumps	40.61	3.35	11.86	3.89

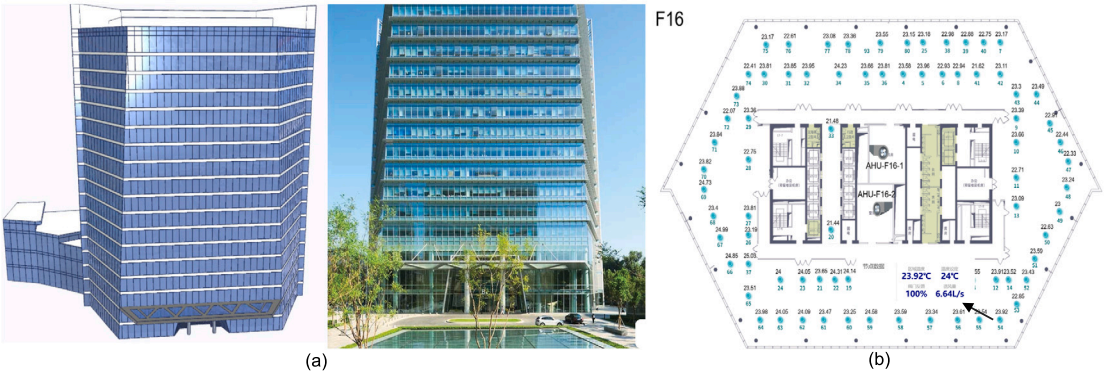

Fig. 2. (a) Modeling diagram of a typical large commercial office building in Beijing, China. The building has 18 floors, 80 m high and 96,983 square meters. (b) The VAV system sensors layout on the 16th floor of the building. The parameters read by the sensors include: actual regional temperature, set temperature, air supply volume, valve opening and other parameters.

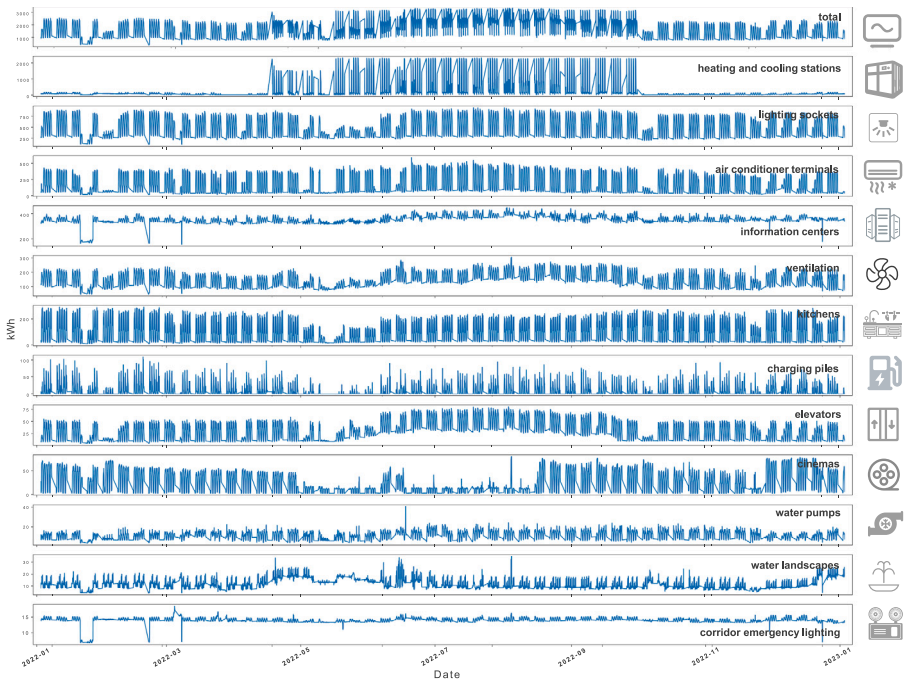

Fig. 3. Hourly resolution energy consumption curves for 12 categories in 2022.

directly treating the timestamp as a separate dimension, we encode it into the energy consumption data with the aim of improving prediction performance. Formally, we define $\hat{\boldsymbol{T}} = T_{L+1:L+\tau} \in \mathbb{R}^{\tau \times d_t}$ and $\boldsymbol{T} = T_{1:L} \in \mathbb{R}^{L \times d_t}$ as the future and past timestamp information respectively, where D is the dimension of the timestamp. We introduce a learnable nonlinear embedding layer to map $\boldsymbol{X}$ and $\boldsymbol{T}$ to the latent space: $\boldsymbol{X}, \boldsymbol{T} \rightarrow H \in \mathbb{R}^{L \times d_{model}}$, where d_{model} represents the latent space dimension. See Section 4 for details.

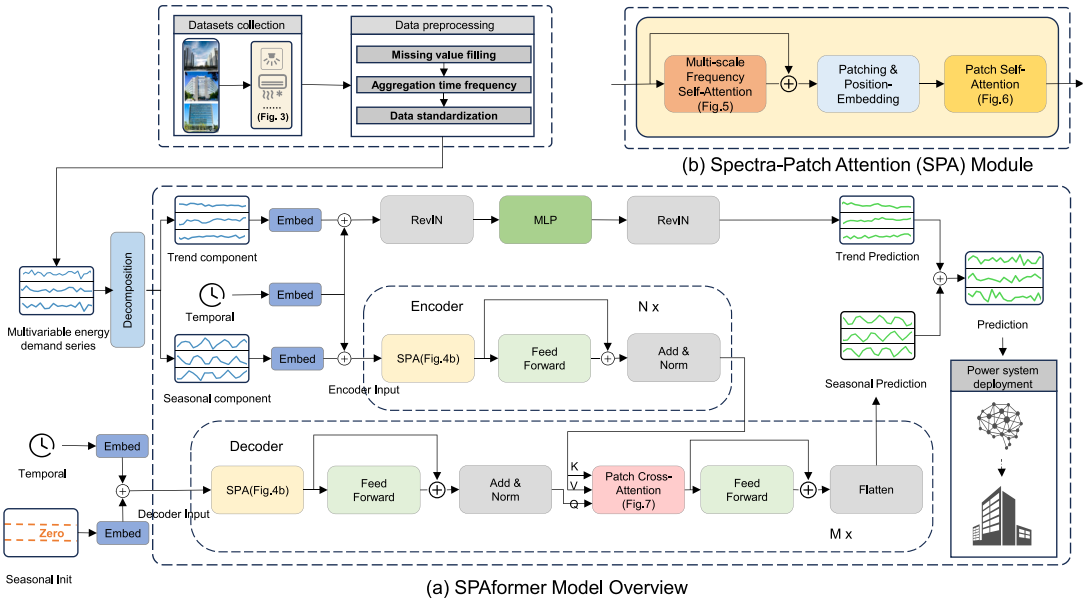

Fig. 4. SPAformer network framework. (a) Collecting energy consumption data from Rongke commercial buildings through sensors. The dataset is preprocessed, including missing value filling, aggregation time frequency and data normalization. We feed the processed dataset into the SPAformer network for model training. (b) Spectra-Patch Attention (SPA) mechanism including multi-scale frequency self-attention, patch self-attention and patching operation.

4. Methodology

In this chapter, we will introduce the overall structure of SPAformer. Then each module is introduced in detail, including data preprocessing, time encoding, sequence decomposition block, trend prediction, cycle prediction and pectra-patch attention (SPA) mechanism.

4.1. Overview

The accuracy of building energy data collection is highly dependent on the stability of the sensors. Uncertain factors in the collection process cause problems such as local missing data and noise. We use various preprocessing methods to improve the accuracy of the input data, as shown in Fig. 4. We then feed the processed data into a sequence decomposition block to achieve period and trend disentanglement. Energy consumption data is highly correlated with time information. We propose to encode the temporal information through an embedding layer and embed it into the period and trend components respectively. After that, the trend component undergoes the ReVIN module to solve the distribution bias and undergoes the MLP to learn the future trend. The periodic components embedded with temporal information are fed into the encoder to learn periodic representations. In the encoder, we propose SPA, which collaboratively utilizes time-domain and frequency-domain signals to enhance the ability to capture multi-scale periodic patterns of long sequence data. At the same time, the early stage of the decoder uses the same method as the encoder. Then, patch-cross attention is used to improve the similarity between the latent space states of the decoder and encoder. Finally, we aggregate trend and cycle forecasts to obtain future energy consumption series. The detailed introduction and formula definitions of each module are provided below.

4.2. Data preprocessing

As part of the modeling process, preprocessing input data can improve the accuracy and reliability of data prediction results. The collection of building energy consumption data is highly dependent on the stability of the sensor network, making the process subject to many uncertainties. The process can lead to problems such as incomplete, noisy, and inconsistent stored data. This study uses moving average and normalization methods to preprocess building energy consumption data, as shown in (Fig. 4a):

1. **Missing value filling:** Missing values and outliers are filled in through the moving average method, and exponential weighting is used to calculate the mean, which gives greater weight to the nearest data points.
2. **Aggregation time frequency:** The original dataset is timestamped at 15 min, which is too fine-grained for our research. Therefore, we aggregate it to 1-hour timestamps by downsampling.
3. **Data standardization:** Standardization can reduce model prediction errors and improve convergence speed without changing the distribution of original data. We use the library function StandardScaler in sklearn to calculate the mean and standard deviation of each dimension of the training set, and then apply them to both the training set and the test set.

4.3. Sequence decomposition block

Long series data of building energy consumption need to be disentangled from the intertwining of cycles and trends and to learn complex temporal patterns. We adopt the idea of sequence decomposition to decompose the sequence into periodic and trend components. These two parts reflect the long-term progression and cyclical fluctuations of the series respectively. Specifically, we use a moving average to smooth the periodicity to get the trend, and then subtract the trend from the original series to get the periodicity. Assuming that the past observation is $\boldsymbol{x} \in \mathbb{R}^{L \times d_x}$, where L and d_x are the input sequence length and dimension respectively. The process is defined as:

$$\begin{aligned} x_t &= \mathrm{AvgPool}(\mathrm{Padding}(\boldsymbol{x})), \\ x_s &= x - x_t. \end{aligned} \tag{2}$$

where $x_s, x_t \in \mathbb{R}^{L \times d_x}$ are the period and trend component respectively. We perform a mean padding operation on the original sequence, and then use $\mathrm{AvgPool}(\cdot)$ to perform a moving average to obtain the trend term.

4.4. Time embedding

There is a strong correlation between office building energy consumption data and time characteristics. It is a prerequisite that the time data of the future sequence is known. Reasonable embedding of time information can effectively improve the accuracy of the model. This paper proposes to encode and embed different temporal features into trend and period components through an embedding layer.

We project the trend and period components into the latent space separately and fuse the temporal information $\boldsymbol{x}_{time} \in \mathbb{R}^{L\times d_t}$, where $d_t = 7$ denotes the dimension of the timestamp sequence, including hours of the day, days of the week, days of the month, weeks of the month, days of the year, weeks of the year, months of the year at each time point. The process is defined as:

$$\begin{aligned} x_{t,embed} &= \mathrm{Embed}_{\mathrm{value}}(x_t) + \mathrm{Embed}_{\mathrm{time}}(\boldsymbol{x}_{time}). \\ x_{s,embed} &= \mathrm{Embed}_{\mathrm{value}}(x_s) + \mathrm{Embed}_{\mathrm{time}}(\boldsymbol{x}_{time}). \end{aligned} \tag{3}$$

where $x_{t,embed}, x_{s,embed} \in \mathbb{R}^{L\times d_{model}}$ are the trend and period components obtained by the sequence decomposition block respectively and d_{model} is the dimension of the latent space. We use two linear layers for the embedding layer $\mathrm{Embed}(\cdot)$ to project multi-dimensional time series and timestamp sequences into the latent space with the same number of channels. The same embedding layer is applied in the subsequent decoder part.

4.5. Trend forecast

The attention model cannot extrapolate linear trends well and has large errors because the natural attention mechanism works by interpolating context history. TFDformer [23] also theoretically proves that for trend data, with the polarization effect of softmax, the attention score emphasizes low-frequency components more and produces misleading reconstruction results. In contrast, MLP perfectly predicts this trend signal. Therefore, for the prediction of trend items, we use a five-layer MLP to predict future trends. The symbols are defined as follows:

$$x_{trend} = \mathrm{RevIN}(\mathrm{MLP}(\mathrm{RevIN}(x_{t,embed}))). \tag{4}$$

where $x_{trend} \in \mathbb{R}^{\tau\times d_x}$ is the prediction result of the trend prediction branch in SPAformer for the future trend and τ is the sequence prediction length.

4.6. Period forecast

As shown in (Fig. 4a), we use an encoder–decoder architecture to predict periodicity. In the encoder, we propose the spectra-patch attention mechanism, which captures the mixed patterns of long and short periods while effectively capturing the short-range dependencies of time series. Specifically, we first feed the periodic component to the N-layer encoder:

$$\begin{aligned} x_{en}^{l,1} &= \mathrm{SPA}(x_{en}^{l-1}), \\ x_{en}^{l,2} &= \mathrm{Norm}(\mathrm{FeedForward}(x_{en}^{l,1}) + x_{en}^{l,1}), \\ x_{en}^{l} &= x_{en}^{l,2}, l = 1, 2, \ldots, N. \end{aligned} \tag{5}$$

where $x_{en}^{0} = x_s$, and x_{en}^{l} denotes the output of the lth encoder layer. SPA is the spectra-patch attention module, and its input and output are both in the time domain. We will describe SPA in detail in the next subsection, which replaces the classic point-wise self-attention mechanism. $\mathrm{FeedForward}$ represents feed-forward neural network, and Norm denotes the usual tricks after residual connection, including linear mapping, activation function, dropout, etc.

In the decoder, we similarly introduce the SPA module. At the same time, we propose the patch cross-attention mechanism to guide the model to improve the prediction accuracy by increasing the similarity between the hidden space states of the decoder's input sequence and the encoder's input sequence. Specifically, we fill the future part of the periodicity with zeros and feed it to the M-layer decoder:

$$\begin{aligned} x_{de}^{l,1} &= \mathrm{SPA}(x_{de}^{l-1}), \\ x_{de}^{l,2} &= \mathrm{Norm}(\mathrm{FeedForward}(x_{de}^{l,1}) + x_{de}^{l,1}), \\ x_{de}^{l,3} &= \mathrm{PatchCrossAttention}(x_{de}^{l,2}, x_{en}^{N}), \\ x_{de}^{l,4} &= \mathrm{Flatten}(\mathrm{Norm}(\mathrm{FeedForward}(x_{de}^{l,3}) + x_{de}^{l,3})), \\ x_{de}^{l} &= x_{de}^{l,4}, l = 1, 2, \ldots, M. \end{aligned} \tag{6}$$

where $x_{de}^{0} = \mathrm{Padding}(x_s)$, which means padding the future period components with zeros as input to the decoder, and x_{de}^{l} denotes the output of the l-th decoder layer. $\mathrm{PatchCrossAttention}$ is the patch cross-attention mechanism in the decoder, which calculates the similarity between the decoder and encoder sequences at the patch level. $\mathrm{Flatten}$ means to unfold the output of the cross-attention mechanism according to multiple-heads, and then connect a linear layer to map to the target sequence length.

Finally, we add the predictions from the trend branch and the periodic branch to get the final prediction output, i.e. $\hat{\boldsymbol{x}} = x_{trend} + x_{de}^{M}$, where $\hat{\boldsymbol{x}} \in \mathbb{R}^{\tau\times d_x}$. We choose mean squared error (MSE) as the optimizer to measure the discrepancy between the prediction and the ground truth. The loss in each dimension is computed and averaged over d_x time series to get the overall target loss:

$$\mathcal{L} = \mathbb{E}_x \frac{1}{d_x} \sum_{i=1}^{d_x} \| \hat{\boldsymbol{x}}_{L+1:L+\tau}^{(i)} - \boldsymbol{x}_{1:L}^{(i)} \|_2^2. \tag{7}$$

4.7. Spectra-Patch Attention (SPA) module

The long-term series of building energy consumption has complex periodic and local dynamic change patterns. We propose the Spectra-Patch attention module, which collaboratively utilizes time-domain and frequency-domain signals to enhance the ability to capture multi-scale periodic patterns in long sequence data. As shown in (Fig. 4b), we innovatively propose a multi-scale frequency self-attention mechanism for frequency domain signals and a patch self-attention mechanism for time domain signals. They are connected in series to form the SPA module, which is introduced into the encoder and decoder. The two modules are described in detail below.

4.7.1. Multi-scale frequency self-attention mechanism

Time domain signals are often affected by capturing long-range dependencies. When data has strong periodicity and long-distance correlation, this related information may not be correctly captured by time series signals. To this end, we propose a multi-scale frequency self-attention mechanism as a supplement to enhance the model's ability to capture long-range dependencies. Specifically, we directly feed the complete input sequence into the frequency attention module. First, we use FFT to obtain the complex domain sequence. The low-frequency part of different frequency components reflects the medium and long periods, while the high-frequency part contains noise and local detail information. In view of this feature, we divide the frequency into multiple intervals from low to high. The length of each interval increases step by step, and different weights are assigned to it. Then the attention calculation is performed on different frequency segments separately. In addition, considering that information loss may occur between the FFT and iFFT processes. In order to make up for the missing information, inspired by RestNet [59], we introduce a residual connection between the input and output of the frequency self-attention module, as shown in (Fig. 4b). This helps to recover the information that may be lost in the FFT and iFFT processes during time domain reconstruction.

The specific formula definition of the module is given below. As shown in Fig. 5, in order to obtain frequency domain features, the input sequence first needs to be time-frequency transformed using discrete Fourier transform (DFT). Given a sequence x_n in the time

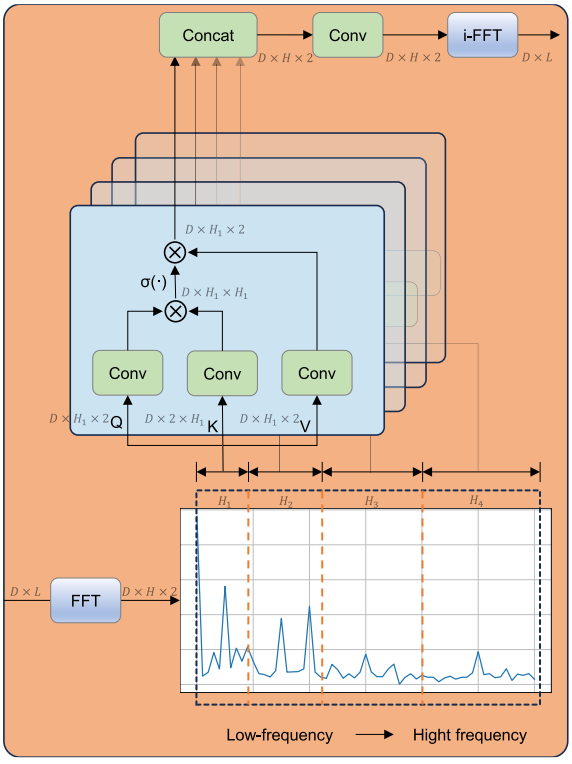

Fig. 5. Multi-scale frequency self-attention.

domain, where $n = 1, 2...L$. DFT is defined as $x_f = \sum_{n=0}^{N-1} x_n e^{-i\omega ln}$, where i is the imaginary unit and x_f, $f = 1, 2...H$, is a sequence of complex numbers in the frequency domain. Similarly, the inverse Discrete Fourier transform (iDFT) is defined as $x_n = \sum_{f=0}^{H-1} x_f e^{i\omega ln}$. To avoid $\mathcal{O}(L^2)$ complexity of DFT, we use Fast Fourier transform (FFT) and inverse transform (iFFT), whose complexity is reduced to $\mathcal{O}(L \log L)$. The complex matrix is obtained after FFT transformation. In order to facilitate matrix calculation, we extract and stack the real part and imaginary part respectively to obtain $x_f \in \mathbb{C}^{D \times H \times 2}$, where $H = \frac{L}{2} + 1$ represents the length of the complex sequence in the frequency domain, and 2 in the last dimension represents the real part and imaginary part.

As mentioned above, in order to extract the long-term periodic characteristics and local detail information in sequence data, we developed a multi-head self-attention model driven by multi-scale frequency. This model divides the frequency signal of time series data into multiple frequency bands according to the frequency. The length of each frequency band increases with the increase of frequency, and each frequency band is assigned a corresponding weight to highlight its importance in sequence analysis. The formula is as follows:

$$\mathrm{FFT}(x_{en}^{l-1}) = \mathrm{Concat}(H_1^{l-1}, H_2^{l-1} \dots H_m^{l-1}). \tag{8}$$

where H_i denotes the ith frequency interval divided from the complex sequence.

We feed each frequency interval into a separate self-attention layer. Then we concatenate the different interval outputs, and use iFFT to convert the output into the time domain. The specific definition is as follows:

$$\mathrm{Atten}_i(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) = \sigma\left(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^T}{\sqrt{d_k}}\right)\boldsymbol{V}_i, \tag{9}$$

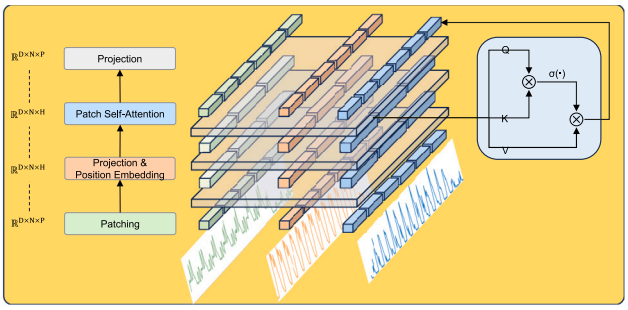

Fig. 6. Patch self-attention mechanism.

where $\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i = H_i^{l-1} \in \mathbb{C}^{D \times h_i \times 2}, i = 1, 2 \dots m$, h_i is the length of the ith intervals, and $\sigma(\cdot)$ denotes the softmax operation. Then, the multi-scale frequency self-attention module can be defined as:

$$x_f^{l-1} = \mathrm{iFFT}(\mathrm{Conv}(\mathrm{Concat}(\mathrm{Atten}_1, \mathrm{Atten}_2 \dots \mathrm{Atten}_m))), \tag{10}$$

4.7.2. Patch self-attention mechanism

The Patch self-attention mechanism is designed to extract local features in time-domain sequence data. This mechanism works by splitting the input sequence into multiple patches. During attention calculation, each patch is treated as an independent token. The advantage of this design is that it can simulate the concept of receptive fields in traditional convolutional networks while allowing the model to process data in a more fine-grained manner. In this way, the model can focus more on capturing local detailed information, rather than paying more attention to the global context like traditional self-attention mechanisms. This makes up for the shortcomings of the frequency attention mechanism in capturing local information. In addition, a key advantage of the Patch self-attention mechanism is its ability to process individual patches in parallel, which significantly improves computational efficiency. As shown in Fig. 6, we first need to perform a patching operation on the original sequence $x_{en}^{l-1} \in \mathbb{R}^{D \times L}$. Suppose the stride is S and the length of the patch to be split is P. Before performing the patching operation, we need to fill the input sequence with S numbers using end elements. The symbols are defined as follows:

$$x_{p,1}^{l-1} = \mathrm{Patching}(\mathrm{Padding}(x_{en}^{l-1})). \tag{11}$$

where $x_{p,1}^{l-1} \in \mathbb{R}^{D \times N \times P}$, and $N = \lfloor \frac{(L-P)}{S} \rfloor + 2$ is the number of patches. Then to make the patches more representational, we map them to the latent space via a learnable linear projection $\mathrm{M}_p \in \mathbb{R}^{D \times P \times H}$. Similar to the transformer model, we position encode each patch through $\mathrm{M}_{pos} \in \mathbb{R}^{D \times N \times H}$ to represent the temporal order between them. The formula is as follows:

$$x_{p,2}^{l-1} = \mathrm{M}_p\, x_{p,1}^{l-1} + \mathrm{M}_{pos}\,. \tag{12}$$

where $x_{p,2}^{l-1} \in \mathbb{R}^{D \times N \times H}$ is the sequence of patches in the hidden space. As in Eq. (9), we compute self-attention on the patches sequence and project it back to the dimensions of the input sequence, i.e.,

$$x_p^{l-1} = \mathrm{Projection}(\mathrm{Atten}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})), \tag{13}$$

where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} = x_{p,2}^{l-1}$, and $x_p^{l-1} \in \mathbb{R}^{D \times N \times P}$ is the output of the patch self-attention module.

4.8. Patch cross-attention mechanism

The encoder–decoder structure requires a cross-attention mechanism to calculate the similarity of the latent space states of energy sequences. we have previously extracted the frequency signal of time series data through the SPA module. To prevent model overfitting,

Table 2
MSE and MAE of multivariate long-term time-series forecasting with input context length $120h$ (i.e. 5 days) and forecasting horizon $\{120h, 240h, 360h, 480h, 600h, 720h, 840h\}$. The best results are in **bold** and the second best are underlined.

Models		SPAformer		PatchTST [16]		FEDformer [15]		Autoformer [21]		Informer [55]		Transformer [60]	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Length	120	**0.283**	0.342	0.290	**0.334**	0.307	0.370	0.346	0.397	0.644	0.546	0.444	0.446
	240	**0.330**	0.386	0.355	**0.378**	0.366	0.411	0.411	0.437	0.725	0.590	0.519	0.490
	360	**0.370**	**0.407**	0.409	0.409	0.410	0.438	0.436	0.458	0.769	0.624	0.512	0.489
	480	**0.390**	**0.420**	0.455	0.434	0.452	0.465	0.478	0.471	0.843	0.667	0.526	0.484
	600	**0.411**	**0.438**	0.493	0.451	0.487	0.487	0.503	0.491	0.806	0.636	0.554	0.521
	720	**0.435**	**0.454**	0.529	0.470	0.518	0.503	0.537	0.500	0.820	0.649	0.641	0.549
	840	**0.440**	**0.459**	0.555	0.485	0.525	0.506	0.575	0.530	0.830	0.660	0.653	0.549

we propose patch-cross attention in the time domain and no longer consider frequency domain signals. In this cross-attention, $\boldsymbol{K}$ and $\boldsymbol{V} \in \mathbb{R}^{N\times P}$ come from the encoder and are obtained through N encoder layers, see Eq. (5). $\boldsymbol{Q} \in \mathbb{R}^{M\times P}$ comes from the decoder, which is obtained by TFA, see Eq. (6). M denotes the number of patches divided by splitting with P length. The structure of patch cross-attention is shown in Fig. 7.

5. Experiments

5.1. Experimental settings

We conducted experiments on the energy consumption dataset of a commercial office building, as described in Section 3.1. The input of the model is a multivariate time series of energy consumption in 12 categories, and the output is a future prediction of energy consumption in these categories. In addition, we conduct a comparative study on the univariate prediction accuracy of total energy consumption.

We use the Adam optimizer where the learning rate is updated with the number of iterations. A large learning rate is used at the beginning of training to quickly approach the optimum. The learning rate is continuously reduced in the later stages of training to fine-tune the model parameters and avoid overfitting. Batch-size that is too small will cause the training speed to slow down, and too large will cause the accuracy to decrease. Considering the training environment and practical experience, we set the batch size to 32. In the patching operation, we set the patch-size to 48 (i.e. 2 days) and the step-size to 24 (i.e. 1 day). Experiments show that the patch of length 48 can capture sufficient historical information, while an overly large patch will result in slower training. The step size of 24 is to allow some overlap of each newly generated sample and increase the diversity of the data. We use the data from the whole year of 2022 as the training set, and the data from January to July 2023 as the test set. MSE and MAE are used as evaluation metrics of the model to evaluate the performance of the model from different perspectives. The experimental environment is Pytorch on NVIDIA Quadro RTX 8000 48 GB GPUs.

5.2. Comparison study

5.2.1. Baselines

We select the latest state-of-the-art transformer-based models as our baselines, including FEDformer [15], PatchTST [16], Autoformer [21], Informer [55] and classic Transformer. All models follow the same experimental setup.

To understand how well our forecasting models perform over varying future time periods, we set up our experiments with a consistent starting point: we use an input length of $L = 120h$. This means we are looking at data from the past 5 days (assuming each unit in L represents an hour, which is common in such studies) as the basis for making our predictions. Then, we test how accurate these models are at predicting energy use for different lengths of time into the future. Specifically, we are interested in seeing how well they can forecast the next 120 h (5 days), 240 h (10 days), 360 h (15 days), 480 h (20 days), 600 h (25 days), 720 h (30 days), and 840 h (35 days). In other words, we are

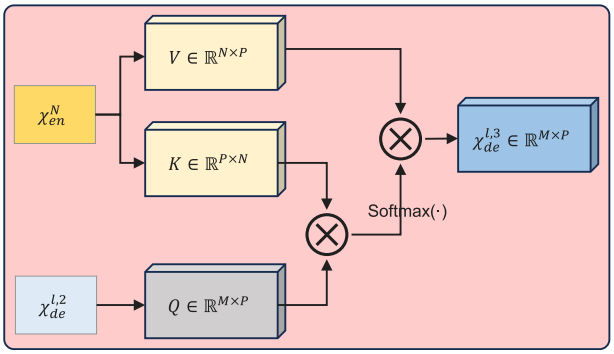

Fig. 7. Patch cross-attention mechanism.

checking the performance of our models when they are tasked with predicting energy consumption from as short as the next 5 days to as long as the next 35 days, based on the past 5 days of data.

This approach lets us compare the models' effectiveness across a range of future time horizons, giving us a clear picture of how well each model can adapt and maintain accuracy over shorter and longer forecasting periods.

We compare the performance of the models from both quantitative and qualitative perspectives. In the quantitative analysis, we choose MSE and MAE metrics to evaluate the performance of SPAformer and baselines. By comparing the MSE and MAE of the model in the short-term (such as 5 days) and long-term (such as 35 days) prediction periods, we can understand the adaptability and robustness of different models under different prediction horizons. This provides reliable data support for energy management and planning. In the qualitative analysis, we compare the predicted curves of each model with the actual energy consumption curves. This approach allowed us to visually assess and compare each model's ability to capture trends and patterns in energy use and to provide a qualitative evaluation of each model's accuracy and reliability.

5.2.2. Results of multivariate energy consumption prediction

In order to understand how our forecasting model performs for multi-category energy consumption forecasts under different forecasting horizons, we compared it with the state-of-the-art baselines with respect to the itemized energy consumption forecasts. The results of the experiment are shown in Table 2. Overall, our model outperforms baseline methods. In terms of specific values, compared with the best results that the Transformer-based model can provide, SPAformer overall reduces MSE by 12% and MAE by 4%. And it can be seen from the experimental results that SPAformer still has great advantages in long-distance sub-item energy consumption prediction tasks. In addition, it is easy to find that the PatchTST [16] accuracy becomes worse than FEDformer [15] as the prediction length becomes longer. This further suggests that the periodicity of long time series is more easily captured in the frequency domain. In contrast, short sequence predictions are more sensitive to local dependence, which is more easily captured in

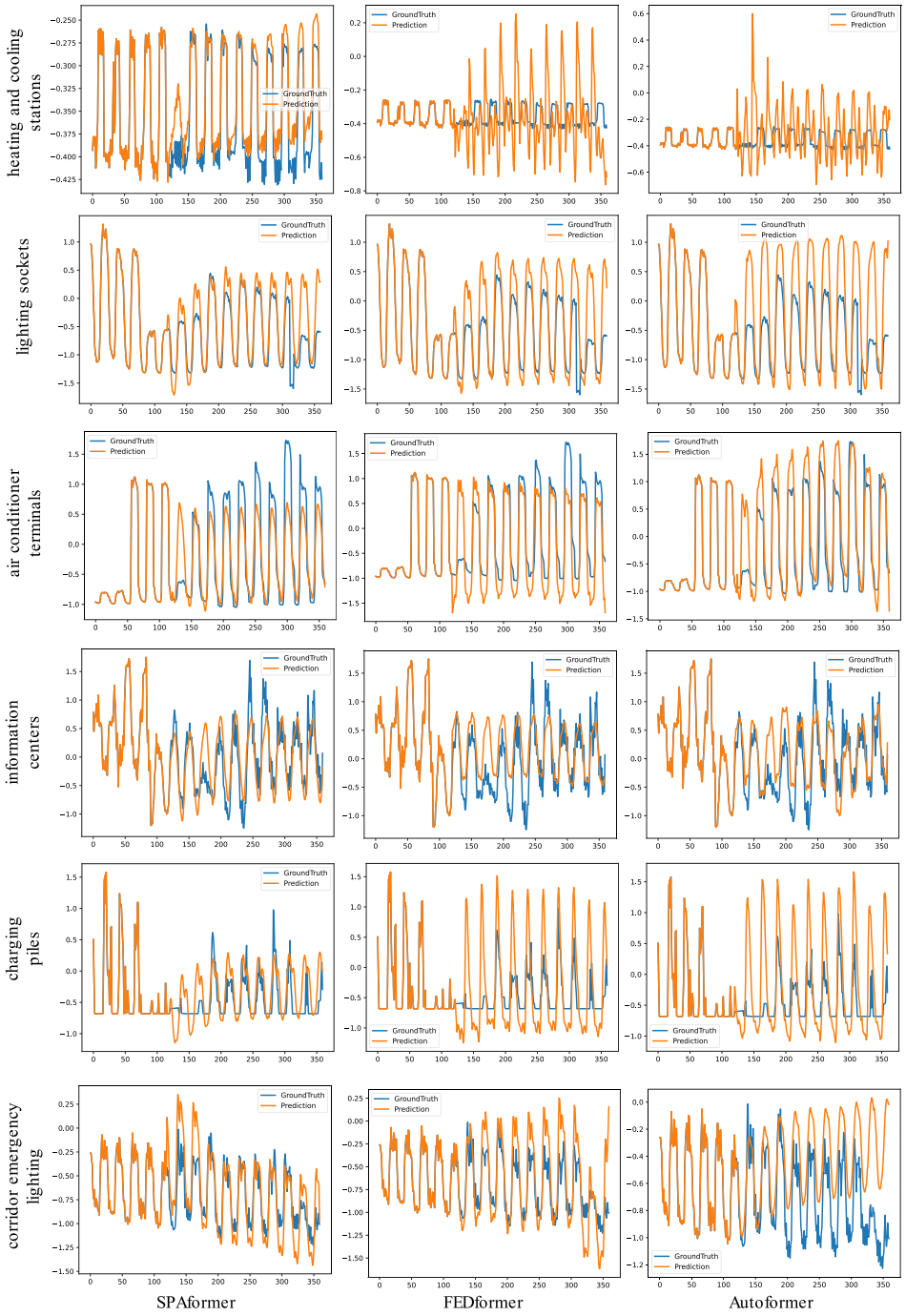

Fig. 8. Six category prediction cases of the building energy consumption dataset under the input-120*h*-predict-240*h* setting: heating and cooling stations, lighting sockets, air conditioning terminals, information centers, charging piles and corridor emergency lighting.

the time domain. Therefore, SPAformer, which introduces the spectra-patch attention mechanism, achieves state-of-the-art results in both short and long sequence prediction tasks.

In order to qualitatively compare the predictions of different models, we plot the prediction results from the test set of building energy consumption in six important categories, including heating and cooling stations, lighting sockets, air conditioning terminals, information centers, charging piles and corridor emergency lighting, Fig. 8. Our model shows the best performance among different models. In particular, SPAformer is significantly better than other models in predicting the power consumption of heating and cooling stations. Additionally, our model is better able to predict local details, periodicity and overall trends.

5.2.3. Results of univariate energy consumption prediction

The prediction of total building energy consumption is a univariate time series prediction task. Compared to itemized energy consumption, the total energy consumption series contains a single pattern, which is easier to be accurately predicted by the model. However, complex models may suffer from overfitting when capturing single or simple patterns. To explore SPAformer's ability to generalize the prediction of total energy consumption, we compared it with the baselines. The results of the experiment are shown in Table 3. Overall, SPAformer reduces the MSE by about 10% and the MAE by about 5% compared to state-of-the-art models (PatchTST [16], FEDformer [15], Autoformer [21]). In particular, the generalization ability of SPAformer

Table 3
MSE and MAE of univariate long-term time series forecast of total energy consumption with input context length $120h$ (i.e. 5 days) and forecasting horizon $\{120h, 240h, 360h, 480h, 600h, 720h, 840h\}$. The best results are in **bold** and the second best are underlined.

Models		SPAformer		PatchTST [16]		FEDformer [15]		Autoformer [21]		Informer [55]		Transformer [60]	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Length	120	**0.208**	**0.310**	0.213	0.315	0.209	0.326	0.285	0.374	0.249	0.350	0.280	0.335
	240	**0.227**	**0.331**	0.252	0.341	0.233	0.347	0.317	0.395	0.278	0.364	0.306	0.354
	360	**0.271**	**0.363**	0.299	0.374	0.282	0.374	0.350	0.407	0.290	0.386	0.327	0.377
	480	0.305	0.393	0.348	0.407	0.315	0.412	0.369	0.428	**0.300**	**0.391**	0.371	0.395
	600	0.321	0.417	0.397	0.436	0.376	0.434	0.393	0.446	**0.306**	0.403	0.360	**0.390**
	720	0.338	0.425	0.456	0.476	0.383	0.437	0.427	0.477	**0.319**	0.417	0.367	**0.390**
	840	0.355	0.461	0.504	0.507	0.424	0.466	0.479	0.519	**0.345**	0.428	0.395	**0.407**

is significantly better than the baselines in total energy prediction across long periods.

However, it is worth noting that in univariate long sequence prediction, Informer [55] and Transformer [60] outperform our model in most cases. Transformer is the original model to introduce the attention mechanism, which employs a point-by-point mechanism to compute the attention scores. Informer was proposed to address the high complexity of Transformer. It proposed ProbSparse self-attention based on the sparsity of the attention matrix. These two models have simpler structures than other baselines, so they can better avoid learning irrelevant information and introducing more noise. Although SPAformer's generalization ability in total energy consumption prediction is not optimal, this paper focuses more on the prediction of multivariate time series of itemized energy consumption. We believe that it is acceptable to sacrifice part of the generalization ability of total energy consumption prediction in exchange for high-precision prediction accuracy of multi-category energy consumption.

5.3. Sensitivity analysis

Different input sequence lengths usually affect the accuracy of energy consumption predictions. Too short input sequences contain less historical information, while too long input sequences introduce more noise and instability. On the other hand, the patch length and transformer hyper-parameter settings may also affect the stability of the model. Therefore, in this section, we conduct three aspects of sensitivity analysis: (I) exploring the performance of each model under different input lengths, (II) comparing the impact of different patch lengths on model performance and (III) hyper-parameter sensitivity.

5.3.1. Impact of input sequence length on model prediction performance

We performed sensitivity analysis on the input sequence length in the short-term and long-term energy consumption prediction tasks respectively. We set the input length $L \in \{120, 240, 360, 480, 600, 720, 840\}$. As shown in Fig. 9, in the short-term prediction, the prediction window length is $360h$. It is obvious that FEDformer [15], Autoformer [21], and Informer [55] show an increasing trend in MSE with increasing input length. This is because longer encoder inputs introduce more noise, resulting in these baselines not benefiting from the longer lookback window. The performance of our model is close to that of PatchTST [16] in short-term prediction, which shows that dividing time series into patches can capture richer local information.

In the long-term prediction task, the forecasting window is $720h$. We can easily find that for most baselines, as the input length increases, the MSE first decreases and then increases. Because longer time series contain more complex periodic patterns. PatchTST [16] has little change in MSE as the lookback window increases. This suggests that it takes into account local dependencies well but ignores longer periodic patterns. In our model, the ability to capture long and short periods is enhanced by introducing a multi-scale frequency correlation mechanism. Compared with the baseline, our model achieves better performance overall, although the MSE has an upward trend after the input sequence length is greater than 600. We analyze that when the lookback window is larger than 600, the noise and instability of the long sequence will be further aggravated, causing interference to the capture of the cycle.

5.3.2. Effect of patch length on model stability

In order to verify the stability of the SPAformer model, we compared the model prediction performance under different patch lengths. In the experiment, we fixed the lookback window to $120h$ and changed the patch length $P = \{8, 16, 24, 32, 40, 48, 56, 64\}$. The stride is set to half of the corresponding patch length so that each newly generated sample has a certain overlap and increases the diversity of the data. The model is trained to predict $120h$ and $360h$ step lengths. The experimental results are shown in Fig. 10. As the patch size increases, there is no significant difference in the MSE indicator. This shows that SPAformer is stable and robust to the patch length hyperparameter. In addition, it can be found that the prediction performance is best when the patch length is set to 48 on the building energy consumption dataset.

5.3.3. Hyper-parameter sensitivity

To study the sensitivity of SPAformer to the transformer parameter settings, we experimented with different combinations of model parameters. We set the number of encoders $N = \{1, 2, 3\}$ and the number of decoders $M = \{1, 2, 3\}$. In addition, we set the model latent space dimension $d_{model} = \{128, 256\}$ and the number of heads of the attention mechanism $head = \{4, 8\}$. There are a total of 12 different combinations of model hyper-parameters. Fig. 11 shows the MSE scores on the 12 hyper-parameter combinations, where the input sequence is fixed to $120h$ and the future horizon is $120h$ and $360h$. It can be seen that SPAformer is highly robust to the choice of hyper-parameters regardless of whether the prediction horizon is $120h$ or $360h$. Of course, in the second parameter combination $(N, M, d_{model}, head) = (1, 1, 128, 8)$, the model achieves the best performance. We followed this hyper-parameter setting in the previous experimental section.

5.4. Ablation experiment

In this subsection, we perform ablation experiments on SPAformer, which includes three parts: (I) We compare the prediction performance of the model using different decomposition methods and without decomposition. (II) Compare the impact of MLP and different self-attention mechanisms on the accuracy of trend prediction. (III) Compare the accuracy of SPA and state-of-the-art attention mechanisms on cycle component prediction to explore the effectiveness of the SPA module.

5.4.1. Trend-period decomposition block

We first performed ablation experiments on the sequence decomposition block to understand the performance improvement it brings. We compare the performance under five settings (I) Feeding the input sequence directly to the encoder–decoder without sequence decomposition. (II) Using the moving average method for sequence decomposition. (III) Using exponential smoothing method for sequence decomposition. (IV) Using the prediction result of the trend component as the final output without considering the period component. (V) Using the prediction of the period component as the final output without considering the trend component. The experimental results are shown in Table 4. The performance of the model that introduces the sequence

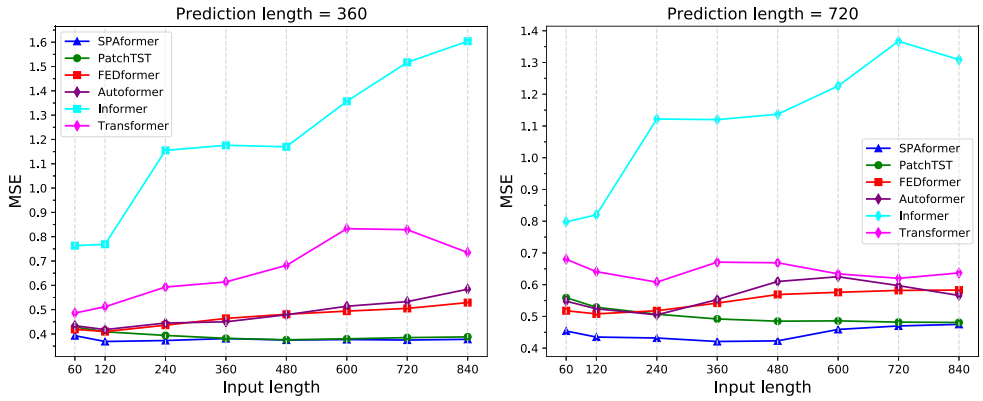

Fig. 9. Prediction performance (MSE) under different lookback windows. The lookback window is set to $\{60h, 120h, 240h, 360h, 480h, 600h, 720h, 840h\}$ and the prediction range are 360 and 720. We select the state-of-the-art models as the baselines.

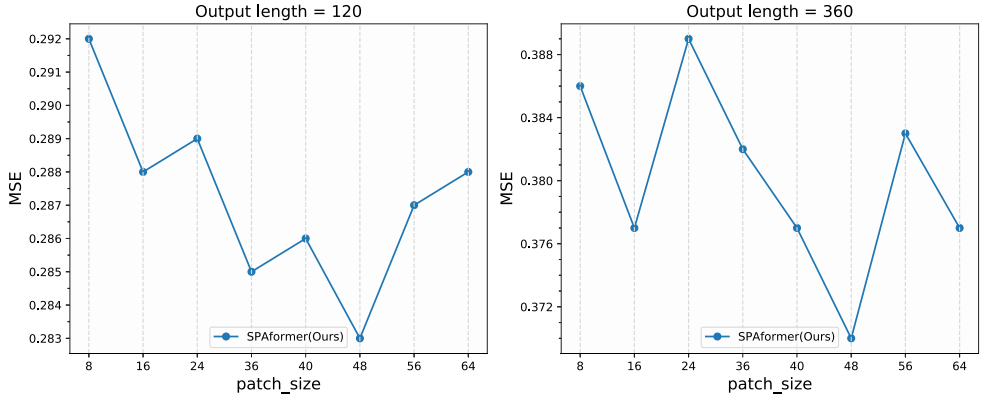

Fig. 10. MSE scores for different patch lengths $P = \{8, 16, 24, 32, 40, 48, 56, 64\}$, where the input sequence length is $120h$ and the prediction lengths are $120h$ and $360h$.

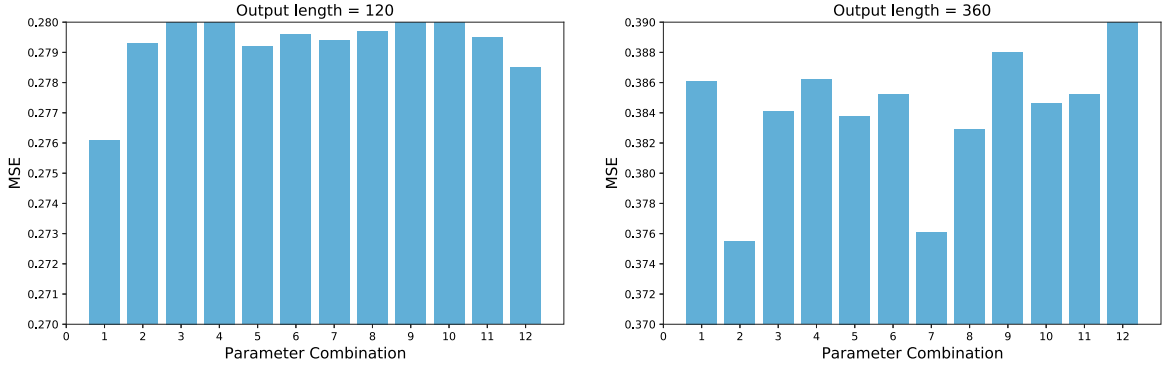

Fig. 11. MSE scores for different hyper-parameter combinations. The combinations $(N, M, d_{model}, head)$ = (1, 1, 128, 4), (1, 1, 128, 8), (1, 1, 256, 4), (1, 1, 256, 8), (2, 2, 128, 4), (2, 2, 128, 8), (2, 2, 256, 4), (2, 2, 256, 8), (3, 3, 128, 4), (3, 3, 128, 8), (3, 3, 256, 4), (3, 3, 256, 8) are labeled 1 to 12 in the figure in order. N and M are the number of encoders and decoders respectively, d_{model} represents the latent space dimension and head is the number of attention heads. The model fixes the input length to 120, and the future horizons are 120 and 360 respectively.

decomposition block is significantly better than that of direct forecasting, especially for long time series. In addition, using the prediction of cycle or trend components as the output of the model does not lead to optimal performance. Because this will ignore the joint impact of both on long sequence prediction. On the contrary, optimal performance can be achieved by using different forecasting methods for the cycle and trend components and aggregating the forecast results of the two to obtain a future sequence. In order to choose a more appropriate decomposition method, we compare moving average and exponential smoothing. Experiments show that moving smoothing is more suitable for our building energy consumption data.

5.4.2. MLP vs self-attention in trend prediction

In this subsection, we conduct ablation experiments on the trend component prediction method aiming to compare the performance of MLP and its alternatives. Three SOTA attention mechanisms are used as comparison models. Similarly, we fix the input sequence length to 120 and the prediction length to $\{60h, 120h, 240h, 360h, 480h, 600h, 720h, 840h\}$. The results are shown in Table 5. The results show that the MLP-based trend component prediction method we adopted achieved SOAT results and is significantly better than other models. Unlike the periodic component, although the trend component is the main part of the time series, the pattern it contains is simpler. All attention-based models will overfit the trend, which will produce large errors. In contrast, MLP has an advantage in predicting such trend signals.

Table 4
Ablation experiments for trend-periodic decomposition block: **SPAformer w/o Decomp** means that the model does not use serial decomposition. **SPAformer-MA/SPAformer-ES means** using the moving average/exponential smoothing method for sequence decomposition. **SPAformer w/o Seasonal/Trend** means trend/periodic forecast only. The input length of the model is fixed at 120, and the output length is $\{60h, 120h, 240h, 360h, 480h, 600h, 720h, 840h\}$.

Models	Sequence decomposition block		Output length													
	Moving average	Exponential smoothing	120		240		360		480		600		720		840	
			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
SPAformer w/o Decomp			0.290	**0.334**	0.350	**0.375**	0.405	0.408	0.448	0.432	0.489	0.452	0.539	0.471	0.556	0.489
SPAformer-MA	✓		**0.283**	0.342	**0.330**	0.380	**0.370**	**0.407**	**0.390**	**0.420**	**0.411**	**0.438**	**0.435**	**0.454**	**0.440**	**0.459**
SPAformer-ES		✓	0.322	0.374	0.362	0.401	0.406	0.430	0.435	0.448	0.457	0.462	0.473	0.478	0.504	0.493
SPAformer w/o Seasonal	✓		0.305	0.359	0.364	0.403	0.409	0.433	0.438	0.452	0.459	0.464	0.476	0.474	0.488	0.483
SPAformer w/o Trend	✓		0.458	0.478	0.473	0.488	0.483	0.491	0.487	0.491	0.493	0.496	0.499	0.492	0.502	0.496

Table 5
Ablation experiments of trend prediction: Comparison of MLP and three SOTA attention mechanisms on trend component prediction. **SPAformer-MLP-SPA** uses MLP to predict the trend component. **SPAformer-PatchAtt-SPA/SPAformer-AutoCorr-SPA/SPAformer-ProbAtt-SPA** uses patch attention/auto-correlation/prob attention mechanism to predict trend components. SPA is applied equally to periodic to component predictions. The input length of the models is fixed at 120, and the output length is $\{60h, 120h, 240h, 360h, 480h, 600h, 720h, 840h\}$.

Models	Trend				Output length											
	MLP	PatchAtt	AutoCorr	ProbAtt	120		240		360		480		600		720		840	
					MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
SPAformer-MLP-SPA	✓				**0.283**	**0.342**	**0.330**	**0.380**	**0.370**	**0.407**	**0.390**	**0.420**	**0.411**	**0.438**	**0.435**	**0.454**	**0.440**	**0.459**
SPAformer-PatchAtt-SPA		✓			0.405	0.451	0.462	0.481	0.520	0.508	0.508	0.511	0.540	0.524	0.525	0.509	0.541	0.512
SPAformer-AutoCorr-SPA			✓		0.338	0.398	0.483	0.503	0.560	0.554	0.600	0.577	0.614	0.588	0.605	0.580	0.594	0.577
SPAformer-ProbAtt-SPA				✓	0.406	0.451	0.502	0.518	0.589	0.567	0.628	0.589	0.633	0.593	0.628	0.587	0.629	0.587

Table 6
Ablation experiment of periodic prediction: forecasting results with input length $I = 120h$ and predicted length $O \in \{120h, 240h, 360h, 480h, 600h, 720h, 840h\}$. Three variants of TDFformer are compared to the baseline. The best and second best results are highlighted in bold and underlined respectively.

Models		Transformer [60]		Informer [55]		Autoformer [21]		SPAformer		SPAformer V1		SPAformer V2		SPAformer V3	
Self-att		FullAtt		ProbAtt		AutoCorr		SPA		ProbAtt		ProbAtt		SPA	
Cross-att		FullAtt		ProbAtt		AutoCorr		PatchAtt		PatchAtt		PatchAtt		AutoCorr	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Length	120	0.444	0.446	0.644	0.546	0.346	0.397	**0.283**	**0.340**	0.325	0.368	0.331	0.372	0.296	0.355
	240	0.519	0.490	0.725	0.590	0.411	0.437	**0.327**	**0.374**	0.348	0.390	0.369	0.409	0.349	0.395
	360	0.512	0.489	0.769	0.624	0.436	0.458	**0.371**	**0.406**	0.393	0.421	0.414	0.435	0.392	0.422
	480	0.526	0.484	0.843	0.667	0.478	0.471	**0.390**	**0.420**	0.446	0.456	0.449	0.452	0.412	0.439
	600	0.554	0.521	0.806	0.636	0.503	0.491	**0.411**	**0.435**	0.491	0.481	0.456	0.453	0.434	0.453
	720	0.641	0.549	0.820	0.649	0.537	0.500	**0.423**	**0.444**	0.527	0.503	0.497	0.481	0.440	0.455
	840	0.653	0.549	0.830	0.660	0.575	0.530	**0.440**	**0.448**	0.553	0.524	0.525	0.493	0.460	0.466

5.4.3. SPA vs SOTA attention mechanisms in period prediction

We propose to use an encoder–decoder structure for prediction of periodic components. The SPA module is introduced into the encoder to improve the capture ability of long and short period patterns, and the patch cross-attention is introduced into the decoder to improve the similarity score between the input and predicted sequences. In order to compare the performance of the method and its alternatives, we performed ablation experiments on different attention mechanisms. Three ablation variants of SPAformer were tested: (I) SPAformer V1: we use ProbSparse self-attention [55] to replace the SPA module in the encoder. (II) SPAformer V2: we use ProbSparse self-attention and cross-attention to replace SPA and patch cross-attention respectively. (III) SPAformer V3: we use Auto-Correlation [21] to replace the patch cross-attention mechanism in the decoder. The results are shown in Table 6. Our proposed method achieves SOTA results. And we find that the model and its variants that introduce the SPA module in the encoder are significantly better than the other models. For the cross-attention mechanism in the decoder, the performance of the V3 variant using Auto-Correlation is closer to our model. However, limited by the high complexity due to Time Delay Aggregation, the training of the V3 variant consumes more memory and takes longer.

5.5. Input and future sequence distribution experiments and analysis

In Section 5.2, we verify the effectiveness of SPAformer using MSE and MAE metrics. However, it may be incomplete to use only these metrics to demonstrate that our model outperforms the baselines. Therefore, in this section, we will consider using statistical significance to further compare the prediction accuracy of SPAformer with the baselines. We first introduce the Kolmogorov–Smirnov (KS) test method, and then conduct a comparative analysis based on the experimental results.

5.5.1. Kolmogorov–Smirnov test

The Kolmogorov–Smirnov (KS) test is a nonparametric statistical method used to evaluate the similarity between two probability distributions. This test does not require the data to follow a normal distribution. It is mainly used to evaluate the difference in the distribution form of two samples [61]. Essentially, the test answers the question : what is the probability that these two sets of samples come from the same (but unknown) probability distribution [15]. It quantifies the distance between the empirical distribution functions of the two samples. Unlike other tests of central tendency and dispersion, the KS test focuses on determining the difference in the overall shape of the distribution. The Kolmogorov–Smirnov statistic is:

$$D_{n,m} = \sup_x \left| F_{1,n}(x) - F_{2,m}(x) \right| \tag{14}$$

where $F_{1,n}$ and $F_{2,m}$ are the empirical distribution functions of the first and the second sample respectively, and sup is the supremum function. For large samples, the null hypothesis is rejected at level α if:

$$D_{n,m} > \sqrt{-\frac{1}{2}\ln\left(\frac{\alpha}{2}\right)} \cdot \sqrt{\frac{n+m}{n \cdot m}} \tag{15}$$

Table 7
Kolmogorov–Smirnov test P-values for long-term time series forecast output on the building energy consumption datasets. All models have a unified input context length of $120h$ (i.e. 5 days) and a forecast horizon of $\{120h, 240h, 360h, 480h, 600h, 720h, 840h\}$. The best results are in **bold**.

Models		SPAformer	PatchTST [16]	FEDformer [15]	Autoformer [21]	Informer [55]	Transformer [60]	True
Length	120	0.332	**0.338**	0.288	0.238	0.121	0.158	0.304
	240	**0.339**	0.278	0.272	0.211	0.022	0.127	0.314
	360	**0.349**	0.249	0.233	0.158	0.007	0.119	0.307
	480	**0.334**	0.193	0.225	0.149	0.018	0.101	0.298
	600	**0.325**	0.184	0.131	0.134	0.004	0.123	0.292
	720	**0.345**	0.179	0.090	0.139	0.005	0.126	0.294
	840	**0.347**	0.166	0.095	0.109	0.001	0.135	0.301

Table 8
Complexity analysis of different models. L is the length of the input sequence and N is the number of patches.

Models	SPAformer	PatchTST [16]	FEDformer [15]	Autoformer [21]	Informer [55]	Transformer [60]
Time	$\mathcal{O}(L \log L)$	$\mathcal{O}(N^2)$	$\mathcal{O}(L)$	$\mathcal{O}(L \log L)$	$\mathcal{O}(L \log L)$	$\mathcal{O}(L^2)$
Memory	$\mathcal{O}(L \log L)$	$\mathcal{O}(N^2)$	$\mathcal{O}(L)$	$\mathcal{O}(L \log L)$	$\mathcal{O}(L \log L)$	$\mathcal{O}(L^2)$

where n and m are the sizes of the first and second samples respectively.

5.5.2. P-values test results

This section uses the KS test to quantitatively evaluate the distribution similarity between the input and output sequences of different models. The null hypothesis is that the two samples come from the same distribution. The larger the P-value in the KS test, the less likely the null hypothesis is to be rejected, that is, the greater the probability that the input and predicted sequences come from the same distribution. We apply the KS test to the building energy consumption datasets, where the input sequence length is fixed to $120h$ and the output sequence is $O \in \{120h, 240h, 360h, 480h, 600h, 720h, 840h\}$.

The experimental results are shown in Table 7. The experiment sets the confidence level P-value to 0.01. Obviously, the SPAformer model significantly outperforms the baselines, especially for long-distance time series prediction. Furthermore, SPAformer achieved a larger P-value than the real output sequence, indicating better model performance rather than just more accurate predictions. It should be noted that the P-value of the Informer model is much lower than 0.01, which indicates that its predicted sequence is more likely to come from a different distribution than the input sequence.

6. Complexity analysis

The vanilla transformer has $\mathcal{O}(L^2)$ time complexity and memory usage due to the point-wise connection in self-attention [60]. Transformer variants reduce the theoretical complexity from $\mathcal{O}(L^2)$ to $\mathcal{O}(L)$ by introducing coefficient bias or exploring low-rank approximations of the self-attention matrix. However, it is not clear whether the actual inference time and memory cost on the device are improved. To fully evaluate the complexity of SPAformer, we first perform a theoretical analysis of the complexity. Then, we statistically analyze the actual time and memory efficiency of SPAformer and the baselines on the device.

6.1. Theoretical complexity analysis

The theoretical time and memory complexity of SPAformer and baselines are shown in Table 8. Informer [55] extends transformer attention with ProbSparse based on KL divergence, achieving $\mathcal{O}(L \log L)$ complexity, which selects dominant queries based on queries and key similarities. Autoformer [21] introduces an auto-correlation mechanism to replace self-attention, which discovers subsequence similarities based on sequence period and aggregates similar subsequences from the underlying period. This sequence mechanism achieves $\mathcal{O}(L \log L)$ complexity for series of length L and breaks the information utilization bottleneck by extending the point-wise representation aggregation to the subsequence level. FEDformer [15] applies attention operations in the frequency domain using Fourier transform and wavelet transform. It achieves linear complexity by randomly selecting a fixed-size subset of frequencies. PatchTST [16] reduces the complexity to $\mathcal{O}(N^2)$ by applying patches, which reduces L by a stride factor, where $N \approx \frac{L}{P}$.

Similar to PatchTST, assuming the input sequence length is L, SPAformer first performs a patch partitioning operation in the time-domain attention module. This operation reduces the complexity from $\mathcal{O}(L^2)$ to $\mathcal{O}(N^2)$, where N is much smaller than L. In the multi-scale frequency attention module, we first transform the time-domain sequence to the frequency-domain signal by fast Fourier transform (FFT), with a complexity of $\mathcal{O}(L \log L)$. Then, we divide the frequency-domain signal into low, medium and high frequencies, and use pattern indexing to reduce the complexity to $O(H)$, where $H = \frac{L}{2} + 1$. We perform convolution operations in different frequency intervals i (see Fig. 5), with a complexity of $O(H_i \cdot K^2)$, where K can be regarded as a constant as the size of the convolution kernel. The attention calculation complexity is $O(H_i^2)$, where H_i is much smaller than H. In summary, the overall time and memory complexity of SPAformer is $\mathcal{O}(L \log L)$.

6.2. Actual time and memory efficiency

The previous chapter mentioned that the transformer variant theoretically reduces the complexity from $\mathcal{O}(L^2)$ to $\mathcal{O}(L \log L)$ or even $\mathcal{O}(L)$. This seems like a significant improvement, but in practice, this theoretical improvement does not always translate into real efficiency gains. In order to fully understand the actual memory complexity and time complexity of SPAformer during training and inference, we conducted experiments on an NVIDIA GeForce RTX 3090 24 GB GPU device. We fix the input sequence length to $120h$, the output sequence length $O = \{120h, 240h, 360h, 480h, 600h, 720h, 840h\}$ and set the batch size to 32.

The experimental results are shown in Fig. 12. We compare the memory consumption of SPAformer with the baselines in (Fig. 12a). It is obvious that our model significantly outperforms other models, and the memory complexity does not increase significantly as the prediction length increases. In addition, our model adopts encoder–decoder architecture and direct prediction method. In contrast, PatchTST only uses the encoder for prediction, which loses some important information. It is worth noting that SPAformer has reached a level close to PatchTST in terms of memory complexity. In terms of time complexity, we compared the average training time of each epoch, as shown in (Fig. 12b). SPAformer outperforms all baselines when prediction length exceeds $360h$. After three trainings, our model converges after 60 iterations on average, while the average iterations of other baselines are 50. SPAformer takes about 15 s per epoch on average, and the total training time is about 900 s. In addition, due to the patch operation, the

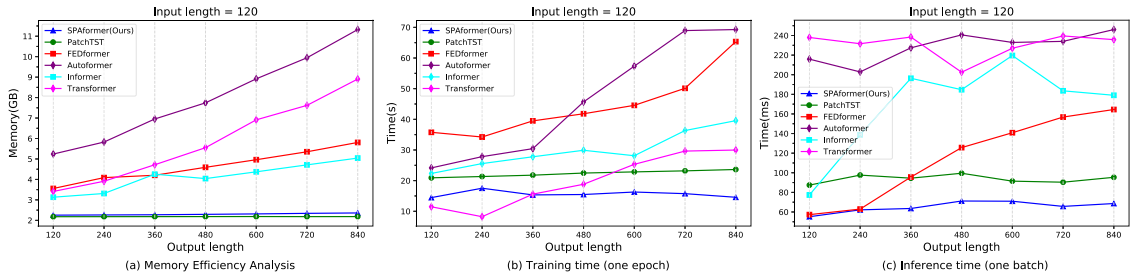

Fig. 12. Efficiency analysis. We fix the input sequence length to 120 and the prediction length to $O = \{120h, 240h, 360h, 480h, 600h, 720h, 840h\}$. (a) Memory usage comparison between SPAformer and the baselines. (b) Average training time per epoch. (c) Average inference time per batch of samples.

Table 9
Statistics of popular public datasets for benchmark.

Datasets	Weather	Electricity	ETTh1	ETTh2	ETTm1	ETTm2
Features	21	321	7	7	7	7
Timesteps	52 696	26 304	17 420	17 420	17 420	17 420

complexity of our model is not easily affected by the prediction length. Similarly, SPAformer also achieved state-of-the-art results in average inference time per batch of samples, as shown in (Fig. 12c). Overall, although it is not theoretically optimal for the complexity $\mathcal{O}(L \log L)$ of the SPAformer model, in practical applications, our model is state-of-the-art in terms of time efficiency and memory usage, which is crucial for long-distance time series forecasting tasks.

7. Experiments on public datasets

To further verify the generalization ability of SPAformer, we evaluated the model on 6 popular benchmark datasets, including electricity, weather, and ETT datasets [55]. In this section, we first introduced the basic information of the selected public datasets, including data sources, data scale, and characteristics. Then, we described the experimental settings and evaluation indicators in detail. By comparing the key indicators of the model on different datasets, the generalization ability of the model was comprehensively evaluated.

7.1. Public datasets

The following are six public datasets: (1) Four ETT datasets: ETT is a key indicator in the long-term deployment of electricity. This dataset contains the transformer load and oil temperature in two different counties in China from July 2016 to July 2018. The dataset is divided into hourly datasets (ETTh1, ETTh2) with a timestamp of 1 h and minute-level datasets (ETTm1, ETTm2) with a timestamp of 15 min. (2) The Electricity dataset contains hourly electricity consumption of 321 customers from 2012 to 2014. (3) The Weather dataset is data recorded every hour throughout 2020, which contains 21 meteorological indicators, such as temperature, humidity, wind speed, rainfall, etc. The statistical data of the above datasets are shown in Table 9.

7.2. Experimental settings

We follow the standards of the baselines and divide all datasets into training, validation, and test sets in chronological order. Among them, the ETT and Electricity datasets are divided into 12/4/4 months. The Weather dataset is divided into 6/3/3 months. For better comparison, all models follow the same experimental settings [21], where the input length $\boldsymbol{I}$ is fixed to 96, the prediction length is $\boldsymbol{T} \in \{96, 192, 336, 720\}$, and the Adam optimizer with an initial learning rate of 10^{-4} is used. The batch size is set to 32. In the patch operation of SPAformer, we fixed the patch-size and step-size to 16 and 8 respectively, referring to the parameter settings of PatchTST. MSE and MAE are used as the evaluation indicators of all models. All experiments are repeated 3 times, implemented in PyTorch, and performed on a single NVIDIA GeForce RTX 3090 24 GB GPU.

7.3. Comparison study

7.3.1. Baselines

Similar to Section 5.2, We select the latest state-of-the-art transformer-based models as our baselines, including FEDformer [15], PatchTST [16], Autoformer [21], Informer [55] and classic Transformer. All models follow the same experimental setup.

7.3.2. Results of multivariate time series forecasting

We conducted extensive experiments to verify the generalization ability of the SPAformer model for multivariate time series forecasting. The comparative experimental results are shown in Table 10. Overall, SPAformer reduces the MSE indicator by about 5% and the MAE indicator by about 4%. For the four ETT datasets and the Weather dataset, we can find that the prediction accuracy of SPAformer is significantly improved with the increase of the prediction length O. This means that SPAformer has good generalization ability and robustness for long-term prediction, which is meaningful for practical applications in the real world, such as weather forecasting and long-term energy consumption planning.

It should be noted that the prediction results of our model on the Electricity dataset are slightly inferior to those of FEDformer. As mentioned in Section 7.1, the dimension of the Electricity series data is 321. We believe that the method of multi-step sequence decomposition and then mapping to the latent space adopted by FEDformer may be more effective in capturing the relationship between variables. But this will increase the complexity to a certain extent. In contrast, SPAformer is more suitable for ultra-long-term prediction of strongly periodic time series data. We believe that a small decrease in accuracy is acceptable in exchange for predicting longer sequences.

8. Discussion

8.1. Stability analysis of the model

Stability is one of the important characteristics of the model, which can ensure that the model can provide reliable and consistent performance when facing different data and environmental conditions. In this section, we will explore the stability of SPAformer from three aspects: (I) computational efficiency, (II) prediction performance and (III) generalization ability.

8.1.1. Stability of computational efficiency

As mentioned in Section 6.2, in order to fully understand the actual memory complexity of SPAformer and the time complexity of training and inference, we conducted experiments on an NVIDIA GeForce RTX 3090 24 GB GPU device. We compared the memory consumption of SPAformer with the baselines, as shown in (Fig. 12a). SPAformer

Table 10
Multivariate long-range time series forecasting results for 6 public benchmark datasets with input context length $I = 96$ and forecast horizons $O \in \{96, 192, 336, 720\}$. The best results are in **bold**, and the second best are underlined.

Datasets		SPAformer		PatchTST [16]		FEDformer [15]		Autoformer [21]		Informer [55]		Transformer [60]	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.383	**0.400**	0.394	0.408	**0.376**	0.419	0.449	0.459	0.941	0.769	1.035	0.820
	192	0.436	**0.425**	0.446	0.438	**0.423**	0.448	0.500	0.482	1.007	0.792	1.152	0.864
	336	**0.452**	**0.433**	0.485	0.455	0.459	0.465	0.521	0.496	1.107	0.809	1.161	0.890
	720	**0.465**	**0.451**	0.495	0.474	0.506	0.507	0.515	0.517	1.181	0.865	1.250	0.925
ETTh2	96	**0.280**	0.343	0.294	**0.342**	0.346	0.388	0.358	0.397	3.755	1.525	2.212	1.221
	192	**0.353**	**0.380**	0.378	0.394	0.429	0.446	0.456	0.452	5.602	1.931	5.354	1.945
	336	**0.360**	**0.392**	0.382	0.410	0.496	0.487	0.482	0.486	4.721	1.835	4.076	1.413
	720	**0.382**	**0.410**	0.412	0.433	0.463	0.474	0.515	0.511	3.647	1.625	2.982	1.453
ETTm1	96	0.330	**0.348**	**0.324**	0.361	0.379	0.419	0.510	0.492	0.672	0.571	0.522	0.508
	192	0.368	**0.375**	**0.362**	0.383	0.426	0.441	0.553	0.496	0.795	0.669	0.748	0.648
	336	**0.385**	**0.395**	0.390	0.402	0.445	0.459	0.621	0.537	1.212	0.871	0.991	0.780
	720	**0.430**	**0.428**	0.461	0.438	0.543	0.490	0.671	0.561	1.166	0.845	1.099	0.819
ETTm2	96	0.179	**0.251**	**0.177**	0.260	0.203	0.287	0.255	0.339	0.365	0.462	0.426	0.493
	192	**0.240**	**0.286**	0.248	0.306	0.269	0.328	0.281	0.340	0.533	0.586	0.869	0.685
	336	**0.282**	**0.334**	0.304	0.342	0.325	0.366	0.339	0.379	1.363	0.887	1.213	0.838
	720	**0.371**	**0.388**	0.403	0.397	0.421	0.415	0.422	0.419	3.379	1.338	2.971	1.245
Electricity	96	**0.165**	**0.235**	0.180	0.264	0.186	0.302	0.196	0.313	0.304	0.393	0.533	0.489
	192	**0.181**	**0.261**	0.188	0.275	0.197	0.311	0.211	0.324	0.327	0.417	0.547	0.498
	336	0.211	0.301	**0.206**	**0.291**	0.213	0.328	0.214	0.327	0.333	0.422	0.570	0.499
	720	0.261	0.332	0.247	**0.328**	**0.233**	0.344	0.236	0.342	0.351	0.427	0.625	0.510
Weather	96	0.180	0.220	**0.177**	**0.218**	0.238	0.314	0.249	0.329	0.300	0.384	0.612	0.493
	192	0.232	**0.240**	**0.224**	0.258	0.275	0.329	0.325	0.370	0.598	0.544	0.724	0.638
	336	**0.246**	**0.280**	0.277	0.297	0.339	0.377	0.351	0.391	0.702	0.620	0.956	0.713
	720	**0.330**	**0.332**	0.350	0.345	0.389	0.409	0.415	0.426	1.059	0.731	1.437	0.858

significantly outperforms other models, and the memory consumption is relatively stable as the prediction length increases. Unlike baselines such as FEDformer and Autoformer, which require more encoder–decoder layers, SPAformer only needs 1 to 2 encoder and decoder layers to achieve optimal prediction performance, as described in Section 5.3.3. This contributes to the stability of memory consumption during model training, which is helpful for longer time series prediction. In Section 6.2, we also compared the actual time complexity of the model. SPAformer's average training time per epoch exceeds all baselines. On the other hand, our model also achieves state-of-the-art results in average inference time per batch of samples, which is mainly due to the reduced complexity of the patch operation we introduced.

8.1.2. Stability of prediction performance

In Section 5.3, we conducted three aspects of sensitivity analysis: (I) the performance of the model under different input lengths, (II) the impact of different patch lengths on model performance and (III) The impact of transformer hyper-parameter settings on model prediction stability. **In the short-term (360h)** energy consumption prediction task, our model performance is close to PatchTST and exceeds all other models. This shows that dividing the time series into patches can capture richer local information. **In the long-distance (720h)** prediction task, most baselines first decrease and then increase the MSE as the input length increases. They are unable to capture more complex periodic patterns from longer time series. In contrast, SPAformer shows stable prediction performance, which benefits from the multi-scale frequency attention mechanism and enhances the ability to capture long periods. **Patch operation**, as a key module of SPAformer, is studied for the impact of different lengths on model stability. The experimental results are shown in Fig. 10. There is no significant difference in the MSE score as the patch size increases. This shows that SPAformer has good stability and robustness to the patch length hyper-parameter. On the other hand, **in the hyper-parameter sensitivity analysis** in Section 5.3.3, we set 12 hyper-parameter combinations for the number of encoder and decoder layers, latent space dimensions, and the number of attention mechanism heads. When the prediction field of view is 120 and 360, SPAformer shows robustness to the hyper-parameter selection.

8.1.3. Generalization ability of the model

To verify the generalization ability of the SPAformer model, we evaluate the performance of the model in multivariate time series forecasting tasks on 6 popular benchmark datasets in Section 7.3. Overall, SPAformer reduces the MSE indicator by about 5% and the MAE indicator by about 4%. As the prediction length O increases, the prediction accuracy of SPAformer is significantly better than the baselines. This means that SPAformer has good generalization ability and robustness for long-term predictions, which is meaningful for practical applications in the real world, such as weather forecasting and long-term energy consumption planning.

8.2. Limitations of the model

While we are deeply discussing the advantages and application potential of the SPAformer model, we also need to face up to some limitations in the design and application of SPAformer. In this section, we will discuss three aspects: (I) The model may face the risk of overfitting, (II) The model may be more suitable for processing time series data with strong periodicity and (III) The model currently does not support online real-time training.

8.2.1. Overfitting problem

As the networks of Transformer variants become more complex, they often suffer from model overfitting. Our model will inevitably face this problem. As described in Section 5.2.3, SPAformer achieved state-of-the-art results in short-term prediction of univariate prediction of total building energy consumption. However, as the prediction length increases, Informer, which has a simpler structure, outperforms all models. The reason is that the complex encoder–decoder architecture used in these models leads to training overfitting. On the other hand, too many network layers can also lead to model overfitting. In Section 5.3.3, we compared the impact of different numbers of encoders and decoders on the MSE score of SPAformer. As shown in Fig. 11, in the experiment with an output length of 360, the MSE increases slightly when the number of encoders and decoders increases from 1 to 3. We believe that overfitting may have occurred when the number of layers is greater than or equal to 3. In order to minimize overfitting, we fixed the number of encoders and decoders of SPAformer to 1, and adopted regularization and early stopping techniques.

8.2.2. Suitable for strongly periodic time series data

In the field of large commercial office buildings, building energy consumption often shows obvious periodicity, including short periods of days and weeks and long periods of quarters and years. SPAformer is proposed to address the strong periodicity of building energy consumption. The experimental results in Section 5.2.2 show that the model is significantly better than other models in capturing periodicity. This is due to the introduction of the period-trend decomposition block and the SPA module. At the same time, the encoder–decoder structure is used in the prediction of the periodic component, which uses a more complex network model to unravel the multiple periodic patterns of building energy consumption at multi-scale resolutions. However, our model may not be suitable for weak-periodic multivariate time series forecasting tasks.

8.2.3. Incremental data cannot be trained in real time

In the field of building energy management, real-time energy consumption prediction and automated decision-making are essential for improving energy efficiency, reducing costs, and achieving sustainable development. However, SPAformer currently lacks the ability of real-time prediction and online real-time training, which limits its ability to perform effective energy consumption management and automated decision-making in a dynamically changing environment. Due to the inability to train online in real time, the model may need to be updated offline regularly, which is not only time-consuming, but may also lead to a decrease in the model's predictive ability during the update period. In order to overcome these limitations, it may be necessary to develop new algorithms, improve the model's learning ability, and optimize the data processing and prediction mechanism.

8.3. Application scenarios of SPAformer

Energy consumption forecasting is an important part of the energy management system, which aims to provide daily management of power utilities, grid planning, and make the best decisions in grid energy management to ensure the safe operation of the power system [62, 63]. The SPAformer prediction model we designed achieves high-precision and long-distance predictions. This helps decision makers formulate and implement energy-saving policies, reduce building energy consumption, and achieve sustainable development. For example, building energy management systems often uniformly manage indoor temperatures by setting constant temperatures. However, the actual outdoor temperature will continue to change over time, so dynamically adjusting the set temperature is crucial for energy saving. Our model achieves high-resolution and accurate prediction of multi-category energy consumption of buildings, which can assist building managers to set temperature thresholds more reasonably and continuously adjust energy supply strategies.

Of course, model deployment is also a challenge. Currently, SPAformer is only a non-online prediction model, which cannot train incremental data in real time. This limits its ability to effectively manage energy consumption and make automated decisions in a dynamically changing environment. In the next stage of work, we plan to jointly deploy SPAformer with AI Agents to optimize the operation strategy and automate the control of the building heating, ventilation and air conditioning system (HVAC). Specifically, we first use SPAformer to predict the future HVAC power consumption based on historical energy consumption. Then feed the prediction results and the real-time parameters collected by the HVAC sensors into the pre-trained large language model (LLM). Based on the power system parameter adjustment strategy given by the LLM, the administrator manually or the system automatically adjusts the parameters. Based on the above technical route, we can develop API interfaces through the open source framework of AI Agents (such as AutoGen, LangChain, etc.), which has the ability to integrate multiple tools, including predictive models, large language models, optimization algorithm libraries, and simulation. We believe that SPAforemer's high-precision prediction and LLM's natural language processing capabilities can help dispatchers quickly learn dispatching knowledge, improve dispatching efficiency and accuracy, and thus achieve economic, safe and low-carbon operation of the power system.

9. Conclusions and future work

This paper studies the problem of long-term series forecasting in the field of building energy consumption, which is important for improving energy efficiency. Complex long and short cycles and trend patterns in energy consumption data prevent models from learning effective dependencies. In this paper, we propose a multi-scale Spectra-Patch Attention mechanism for multivariate long-term series forecasting of building energy consumption. Specifically, we propose to use the MLP and the attention mechanism to model trend and periodicity, respectively, after the decomposition of the periodic trend. In periodic modeling, we propose an attention mechanism that fuses multi-scale frequency domain and patch time-domain signals. The frequency attention in this mechanism focuses on modeling periodicity, while the patch temporal attention focuses on modeling local dependencies. In addition, we conducted a multi-dimensional comparison with five other deep learning models based on real energy consumption data of the typical large commercial office building in Beijing, China. SPAformer achieves state-of-the-art performance on long-term energy consumption series forecasting benchmarks. Moreover, comparative experimental results on 6 public benchmark datasets also verify that the model has a strong generalization ability.

The model proposed in this study has obvious advantages in predicting strongly periodic time series data, especially building energy consumption data. In addition, our model also provides a reference for the improvement of attention mechanisms based on time-frequency fusion. However, our study only considered different categories of building energy consumption data and did not consider the impact of external factors on energy consumption. Energy consumption can be affected by various factors, such as environmental factors (temperature, humidity, etc.), human factors (personnel activities, etc.), equipment factors (set temperatures and valve openings of the VAV system, etc.), spatial factors (glazing orientation, floor space, building materials, etc.) and the relevance of the building complex, etc. Therefore, we can further research from the following aspects in the future: (I) Propose a more complex and effective time-frequency fusion attention mechanism. (II) Explore the correlation between different external factors and building energy consumption to further improve the accuracy and generalization ability of the model. (III) Explore possible correlations between different buildings based on federated learning.

CRediT authorship contribution statement

Chao-fan Wang: Writing – original draft, Visualization, Validation, Software, Data curation. **Kui-xing Liu:** Data curation, Resources, Validation. **Jieyang Peng:** Writing – review & editing, Validation, Investigation. **Xiang Li:** Writing – review & editing, Methodology. **Xiu-feng Liu:** Writing – review & editing, Supervision. **Jia-wan Zhang:** Supervision, Visualization. **Zhi-bin Niu:** Writing – review & editing, Validation, Supervision, Methodology.

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.

Appendix. Terminology list

1. MTS: multivariate time-series
2. RNN: recurrent neural network
3. LSTM: long short-term memory network
4. MLP: multilayer perceptron
5. CNN: convolutional neural networks
6. ARIMA: autoregressive integrated moving average
7. VARMA: vector autoregressive moving average
8. SVR: support vector regression
9. VAR: vector autoregressive
10. GNN: graph neural networks
11. STGNN: spatio-temporal graph neural network
12. EMD: empirical mode decomposition
13. DWT: wavelet transform
14. SPA: spectra-patch attention
15. GESD: generalized extremization deviation
16. DFT: discrete Fourier transform
17. FFT: Fast Fourier transform
18. MSE: Mean Squared Error
19. MAE: Mean Absolute Error
20. KS: Kolmogorov–Smirnov

Data availability

The authors do not have permission to share data.

References

[1] Zhou X, Huang Z, Scheuer B, Lu W, Zhou G, Liu Y. High-resolution spatial assessment of the zero energy potential of buildings with photovoltaic systems at the city level. Sustainable Cities Soc 2023;93:104526.

[2] Zhou X, Huang Z, Scheuer B, Wang H, Zhou G, Liu Y. High-resolution estimation of building energy consumption at the city level. Energy 2023;275:127476.

[3] Meng F, Lu Z, Li X, Han W, Peng J, Liu X, et al. Demand-side energy management reimagined: A comprehensive literature analysis leveraging large language models. Energy 2024;291:130303.

[4] Dong H, Zhu J, Li S, Wu W, Zhu H, Fan J. Short-term residential household reactive power forecasting considering active power demand via deep transformer sequence-to-sequence networks. Appl Energy 2023;329:120281.

[5] Ahmad T, Chen H, Huang R, Yabin G, Wang J, Shair J, et al. Supervised based machine learning models for short, medium and long-term energy prediction in distinct building environment. Energy 2018;158:17–32.

[6] Fang L, He B. A deep learning framework using multi-feature fusion recurrent neural networks for energy consumption forecasting. Appl Energy 2023;348:121563.

[7] Andersen FM, Baldini M, Hansen LG, Jensen CL. Households' hourly electricity consumption and peak demand in Denmark. Appl Energy 2017;208:607–19.

[8] Khalil M, McGough AS, Pourmirza Z, Pazhoohesh M, Walker S. Machine learning, deep learning and statistical analysis for forecasting building energy consumption—A systematic review. Eng Appl Artif Intell 2022;115:105287.

[9] Ramos PVB, Villela SM, Silva WN, Dias BH. Residential energy consumption forecasting using deep learning models. Appl Energy 2023;350:121705.

[10] Gao Y, Ruan Y. Interpretable deep learning model for building energy consumption prediction based on attention mechanism. Energy Build 2021;252:111379.

[11] Chen S, Zhang G, Xia X, Setunge S, Shi L. A review of internal and external influencing factors on energy efficiency design of buildings. Energy Build 2020;216:109944.

[12] Woo G, Liu C, Sahoo D, Kumar A, Hoi S. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. 2022, arXiv preprint arXiv:2202.01575.

[13] Huang L, Zou F, Gan Z. Short term prediction of colleges building itemized energy consumption based on long short-term memory neural network. In: 2020 IEEE 3rd international conference of safe production and informatization. IICSPI, IEEE; 2020, p. 641–5.

[14] Li L, Su X, Bi X, Lu Y, Sun X. A novel transformer-based network forecasting method for building cooling loads. Energy Build 2023;296:113409.

[15] Zhou T, Ma Z, Wen Q, Wang X, Sun L, Jin R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International conference on machine learning. PMLR; 2022, p. 27268–86.

[16] Nie Y, Nguyen NH, Sinthong P, Kalagnanam J. A time series is worth 64 words: Long-term forecasting with transformers. 2022, arXiv preprint arXiv:2211.14730.

[17] Brockwell PJ, Davis RA. Introduction to time series and forecasting. Springer; 2002.

[18] Deb C, Zhang F, Yang J, Lee SE, Shah KW. A review on time series forecasting techniques for building energy consumption. Renew Sustain Energy Rev 2017;74:902–24.

[19] Liu D, Yang Q, Yang F. Predicting building energy consumption by time series model based on machine learning and empirical mode decomposition. In: 2020 5th IEEE international conference on big data analytics. ICBDA, IEEE; 2020, p. 145–50.

[20] Zhou C, Chen X. Predicting energy consumption: A multiple decomposition-ensemble approach. Energy 2019;189:116045.

[21] Wu H, Xu J, Wang J, Long M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Adv Neural Inf Process Syst 2021;34:22419–30.

[22] Lin Y, Koprinska I, Rana M. SSDNet: State space decomposition neural network for time series forecasting. In: 2021 IEEE international conference on data mining. ICDM, IEEE; 2021, p. 370–8.

[23] Zhang X, Jin X, Gopalswamy K, Gupta G, Park Y, Shi X, et al. First de-trend then attend: Rethinking attention for time-series forecasting. 2022, arXiv preprint arXiv:2212.08151.

[24] Chen S, Li N, Yoshino H, Guan J, Levine MD. Statistical analyses on winter energy consumption characteristics of residential buildings in some cities of China. Energy Build 2011;43(5):1063–70.

[25] Melo FC, da Graça GC, Panão MJO. A review of annual, monthly, and hourly electricity use in buildings. Energy Build 2023;113201.

[26] Huang WZ, Zaheeruddin M, Cho S. Dynamic simulation of energy management control functions for HVAC systems in buildings. Energy Convers Manag 2006;47(7–8):926–43.

[27] Savić S, Selakov A, Milošević D. Cold and warm air temperature spells during the winter and summer seasons and their impact on energy consumption in urban areas. Nat Hazards 2014;73:373–87.

[28] Ma H, Du N, Yu S, Lu W, Zhang Z, Deng N, et al. Analysis of typical public building energy consumption in northern China. Energy Build 2017;136:139–50.

[29] Rahman A, Srikumar V, Smith AD. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. Appl Energy 2018;212:372–85.

[30] Peng J, Kimmig A, Wang D, Niu Z, Liu X, Tao X, et al. Energy consumption forecasting based on spatio-temporal behavioral analysis for demand-side management. Appl Energy 2024;374:124027.

[31] Li C, Tang M, Zhang G, Wang R, Tian C. A hybrid short-term building electrical load forecasting model combining the periodic pattern, fuzzy system, and wavelet transform. Int J Fuzzy Syst 2020;22:156–71.

[32] Yan J, Hu L, Zhen Z, Wang F, Qiu G, Li Y, et al. Frequency-domain decomposition and deep learning based solar PV power ultra-short-term forecasting model. IEEE Trans Ind Appl 2021;57(4):3282–95.

[33] Woo G, Liu C, Sahoo D, Kumar A, Hoi S. Etsformer: Exponential smoothing transformers for time-series forecasting. 2022, arXiv preprint arXiv:2202.01381.

[34] Zhang J, Wei Y-M, Li D, Tan Z, Zhou J. Short term electricity load forecasting using a hybrid model. Energy 2018;158:774–81.

[35] Mounir N, Ouadi H, Jrhilifa I. Short-term electric load forecasting using an EMD-BI-LSTM approach for smart grid energy management system. Energy Build 2023;288:113022.

[36] Kuster C, Rezgui Y, Mourshed M. Electrical load forecasting models: A critical systematic review. Sustain Cities Soc 2017;35:257–70.

[37] Wang Z, Srinivasan RS. A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models. Renew Sustain Energy Rev 2017;75:796–808.

[38] Yuan C, Liu S, Fang Z. Comparison of China's primary energy consumption forecasting by using ARIMA (the autoregressive integrated moving average) model and GM (1, 1) model. Energy 2016;100:384–90.

[39] Barak S, Sadegh SS. Forecasting energy consumption using ensemble ARIMA–ANFIS hybrid algorithm. Int J Electr Power Energy Syst 2016;82:92–104.

[40] Guefano S, Tamba JG, Azong TEW, Monkam L. Methodology for forecasting electricity consumption by Grey and vector autoregressive models. MethodsX 2021;8:101296.

[41] Chen S, Wang X, Harris CJ. NARX-based nonlinear system identification using orthogonal least squares basis hunting. IEEE Trans Control Syst Technol 2007;16(1):78–84.

[42] Bouchachia A, Bouchachia S. Ensemble learning for time series prediction. 2008.

[43] Frigola R, Rasmussen CE. Integrated pre-processing for Bayesian nonlinear system identification with Gaussian processes. In: 52nd IEEE conference on decision and control. IEEE; 2013, p. 5371–6.

[44] Jordan MI. Serial order: A parallel distributed processing approach. In: Advances in psychology. vol. 121, Elsevier; 1997, p. 471–95.

[45] Kim T-Y, Cho S-B. Predicting residential energy consumption using CNN-LSTM neural networks. Energy 2019;182:72–81.

[46] Chung J, Jang B. Accurate prediction of electricity consumption using a hybrid CNN-LSTM model based on multivariable data. PLoS One 2022;17(11):e0278071.

[47] Li Y, Yu R, Shahabi C, Liu Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. 2017, arXiv preprint arXiv:1707.01926.
[48] Peng J, Kimmig A, Wang J, Liu X, Niu Z, Ovtcharova J. Dual-stage attention-based long-short-term memory neural networks for energy demand prediction. Energy Build 2021;249:111211.
[49] Verdone A, Scardapane S, Panella M. Multi-site forecasting of energy time series with spatio-temporal graph neural networks. In: 2022 international joint conference on neural networks. IJCNN, IEEE; 2022, p. 1–8.
[50] Hu Y, Cheng X, Wang S, Chen J, Zhao T, Dai E. Times series forecasting for urban building energy consumption based on graph convolutional network. Appl Energy 2022;307:118231.
[51] Duan Z, Xu H, Huang Y, Feng J, Wang Y. Multivariate time series forecasting with transfer entropy graph. Tsinghua Sci Technol 2022;28(1):141–9.
[52] Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018, arXiv preprint arXiv:1810.04805.
[53] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16x16 words: Transformers for image recognition at scale. 2020, arXiv preprint arXiv:2010.11929.
[54] Dong L, Xu S, Xu B. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In: 2018 IEEE international conference on acoustics, speech and signal processing. ICASSP, IEEE; 2018, p. 5884–8.
[55] Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI conference on artificial intelligence. Vol. 35, 2021, p. 11106–15.
[56] Du D, Su B, Wei Z. Preformer: predictive transformer with multi-scale segment-wise correlations for long-term time series forecasting. In: ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing. ICASSP, IEEE; 2023, p. 1–5.
[57] Zerveas G, Jayaraman S, Patel D, Bhamidipaty A, Eickhoff C. A transformer-based framework for multivariate time series representation learning. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021, p. 2114–24.
[58] Liu S, Yu H, Liao C, Li J, Lin W, Liu AX, et al. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In: International conference on learning representations. 2021.
[59] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 770–8.
[60] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst 2017;30.
[61] Alabbasi A, Khalil M, McGrail T. Integrating NARX neural network with KS test for accurate partial discharge detection in transformers. In: 2023 18th conference on electrical machines, drives and power systems. ELMA, IEEE; 2023, p. 1–7.
[62] Somu N, Raman MG, Ramamritham K. A hybrid model for building energy consumption forecasting using long short term memory networks. Appl Energy 2020;261:114131.
[63] He F, Zhou J, Feng Z-k, Liu G, Yang Y. A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm. Appl Energy 2019;237:103–16.