Information Fusion 125 (2026) 103454

Contents lists available at ScienceDirect

Information Fusion

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

Full length article

Multi-Scale Data Fusion and AdaptiveLoss Kolmogorov–Arnold Network for multivariate time series forecasting

Jian Liu [a], Fan Yang [b], Ke Yan [a,*]

[a] Department of Mechanical and Electrical Engineering, Hunan University, Changsha, 410082, China
[b] School of Software Technology, Zhejiang University, Hangzhou, 310027, China

ARTICLE INFO

Dataset link: https://github.com/liujian123223/MDFM-AdaKAN

Keywords:
Time series forecasting
Multi-scale data fusion
Kolmogorov–Arnold network
Artificial intelligence

ABSTRACT

Real-world multivariate time series often exhibit multiple interwoven and highly coupled periodic patterns, along with significant volatility and uncertainty, which present substantial challenges for accurate time series forecasting. Inspired by the concept of multi-scale data fusion and the Kolmogorov–Arnold theory, this study proposes a novel time series forecasting framework that achieves both high predictive accuracy and parameter efficiency. The proposed approach comprises two key modules: the Multi-Scale Data Fusion Model (MDFM) and the AdaptiveLoss Kolmogorov–Arnold Network (AdaKAN). MDFM implements a two-stage fusion process that prioritizes trend and seasonal information across different temporal scales, thereby enhancing the model's ability to capture both long-term trends and fine-grained fluctuations. AdaKAN leverages Gaussian radial basis functions and adaptive loss functions, significantly improving both the computational efficiency and predictive performance of the algorithm compared to the traditional Kolmogorov–Arnold Network. The experimental results indicate that, compared with recently published cutting-edge methods, MDFM-AdaKAN consistently demonstrated superior accuracy and adaptability. Specifically, it achieved a 15.2% improvement in the solar irradiance prediction task and a 19.4% improvement in the electricity transformer temperature prediction task, providing a reliable solution for high-precision time-series predictions.

1. Introduction

Time series forecasting is widely applied in various fields such as energy consumption [1–4], photovoltaics [5–7], transportation [8], and economics [9]. It has gradually become an indispensable component in planning and decision-making tasks [10,11]. Effective time series forecasting algorithms contribute to the optimization of enterprise production and the assistance of governments in scientific resource allocation planning, thereby improving energy efficiency, reducing carbon emissions, etc. [12]. Multivariate time series (MTS) data is among the most commonly used inputs in forecasting tasks. It consists of sequentially ordered observations and is characterized by large scale, high dimensionality, and significant volatility [13,14]. Unlike other data structures, MTS cannot obtain sufficient semantic information by analyzing individual time points. Instead, it requires integrating historical data over a certain period to capture overall trends and seasonal fluctuations within the time series. Therefore, enhancing a model's ability to understand the semantic information contained in historical data is key to improving forecasting accuracy.

Extracting time-series features from multiple temporal scales is an effective approach to enhancing a model's understanding of historical data semantics [15,16]. In real-world scenarios, MTS often exhibits multiple intertwined and highly coupled periodic patterns [17]. For example, in the case of solar irradiance data, it exhibits multiple periodic patterns due to the combined effects of the alternation of day and night as well as seasonal variations in daylight duration. Similarly, electricity transformer temperature data exhibit multiple periodic patterns due to electricity load fluctuations caused by human activity, as well as variations in ambient temperature. These complex periodic patterns make it difficult for single-temporal-scale-based time series forecasting methods to achieve high-precision predictions. However, existing mainstream methods often overlook the diverse semantic information embedded in different periodic patterns and instead tend to construct more complex model architectures or introduce higher-dimensional input features to address this challenge.

This strategy presents two major limitations. First, excessively complex model architectures significantly increase computational costs and are prone to severe overfitting, undermining the model's generalization capability. When multi-step forecasting strategies are employed, model complexity and predictive uncertainty are further amplified, leading to rapid error accumulation. Second, most existing mainstream model

* Corresponding author.
E-mail address: keyan@hnu.edu.cn (K. Yan).

https://doi.org/10.1016/j.inffus.2025.103454
Received 24 March 2025; Received in revised form 5 June 2025; Accepted 21 June 2025
Available online 5 July 2025

Nomenclature	
Abbreviation or symbol	Definition
MDFM	Multi-scale data fusion model
AdaptiveLoss	Adaptive loss function
MSELoss	Mean squared error loss function
KAN	Kolmogorov-Arnold network
AdaKAN	AdaptiveLoss Kolmogorov-Arnold network
GRBF	Gaussian radial basis function
MTS	Multi-variate time series
UAT	Universal approximation theorem
MLP	Multilayer perceptron
CNN	Convolutional neural network
LSTM	Long short-term memory
GRU	Gated recurrent unit
GPU	Graphics processing unit
DFT	Discrete Fourier transform
FTC	Fine-to-coarse
CTF	Coarse-to-fine
Z-score	Standard score
AvgPool	Average pooling
Proj	Projection
IDFT	Inverse discrete Fourier transform
DC	Direct current
GELU	Gaussian error linear unit
RBFs	Radial basis functions
GMC	Geman–McClure
NWP	Numerical weather prediction
GHI	Global horizontal irradiance
MAE	Mean absolute error
MSE	Mean squared error
nRMSE	Normalized root mean squared error
R^2	Coefficient of determination

architectures rely heavily on the Universal Approximation Theorem (UAT) for time series modeling. While UAT allows neural networks to approximate any continuous function with arbitrary precision, it does not account for the inherent structural characteristics of time series data. As a result, these models struggle to automatically determine the optimal network size in practical applications and often require a large number of non-embedded parameters [18], which further increases computational cost and diminishes model interpretability [19]. Therefore, effectively modeling the multi-scale periodic patterns in time series while reducing dependence on the UAT remains a critical challenge.

The Kolmogorov-Arnold Representation Theorem [20,21] provides a solid mathematical foundation for achieving higher forecasting accuracy and interpretability with fewer parameters, offering an effective alternative to UAT-based approaches [22]. Currently, Kolmogorov-Arnold Networks (KAN) [23] have been successfully applied in fields such as image processing [24]. However, due to their high computational cost, their potential in large-scale time series forecasting tasks has not been fully explored.

This study proposes a hybrid KAN model architecture that integrates the MDFM with the AdaKAN to address the limitations of existing approaches. Specifically, the MDFM adopts a two-stage fusion strategy that incorporates both different-scale and equal-scale fusion mechanisms. In the different-scale fusion stage, two opposite fusion pathways are designed for the seasonal and trend components based on their distinct temporal characteristics, progressively integrating local details and global trend information across scales. This design enhances the model's capability to capture latent temporal patterns across multiple temporal scales, effectively addressing the limitations of existing methods in multi-scale time series feature extraction.

The AdaKAN module replaces the third-order B-spline functions used in traditional KAN with Gaussian radial basis functions and incorporates a layer normalization mechanism to regulate the input distribution. This design effectively addresses the high computational cost and limited scalability encountered by existing KAN-based methods when applied to large-scale datasets. To the best of our knowledge, MDFM-AdaKAN is the first KAN-based method deployed in practical forecasting tasks involving real-world, high-dimensional time series. In addition, AdaKAN is trained using an adaptive loss function (AdaptiveLoss) that dynamically adjusts its learnable parameters in response to the data distribution during training. This adaptability enhances the model's robustness to outliers and improves the stability of predictions under varying data conditions. This approach effectively mitigates the limited generalization capability observed in most existing models when dealing with multi-condition and multi-distribution data.

The proposed model provides an effective and generalizable solution that achieves high accuracy and exhibits robustness in multivariate time series forecasting. Its superior performance in solar irradiance and electricity transformer temperature prediction demonstrates its broad applicability to efficient energy scheduling, dynamic load management, and operational safety enhancement in smart grids. The method exhibits strong cross-domain scalability and offers robust technical support for time series forecasting under diverse operational conditions and unexpected environmental disturbances, highlighting its significant potential for engineering applications and real-world deployments.

2. Related works

Several methods have been developed to tackle the challenges of MTS forecasting, which can generally be categorized into statistical models, physical models, machine learning models, and deep learning models [25]. Statistical models primarily leverage mathematical formulas to analyze the statistical patterns in historical data for prediction. These models are computationally efficient and can offer reasonable predictions with limited data. However, they encounter limitations when dealing with nonlinear and non-stationary data [26,27]. Physical models, which rely on physical principles to model time series, tend to exhibit high accuracy and interpretability. Nevertheless, these models are computationally intensive, require substantial prior knowledge, and lack robust generalization capabilities [28]. Traditional machine learning models, such as support vector machines [29] and random forests [30], typically rely on manual feature extraction and struggle to model complex nonlinear relationships. Consequently, they face challenges in handling high-dimensional, complex, nonlinear, and large-scale datasets [31].

With the development of deep learning technologies, deep neural networks have been extensively applied to MTS forecasting tasks. Depending on the model architecture, deep learning models can be categorized into Transformer-based models [18], CNN-based models [32], and MLP-based models [33]. Transformers have been introduced to the time series forecasting domain due to their ability to effectively capture long-term dependencies. Given the high computational complexity inherent in Transformers, most Transformer-based time series forecasting research has focused on model simplification strategies, such as using low-rank matrices to replace traditional fully connected self-attention matrices [34], employing local attention mechanisms instead of global self-attention for computation, and adopting sparse self-attention mechanisms to reduce model complexity [35]. These studies strike an effective balance between computational efficiency and accuracy, making them a viable method for long-term forecasting.

CNN-based methods excel at capturing local time series patterns and automatic feature extraction but struggle with extracting long-term temporal features [36]. CNN-based approaches often incorporate multi-scale convolutions or integrate with GRU or LSTM models to simultaneously capture both short- and long-term time series dynamics. For instance, Qin et al. [37] proposed a CNN-GRU-Attention network

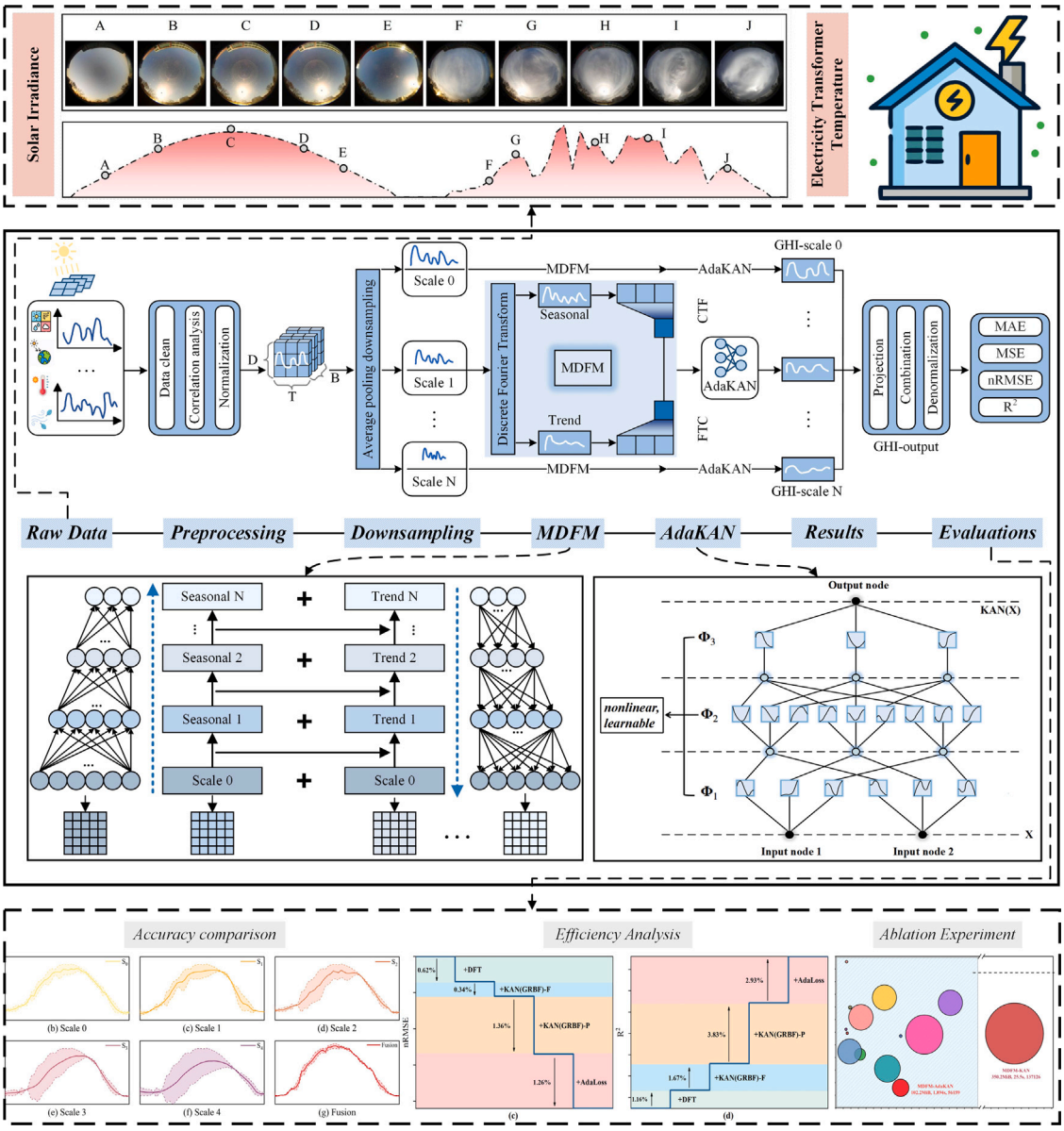

Fig. 1. The overall flowchart of the proposed forecasting framework.

framework for short-term forecasting, which effectively predicted electricity load. Rick et al. [38] developed a time series forecasting method based on LSTM and CNN, which achieved low prediction error on a real-world energy consumption dataset.

The multi-layer perceptron (MLP) is widely employed in time series forecasting due to its simple structure and efficient computational performance. MLP performs nonlinear mapping of input data through multiple hidden layers, enabling it to capture the complex relationships between features and target variables. However, MLP exhibits weak performance in capturing long-term dependencies or handling complex time series patterns. Consequently, MLP-based time series forecasting methods are often combined with other models or signal decomposition and feature fusion strategies [39,40].

The integration of time series forecasting models with multi-scale feature extraction techniques has garnered increasing attention, as it enhances the model's ability to understand semantic information in historical data and improve prediction accuracy. Guo et al. [15] utilized a multi-scale attention mechanism to dynamically select the most relevant information across different scales, employing both input attention and temporal attention for prediction. Zhang et al. [16] introduced a pyramid attention structure based on multi-scale feature extraction, which effectively captured long- and short-term dependencies and enhanced the model's prediction accuracy. Chen et al. [41] proposed a graph neural network model that combines multi-scale temporal feature extraction and attention mechanisms. By extracting features at multiple temporal scales and employing attention mechanisms for weighted fusion, the model's capability to model complex time series data was significantly improved. Wang et al. [39] proposed a multi-scale hybrid architecture that integrates past decomposable mixture and future multi-predictor mixture models. They demonstrated that even a simple MLP architecture could achieve significant improvements in prediction performance when coupled with a well-designed multi-scale feature fusion approach. These studies underscore the considerable advantages of multi-scale feature extraction in time series forecasting, as it effectively captures information across various temporal scales and enhances prediction accuracy when integrated with other models.

Recently, several KAN-based models have achieved notable improvements in time series forecasting by modifying their basis functions or combining them with other models [19,42,43]. [44] integrated convolutional layers with KAN for multi-step time series forecasting. [45]

proposed a convolutional autocorrelation method to replace the traditional self-attention mechanism and utilized KAN to replace MLP, enabling more accurate capture and representation of complex relationships in high-dimensional data. However, KAN's low GPU computational efficiency and susceptibility to dimensionality explosion limit its current applicability to small public datasets, and it struggles to achieve optimal prediction performance in large-scale data scenarios under complex conditions. Consequently, future research should focus on enhancing KAN's computational efficiency and improving its applicability across various complex environments. Furthermore, exploring effective integration strategies between KAN-based models and other data processing techniques, such as multi-scale feature extraction, will be crucial for further improving prediction accuracy and robustness. The main contributions of the proposed MDFM-AdaKAN framework are as follows:

(1) **A novel two-stage multi-scale data fusion module for MTS forecasting.** A two-stage data fusion strategy is proposed for capturing both short-term fluctuations and long-term trends, significantly enhancing overall time series forecasting performance. The two-stage data fusion includes: (1) different-scale fusion, by applying opposing data fusion strategies on the seasonal and trend components at different temporal scales; and (2) equal-scale fusion, by integrating features extracted from both seasonal and trend components within the equal scale, ensuring a comprehensive fusion of multi-scale time series data.

(2) **A refined AdaKAN algorithm on top of the Kolmogorov-Arnold theory.** The refined AdaKAN algorithm uses a Gaussian radial basis function to approximate third-order B-spline functions and incorporates layer normalization for input adjustment, thereby accelerating the training speed. Additionally, AdaKAN employs AdaptiveLoss, which dynamically adjusts shape and scaling parameters, enabling outlier control while enhancing prediction accuracy and robustness under various data conditions.

(3) **Multi-modal data inputs and precise multi-step forecasting.** This work incorporates multi-source data inputs and employs Pearson correlation analysis to select predictive variables, facilitating the model's ability to capture comprehensive temporal information. It also addresses the more challenging multi-step forecasting task by simultaneously predicting the next 48 time steps of solar irradiance data and the next 96 time steps of electricity transformer temperature data in a single inference.

(4) **Comprehensive comparative experiments demonstrate the superior performance.** Evaluations in the distinct domains of solar irradiance and electricity transformer temperature reveal the model's effectiveness across different temporal scales and data scenarios. Compared with 12 cutting-edge time series forecasting methods, the results confirm that the combination of MDFM and AdaKAN significantly enhances the model's forecasting accuracy and robustness.

3. Methodology

The overall flowchart of the proposed method is illustrated in Fig. 1, which can be divided into seven steps: (1) Data Input: historical MTS data are used as inputs, with timestamps synchronized across all sources. (2) Pre-processing: data cleaning and normalization are applied, followed by Pearson correlation analysis to select features strongly correlated with the target variable as the final feature variables. (3) Downsampling: the input sequence is downsampled into subsequences at various temporal scales by average pooling, and each scale's data is decomposed into seasonal and trend components using discrete Fourier transform (DFT). (4) MDFM: this step performs Fine-to-coarse (FTC) and Coarse-to-fine (CTF) fusion on the seasonal and trend components across different temporal scales, followed by equal-scale fusion. (5) AdaKAN: the fused data are fed into AdaKAN, which is trained with a Gaussian radial basis function (GRBF) and an adaptive loss function for feature extraction and prediction. (6) Forecasting Results: the forecasting results at each temporal scale are projected to the original input scale and combined to generate the final output; (7) Performance Evaluation: the effectiveness of the proposed method is validated through comprehensive experimental evaluation. The implementation specifics are outlined in Algorithm 1.

Algorithm 1: MDFM-AdaKAN Algorithm

Input:

The raw time-series data is denoted as $x_{\text{raw}} \in \mathbb{R}^d$, where d is the number of variables.

Data pre-processing:

1. Data clean: Eliminate outliers and fill in missing values: $x_{\text{clean}} = \text{Clean}(x_{\text{raw}})$.
2. Normalization: Standardize x_{clean} to zero mean and unit variance using Z-score: $x = \frac{x_{\text{clean}} - \mu}{\sigma}, \quad \mu = \text{MEAN}(x_{\text{clean}}), \quad \sigma = \text{STD}(\mathbf{x}_{\text{clean}})$.
3. Data division: $set_{\text{train}}, set_{\text{test}}, set_{\text{val}}$ are divided in a ratio of 6:2:2.

DFT data decomposition:

4. Define the data representations at different temporal scales: $X_{\text{multi-scale}} = \{x_0, \ldots, x_m\}$ where m denotes the maximum downsampling level, and x_0 denotes the input time series after preprocessing. The representation at the jth temporal scale is denoted as: $x_j = \begin{cases} x_0, & j = 0 \\ \text{AvgPool}(x_{j-1},\ \text{stride} = 2), & j = 1, 2, \ldots, m \end{cases}$
5. Discrete Fourier Transform: Apply DFT to x_j to obtain $X_j = \text{DFT}(x_j)$, and remove the DC component: $X_j(0) = 0$.
6. Select dominant frequencies: Compute amplitude $|X_j(k)|$, define threshold $f_{j,\text{top5}}$ as the smallest of top-5 amplitudes, and construct index set $\mathcal{K}_j = \{k \mid |X_j(k)| > f_{j,\text{top5}}\}$.
7. Reconstruct the seasonal and trend components: $s_j = \text{IDFT}(X_j[k \in \mathcal{K}_j])$, and $t_j = x_j - s_j$. The multi-scale seasonal and trend components are: $S_{\text{seasonal}} = \{s_0, \ldots, s_m\}$, $T_{\text{trend}} = \{t_0, \ldots, t_m\}$.

MDFM process:

8. Different-scale fusion: Apply FTC to seasonal components and CTF to trend components. For temporal scale j, FTC: $s_j = s_j + \text{Embed}_j^{\text{FTC}}(s_{j-1})$; CTF: $t_j = t_j + \text{Embed}_j^{\text{CTF}}(t_{j+1})$.
9. Equal-scale fusion: Fuse seasonal and trend components at each temporal scale. $X_{\text{mixer}} = \{s_0 + t_0, s_1 + t_1, \ldots, s_m + t_m\}$.

AdaKAN process:

10. Feature enhancement for each temporal scale: Apply AdaKAN_1 to enhance the features, followed by a residual connection: $h_j^{enhanced} = \text{AdaKAN}_1(s_j + t_j) + (s_j + t_j)$.
11. Prediction: Map at each temporal scale to the target prediction length, the predicted output at scale j is: $\hat{y}_j = \text{AdaKAN}_2(h_j^{enhanced})$.
12. Projection and combination: Project to the final output channel, and sum predictions across all temporal scales: $\hat{y}_{\text{project}} = \sum_{j=0}^{m} \text{Proj}(\hat{y}_j)$.

Evaluation:

13. Denormalization: $\hat{y}_{\text{output}} = \hat{y}_{\text{project}} \cdot \sigma + \mu$.
14. Metrics: { mae, mse, nrmse, R^2, Test-time, GPU-mem, Params }.

3.1. MDFM: multi-scale data fusion module

3.1.1. DFT seasonal-trend decomposition

Transforming time-domain data into the frequency domain enables the effective identification of critical frequency components associated with variations in MTS data, effectively extracting inherent periodic patterns and capturing detailed fluctuations within the data. Additionally, this process eliminates irrelevant high-frequency noise while

retaining significant trend information. Consequently, this study utilizes DFT to decompose solar irradiance data across various temporal scales into seasonal and trend components.

The DFT process is as follows: First, the preprocessed MTS data is downsampled via average pooling to obtain sequences at different temporal scales $X_{\text{multi-scale}} = \{x_0, \ldots, x_m\}$. Where x_0 is the original sequence, m denotes the maximum downsampling level, and each x_j for $j > 0$ represents the sequence downsampled at scale j. Then, DFT is applied to convert time-domain data at each scale into the frequency domain, as shown in Eq. (1). Where $x_j(n)$ represents the value of the sequence at scale j at time point n in the time domain. The symbol i represents the imaginary unit. N denotes the sequence length at the current scale. The term $e^{-i2\pi kn/N}$ is the Fourier basis function corresponding to each frequency component. $X_j(k)$ represents the complex value of the kth frequency component in the frequency domain. To focus on data fluctuations, the DC component $X_j(0)$ is filtered out. The amplitude of each frequency k is computed as shown in Eq. (2), where $\text{Re}(X_j(k))$ and $\text{Im}(X_j(k))$ denote the real and imaginary parts of the complex coefficient $X_j(k)$, respectively. To identify dominant periodic patterns, Eq. (3) defines the threshold $f_{j,\text{top5}}$ as the smallest amplitude among the top five largest values of $|X_j(k)|$. Based on this threshold, the corresponding frequency index set $\mathcal{K}_j$ is constructed, as shown in Eq. (4), comprising all indices k for which the amplitude $|X_j(k)|$ exceeds $f_{j,\text{top5}}$. The seasonal component $s_j(n)$ is then reconstructed by applying an inverse DFT to the selected frequency components in $\mathcal{K}_j$, as described in Eq. (5). The trend component $t_j(n)$ is obtained by subtracting the seasonal component from the original sequence, as shown in Eq. (6). Through DFT, the multi-scale seasonal components $S_{\text{seasonal}} = \{s_0, \ldots, s_m\}$ and trend components $T_{\text{trend}} = \{t_0, \ldots, t_m\}$ are extracted across different temporal scales.

$$X_j(k) = \sum_{n=0}^{N-1} x_j(n) \cdot e^{\frac{-i2\pi kn}{N}}, \quad k = 0, 1, \ldots, N-1 \tag{1}$$

$$|X_j(k)| = \sqrt{\text{Re}(X_j(k))^2 + \text{Im}(X_j(k))^2} \tag{2}$$

$$f_{j,\text{top5}} = \min\left(|X_j(k_1)|, |X_j(k_2)|, \ldots, |X_j(k_5)|\right) \tag{3}$$

$$\mathcal{K}_j = \left\{k \mid |X_j(k)| > f_{j,\text{top5}}\right\} \tag{4}$$

$$s_j(n) = \sum_{k \in \mathcal{K}_j} X_j(k) \cdot e^{\frac{i2\pi kn}{N}} \tag{5}$$

$$t_j(n) = x_j(n) - s_j(n) \tag{6}$$

3.1.2. Multi-scale data fusion process

At different temporal scales, the seasonal and trend components of MTS data have distinct practical significance. The seasonal component facilitates the identification of recurring patterns in the short and medium term, with finer seasonal sequences more effectively capturing detailed fluctuations in MTS data. In contrast, downsampling enables the trend component to effectively smooth short-term fluctuations, thereby accurately reflecting long-term trends. Based on these observations, this study proposes a two-stage MDFM to comprehensively integrate multi-scale MTS data through different-scale fusion and equal-scale fusion, as illustrated in Fig. 2.

During the different-scale fusion stage, MDFM applies two opposing strategies to the decomposed seasonal and trend components: FTC and CTF. For the seasonal component, the FTC strategy is employed. It starts from the finest temporal scale, corresponding to the 0-th layer obtained by downsampling the original time series after preprocessing. Seasonal features are then progressively integrated into coarser temporal scales through feature embedding and residual connections. Specifically, the seasonal component at each temporal scale is first transformed via a two-layer feedforward neural network to perform feature embedding and dimensionality transformation. The embedding

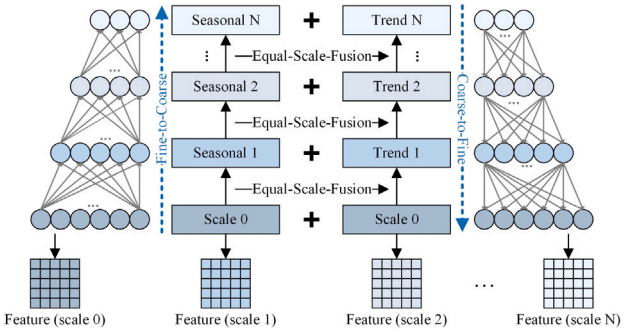

Fig. 2. Multi-scale data fusion process of MDFM.

process from the $(j-1)$-th temporal scale to the jth temporal scale is defined as Eq. (7). Where $W_{j-1}^{(1)}$ and $b_{j-1}^{(1)}$ denote the weight and bias of the first linear mapping layer, while $W_{j-1}^{(2)}$ and $b_{j-1}^{(2)}$ correspond to those of the second layer. The activation function $\phi(\cdot)$ is set to GELU.

After obtaining the embedded features, the model updates the seasonal feature at the jth scale by adding the embedded result $\text{Embed}_j^{\text{FTC}}(s_{j-1})$ via a residual connection. The final representations of seasonal features across all temporal scales under the FTC strategy can be formulated as Eq. (8). The FTC strategy preserves and enhances high-frequency seasonal information at coarser temporal levels by using fine-to-coarse fusion. This prevents the loss of detailed features during downsampling, thereby improving the model's ability to capture short-term periodic patterns.

$$\text{Embed}_j^{\text{FTC}}(s_{j-1}) = \phi\left(s_{j-1}W_{j-1}^{(1)} + b_{j-1}^{(1)}\right)W_{j-1}^{(2)} + b_{j-1}^{(2)} \tag{7}$$

$$s_j = \begin{cases} s_0, & j = 0 \\ s_j + \text{Embed}_j^{\text{FTC}}(s_{j-1}), & j = 1, \ldots, m \end{cases} \tag{8}$$

In contrast to the FTC strategy, the CTF strategy fuses trend components starting from the coarsest temporal scale and progressively incorporates information from finer temporal scales via feature embedding and residual connections. Specifically, the trend component is passed through a two-layer feedforward neural network to obtain an embedded representation. The embedding process from the $(j+1)$-th temporal scale to the jth temporal scale is defined as Eq. (9), where $W_{j+1}^{(1)'}$ and $b_{j+1}^{(1)'}$ denote the weight and bias of the first linear mapping layer, while $W_{j+1}^{(2)'}$ and $b_{j+1}^{(2)'}$ correspond to those of the second layer. After obtaining the embedded features, the model updates the trend feature at the jth temporal scale by adding the embedded result $\text{Embed}_j^{\text{CTF}}(t_{j+1})$ via a residual connection. The final representation of trend features across all temporal scales under the CTF strategy can be formulated as Eq. (10). By applying a coarse-to-fine fusion scheme, the CTF strategy transfers stable trend signals from coarse to fine temporal levels, thereby achieving an effective encoding of long-term trend information across all resolutions. This mechanism enhances the model's ability to perceive long-range trend variations and improves its performance in long-term forecasting tasks.

$$\text{Embed}_j^{\text{CTF}}(t_{j+1}) = \phi\left(t_{j+1}W_{j+1}^{(1)'} + b_{j+1}^{(1)'}\right)W_{j+1}^{(2)'} + b_{j+1}^{(2)'} \tag{9}$$

$$t_j = \begin{cases} t_m, & j = m \\ t_j + \text{Embed}_j^{\text{CTF}}(t_{j+1}), & j = m-1, \ldots, 0 \end{cases} \tag{10}$$

After completing the different-scale fusion of seasonal and trend components, the MDFM model performs equal-scale feature fusion at each temporal level to achieve a unified representation of time series information across resolutions. Specifically, for the jth temporal scale, after applying the FTC and CTF strategies, the fused seasonal and trend feature representations are denoted by s_j and t_j, respectively. Since the two types of features are aligned in both temporal and channel dimensions, element-wise addition can be directly applied to perform

equal-scale fusion, as shown in Eq. (11), where z_j denotes the equal-scale fused representation at the jth temporal scale. The final fused representations across all temporal scales can be formulated as Eq. (12). The multi-scale fused features X_{mixer} are subsequently forwarded to the downstream AdaKAN model, which further extracts semantically meaningful high-level representations for the final forecasting task.

$$z_j = s_j + t_j \tag{11}$$

$$X_{\text{mixer}} = \{z_0, z_1, \ldots, z_m\} = \{s_0 + t_0, s_1 + t_1, \ldots, s_m + t_m\} \tag{12}$$

The MDFM feature fusion method better aligns with the inherent characteristics of the seasonal and trend components, enhancing the model's understanding of the internal structure of the data. This approach ensures that each temporal scale contains both global and detailed features from all scales to varying extents, improving the representation of complex time series data and preventing the loss of important features. Consequently, it balances both short-term fluctuations and long-term trends, generating more comprehensive predictive information and improving the overall understanding and forecasting capability of the time series.

3.2. AdaKAN: adaptive loss KAN

3.2.1. Kolmogorov-arnold network

KAN is a deep learning model based on the Kolmogorov-Arnold theorem, which states that any continuous multivariate function can be represented as a combination of univariate functions, as shown in Eq. (13). Let $x \in \mathbb{R}^d$ denote the input vector to the KAN model, and d denotes the input feature dimension. $\phi_{(q,p)}(x_p)$ represents the nonlinear transformation applied to each input feature x_p. $\boldsymbol{\Phi}_q$ is a univariate nonlinear function applied to the aggregated outputs of the inner univariate transformations. $f(x)$ represents the multivariate function that the KAN model is designed to approximate. Based on the Kolmogorov-Arnold theorem, KAN extends its original structure to a multi-layer recursive structure that progressively learns the nonlinear relationships within the input data. Assuming the output dimension is 1, the recursive structure of KAN can be written as Eq. (14). In this formulation, x_{u_0} refers to the u_0-th element of the input vector x. L denotes the total number of layers in the network. d_l represents the number of summation nodes in the lth layer of the KAN model, with $d_0 = d$ indicating the dimensionality of the input vector. u_l indexes the u_l-th summation node in the lth layer. $\phi_{l,v,u}(\cdot)$ denotes a learnable univariate function defined on the edge from node u in layer l to node v in layer l+1, which applies a nonlinear transformation before aggregation.

Unlike traditional MLP, which relies on the universal approximation theorem, KAN's unique characteristic lies in placing adaptive activation functions on the edges rather than at the nodes, as illustrated in Eq. (15). In this configuration, $\phi(x)$ represent activation functions. w_b and w_s represent training weights. $b(x)$ denotes the base Silu activation function, as presented in Eq. (16). Furthermore, $spline(x)$ denotes the linear combination of B-spline basis functions, as shown in Eq. (17), where g_u indicates learnable parameters, $B_u(x)$ denotes the uth B-spline basis function. This unique architecture enables KAN to capture complex nonlinear relationships effectively, enhancing its accuracy in time series forecasting tasks.

However, KAN-based models for time series forecasting often face limitations in computational efficiency. B-spline exhibits low GPU utilization, and the approximation fitting process significantly increases the number of parameters required, while the initialization of the coefficients deviates from variance-preserving principles. Moreover, prior research lacks an automated approach to adjust the loss function under diverse data scenarios, resulting in prediction instability. This study introduces the AdaKAN to address these challenges, enhancing performance by utilizing GRBF and an adaptive loss function [43], as shown in Fig. 3.

$$f(x) = f(x_1, \ldots, x_d) = \sum_{q=1}^{2d+1} \boldsymbol{\Phi}_q \left(\sum_{p=1}^{d} \phi_{(q,p)}(x_p) \right) \tag{13}$$

$$\begin{aligned} f(x) = \sum_{u_{L-1}=1}^{d_{L-1}} \phi_{(L-1,u_L,u_{L-1})} \Bigg(\sum_{u_{L-2}=1}^{d_{L-2}} \Bigg(\sum_{u_2=1}^{d_2} \phi_{(2,u_3,u_2)} \cdots \\ \Bigg(\sum_{u_1=1}^{d_1} \phi_{(1,u_2,u_1)} \Bigg(\sum_{u_0=1}^{d_0} \phi_{(0,u_1,u_0)}(x_{u_0}) \Bigg) \Bigg) \Bigg) \cdots \Bigg) \end{aligned} \tag{14}$$

$$\phi(x) = w_b b(x) + w_s \, spline(x) \tag{15}$$

$$b(x) = \text{silu}(x) = \frac{x}{1 + e^{-x}} \tag{16}$$

$$spline(x) = \sum_u g_u B_u(x) \tag{17}$$

3.2.2. Gaussian radial basis function

Radial basis functions (RBFs) can approximate any nonlinear function, handle complex regularities in time-series data, demonstrate strong generalization ability, and achieve fast learning convergence speed. The Gaussian radial basis function (GRBF) is one of the most widely used RBFs, defined as $\varphi(r) = e^{-\epsilon r^2}$. In our approach, we set $r = (x - \theta)/h$ and $\epsilon = 1/2$. The variable r represents the Euclidean distance between the input vector x, and the center vector θ, ϵ is a shape parameter that controls the width of the Gaussian function ($\epsilon > 0$), and h denotes the scaling factor for the Gaussian width. The GRBF with N centers $\{\theta_v\}_{v=1}^N$ is formulated as shown in Eq. (18), where w_v represents adjustable weights corresponding to the v-th basis function.

The performance of GRBF is primarily influenced by the spatial distribution and the number of its center vectors. A combined strategy is adopted to enhance the modeling capacity and computational efficiency. Layer normalization is applied to the input data to standardize it into a numerical range aligned with the activation region of the radial basis functions. Furthermore, hyperparameter optimization is employed to determine both the spatial distribution and the optimal number of centers, thereby ensuring that the activation regions adequately cover the normalized input domain and enabling accurate modeling of nonlinear patterns while controlling computational cost. This optimization enhances numerical stability and computational efficiency in the RBF network without sacrificing model accuracy. Compared to B-spline curves, GRBF offers a more scalable approach in high-dimensional spaces.

$$GRBF(x) = \sum_{v=1}^{N} \omega_v \varphi(r_v) = \sum_{v=1}^{N} \omega_v \exp\left(-\frac{1}{2} \left(\frac{x - \theta_v}{h} \right)^2 \right) \tag{18}$$

3.2.3. Adaptive loss function

Loss functions play a critical role in prediction outcomes. The commonly used mean squared error loss function (MSELoss) is highly sensitive to outliers, leading to poor performance when forecasting MTS under volatile or noisy conditions. Although HuberLoss demonstrates improved robustness against outliers, it performs less effectively under relatively stable data conditions. To achieve high-accuracy and reliable predictions under varying data distribution characteristics, AdaKAN employs a general and adaptive loss function (AdaptiveLoss) to enhance algorithm robustness, as shown in Eq. (19). In this formulation, ξ denotes the residual (the deviation between the predicted and true values), α is the shape parameter, and c is the scale parameter.

$$f(\xi, \alpha, c) = \frac{|\alpha - 2|}{\alpha} \left(\left(\frac{(\xi/c)^2}{|\alpha - 2|} + 1 \right)^{\alpha/2} - 1 \right) \tag{19}$$

The shape parameter α controls the sensitivity of the loss function to large residuals. Higher values of α result in a loss function that

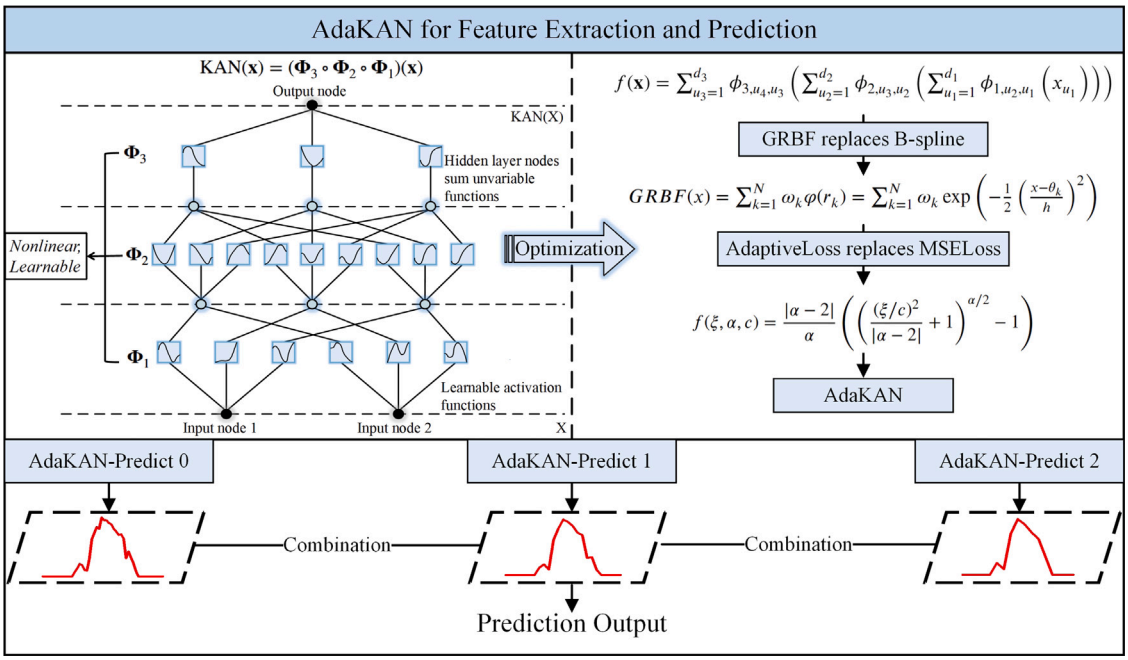

Fig. 3. Feature extraction and prediction process of AdaKAN.

grows more rapidly with increasing residuals, making the model more sensitive to large errors. In contrast, lower values flatten the loss in the tail region, suppressing the influence of outliers. To ensure numerical stability, α is constrained to the interval $(0, 3)$, with $\alpha = 0$ and $\alpha = 2$ excluded due to removable singularities.

The scale parameter c serves as a residual normalization factor, effectively defining the threshold that separates small and large errors. A smaller c leads to more aggressive penalization, while a larger c reduces sensitivity to residuals and mitigates the influence of outliers. In this work, the value of c is set within the range $(10^{-5}, +\infty)$.

During the initial training stage, the parameters α and c are initialized as $\alpha_{\text{init}} = 1.5$ and $c_{\text{init}} = 1.0$, respectively, and subsequently optimized jointly with the main network parameters through backpropagation. This enables the model to dynamically adapt to the data distribution. The mathematical structure of AdaptiveLoss suggests that when residuals are small and the data distribution is stable, the model tends to learn larger values of α and smaller values of c. This increases its sensitivity to small residuals and enhances its capacity to capture fine-grained variations. Conversely, when the data exhibit severe noise or large outliers, the model tends to adaptively reduce α to weaken the influence of large residuals and improve robustness. The scale parameter c is expected to increase accordingly to reduce sensitivity to outliers and avoid excessive influence on the loss function.

Through adaptive adjustment of the shape parameter α, AdaptiveLoss is capable of approximating several classical loss functions under certain conditions, as defined in Eq. (20), thereby providing an effective and unified approach for dynamically controlling the robustness of the model. Specifically, when $\alpha = 2$, the loss function approximates the traditional L2 loss, which is highly sensitive to all residuals and is suitable for modeling Gaussian-distributed noise. When $\alpha = 1$, the function is equivalent to the Charbonnier loss, characterized by smoother gradients and slower growth. In the limiting case $\alpha \to 0$, the function approximates the negative log-likelihood of the Cauchy distribution, exhibiting strong suppression of large residuals. When α takes negative values ($\alpha = -2$ or $\alpha \to -\infty$), the loss function exhibits redescending behavior, resembling that of Geman–McClure and Welsch. These functions exhibit the characteristic redescending behavior, in which the gradient approaches zero once the residual exceeds a certain threshold. This behavior halts parameter updates for extreme outliers and thereby enhances the model's robustness. However, such settings may lead to numerical instability and optimization difficulties.

$$f(\xi, \alpha, c) = \begin{cases} \frac{1}{2}\left(\frac{\xi}{c}\right)^2, & \text{for } \alpha \to 2;\ \text{L2 Loss} \\ \sqrt{\left(\frac{\xi}{c}\right)^2 + 1} - 1, & \text{for } \alpha = 1;\ \text{Charbonnier Loss} \\ \log\left(\left(\frac{\xi}{c}\right)^2 + 1\right), & \text{for } \alpha \to 0;\ \text{Cauchy Loss} \\ \frac{2\left(\frac{\xi}{c}\right)^2}{\left(\frac{\xi}{c}\right)^2 + 4}, & \text{for } \alpha = -2;\ \text{GMC Loss} \\ 1 - \exp\left(-\frac{1}{2}\left(\frac{\xi}{c}\right)^2\right), & \text{for } \alpha \to -\infty;\ \text{Welsch Loss} \end{cases} \tag{20}$$

In summary, the proposed AdaptiveLoss provides a unified formulation for a wide range of classical loss functions. By introducing learnable parameters α and c, AdaptiveLoss can adaptively fit the data distribution. Unlike conventional fixed-form loss functions, this adaptive approach significantly enhances robustness to noise and outliers. Furthermore, by learning loss parameters directly through gradient-based optimization, the method eliminates the need for manual tuning procedures, thereby streamlining the model development pipeline in practical scenarios.

4. Dataset and experimental framework

4.1. Datasets and preprocessing

This research selected two MTS datasets from different domains for the experiments: the solar irradiance dataset and the electricity transformer temperature dataset. The solar irradiance dataset [46] was collected from California's Central Valley between 2014 and 2016. We selected solar irradiance, sky image, meteorological, and NWP data from 2014 to 2015 at 20 min intervals and conducted feature selection and data processing before the experiment. To better represent predictive performance across different seasons, data from 2014 were designated as the training set, and one month from each season in 2015

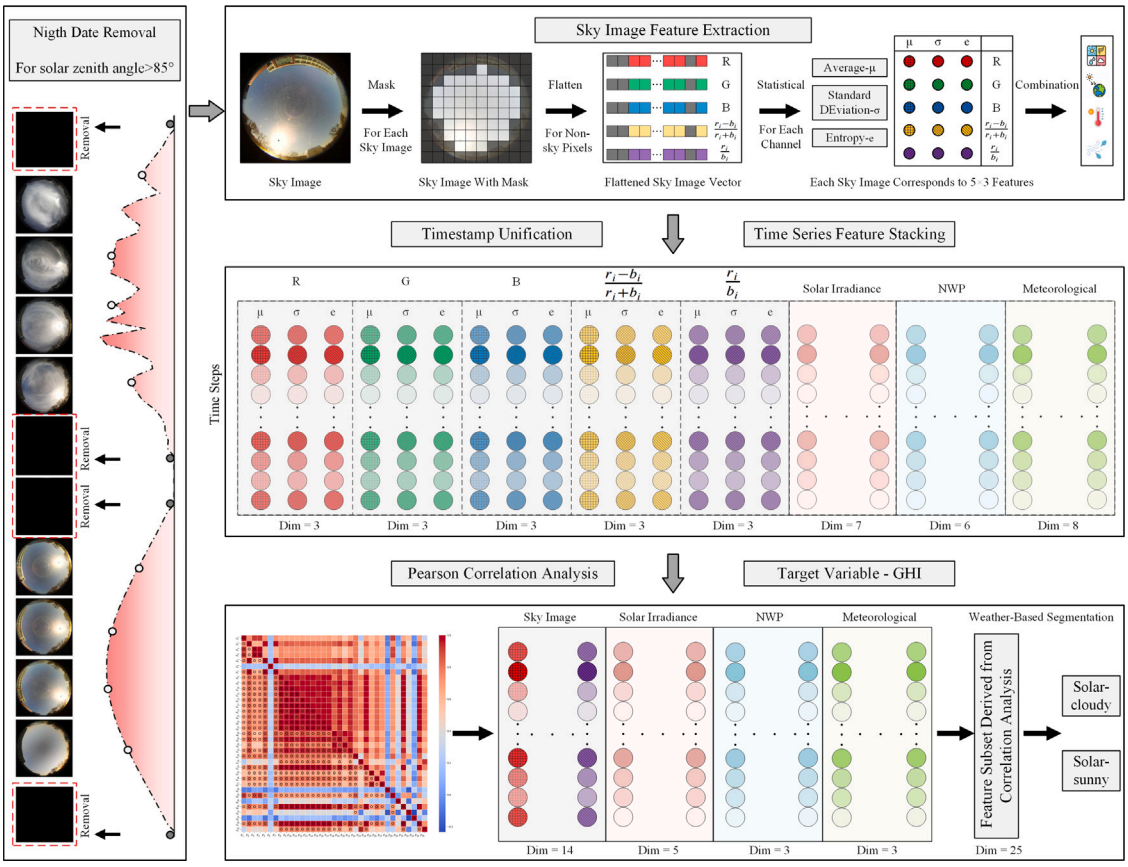

Fig. 4. Preprocessing workflow of solar irradiance dataset.

was allocated to the test and validation sets, resulting in a (6:2:2) split for the training, test, and validation sets.

The preprocessing pipeline for the solar irradiance dataset is illustrated in Fig. 4. Abnormal data corresponding to solar zenith angles greater than 85 degrees were excluded. Missing values were imputed using the mean of adjacent observations. Pixel-level image segmentation was applied to isolate sky regions and remove pixels corresponding to buildings or other occlusions. RGB channel values were extracted from the segmented sky regions. Two pixel-wise features were computed: the red-to-blue ratio ($\frac{r_i}{b_i}$) and the normalized red-to-blue ratio ($\frac{r_i-b_i}{r_i+b_i}$), aimed at enhancing cloud pattern detection and robustness to noise. These derived features, together with the original RGB values, were flattened into one-dimensional vectors. Additionally, statistical metrics, including mean (μ), standard deviation (σ), and entropy (e), were calculated for all five channels to characterize the distribution, variability, and randomness of pixel intensities. Subsequently, the extracted sky image features were combined with other relevant features, including solar irradiance features, NWP features, and meteorological features. Pearson correlation analysis was applied to the resulting 36-dimensional feature set to identify features that were highly correlated with the target variable, global horizontal irradiance (GHI). As a result, 25 key features were selected for subsequent experimental investigation. The corresponding correlation matrix is shown in Fig. 5. To evaluate the model's predictive performance under different data scenarios, we further divided the solar irradiance dataset (Solar-all) into two subsets, Solar-sunny and Solar-cloudy, based on daily rainfall, cloud cover, and GHI fluctuations.

For the electricity transformer temperature dataset, this study utilizes the ETDataset [35], which is widely used for evaluating the performance of time series forecasting models. Four publicly subsets, including ETTh1, ETTh2, ETTm1, and ETTm2, were selected for experimental evaluation. All subsets have been subjected to standardized

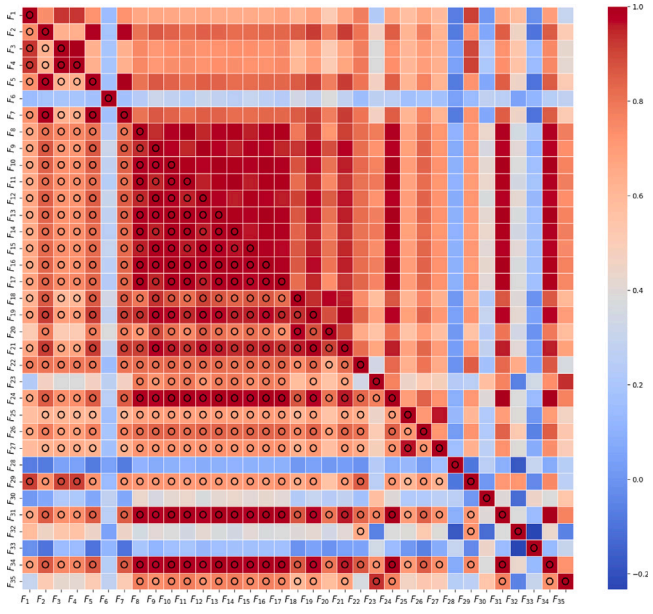

Fig. 5. Feature correlation analysis results.

preprocessing procedures, including anomaly detection and imputation of missing values, and require only feature normalization prior to model deployment. Detailed information on all datasets used in this study is summarized in Table 1.

Table 1
Summary of datasets.

Dataset	Dim	Dataset size	Frequency	Type
ETTh1	7	(8545, 2881, 2881)	1 h	Electricity (°C)
ETTh2	7	(8545, 2881, 2881)	1 h	Electricity (°C)
ETTm1	7	(34465, 11521, 11521)	15 min	Electricity (°C)
ETTm2	7	(34465, 11521, 11521)	15 min	Electricity (°C)
Solar-all	25	(17280, 5760, 5760)	20 min	Energy (W/m^2)
Solar-sunny	25	(10748, 3582, 3582)	20 min	Energy (W/m^2)
Solar-cloudy	25	(6532, 2178, 2178)	20 min	Energy (W/m^2)

4.2. Comparative studies

In this study, 12 cutting-edge time series forecasting methods were selected for comparative evaluation, including TimeMixer [39], Autoformer [47], DLinear [40], FEDformer [48], FiLM [49], Informer [35], iTransformer [50], LightTS [51], PatchTST [52], TiDE [53], TSMixer [54], and TimesNet [17]. A brief overview of these methods is provided below.

The iTransformer is an innovative architecture for time series forecasting that adapts the Transformer model to transposed dimensions. It treats each variable's entire time series as a token in multivariate data. This approach enhances the model's ability to capture correlations across feature dimensions via self-attention. TimeMixer is a time series forecasting model based on MLP, characterized by its simplicity and computational efficiency. Mixing features across time steps effectively captures temporal patterns and dependencies, achieving competitive forecasting accuracy. Autoformer introduces a self-attention mechanism and designs an internal module for sequence decomposition, breaking the convention of using sequence decomposition solely as a preprocessing step. This design enables progressive decomposition of complex time series, enhancing efficiency and accuracy. FEDformer introduces a frequency-enhanced attention mechanism to capture long-term dependencies and short-term fluctuations in time series. However, its complexity may lead to slower computation in large-scale tasks. FiLM employs Legendre polynomial projections to approximate historical information, uses Fourier projections to remove noise, and incorporates low-rank approximations to accelerate computation, significantly improving the accuracy of existing models in both multivariate and univariate long-term forecasting. Informer enhances model computational efficiency through a sparse attention mechanism and effectively captures long-term dependencies in large-scale time series via self-attention distillation, providing an efficient solution for long-term forecasting. LightTS introduces adaptive ensemble distillation, utilizing dimensionality reduction and lightweight sequence modeling strategies to handle large-scale time series data. Through an efficient feature extraction mechanism, LightTS significantly reduces the time and computational resources required for data processing while maintaining predictive accuracy. PatchTST segments MTS into different channels, treating each channel as a univariate time series to avoid potential inter-variable information entanglement present in traditional methods, thereby improving model generalization and efficiency. TiDE proposes an MLP-based time series dense encoder for long-term forecasting, achieving a speed 5 to 10 times faster than Transformer-based models. TSMixer combines temporal and spatial mixing strategies, using linear layers to replace traditional attention mechanisms, and is commonly applied in multivariate forecasting and representation learning for patch time series. TimesNet is a novel time series analysis model that enhances representation capability by decomposing temporal variations into intra-period and inter-period components and transforming them into two-dimensional tensors, demonstrating strong performance across various time series tasks. DLinear is a linear model based on decomposition, with its core idea of decomposing a time series into trend and residual components. These components are modeled separately using two linear layers, yielding promising results in short-term forecasting tasks.

4.3. Experimental setting

This study presents comprehensive experimental results on two distinct MTS datasets, with an in-depth analysis conducted on the more complex solar irradiance dataset. First, prediction accuracy and computational cost are evaluated using two temporal feature extraction strategies: independent modeling at single temporal scales and multi-scale data fusion. Second, accuracy comparisons are conducted with 12 baseline methods on ETDataset (ETTm1, ETTm2, ETTh1, ETTh2) and solar irradiance datasets (Solar-all, Solar-cloudy, Solar-sunny), highlighting the superior performance of the proposed approach. Subsequently, ablation experiments, robustness analyses, and computational efficiency evaluations were conducted to assess the contribution of each module to prediction accuracy, verify the stability of the algorithm, and validate computational performance. Additionally, this study evaluates the model's performance across various forecasting horizons and performs a detailed analysis of the employed GRBF and AdaptiveLoss, as presented in Appendices A–C.

To ensure the fairness of the experimental results, we fixed the batch size at 32 for all methods on the ETDataset, using the default parameters provided in the official code repositories or the optimal parameter combinations mentioned in the respective papers, predicting 96 future steps based on the previous 96 steps. For the solar irradiance datasets, the batch size was set to 16, with all other hyperparameters aligned with those used in our method. Considering the continuous variations of effective daylight hours during the data collection period, the historical and forecast steps were both standardized to 48. Each time step represents a 20 min interval, resulting in a total forecasting horizon of 16 h, aligned with the longest daylight period recorded in summer. Shorter daylight periods were supplemented with zero values to ensure consistency. To comprehensively evaluate the performance of the proposed algorithm, this study adopts four standard evaluation metrics: mean absolute error (MAE), mean squared error (MSE), normalized root mean squared error (nRMSE), and coefficient of determination (R^2) [55].

5. Experimental results analysis

5.1. Multi-scale data fusion effect

This section presents the prediction results obtained from both single temporal scale modeling and multi-scale data fusion. Specifically, the multi-scale fusion results are generated using the proposed MDFM, while the single-scale results are obtained by disabling the different-scale fusion stage within MDFM. Taking the solar irradiance dataset as an example, the experimental results are shown in Fig. 6. The single-scale prediction results are denoted as S_0 to S_4, where S_0 corresponds to the prediction based on the original temporal resolution inputs, and S_1 to S_4 correspond to predictions based on first-order to fourth-order downsampled inputs, respectively. The multi-scale fusion results are denoted by $FS_{(0\text{-}1)}$ to $FS_{(0\text{-}4)}$, where $FS_{(0\text{-}n)}$ represents the prediction obtained by fusing temporal features extracted from S_0 to S_n.

Fig. 6(a) presents the fitted curves of predicted versus actual values within a single day. Fig. 6(b)–(j) illustrates the prediction error bands corresponding to each method, where the bandwidth reflects the degree of deviation between predicted and actual values. As illustrated in Fig. 6(a)–(j), predictions based on the lower downsampling rates (S_0 and S_1) better capture the fine-grained details, such as the fluctuations in solar irradiance over the next day. However, the lower downsampling rates render predictions more sensitive to noise and more likely to be affected by anomalous data fluctuations. In contrast, predictions from higher downsampling rates (S_2) better capture the overall daily trend of solar irradiance. However, as downsampling levels increase, the prediction curves become excessively smooth, resulting in a loss of finer details (S_3 and S_4). The prediction curves produced by the multi-scale data fusion strategy more accurately capture the overall

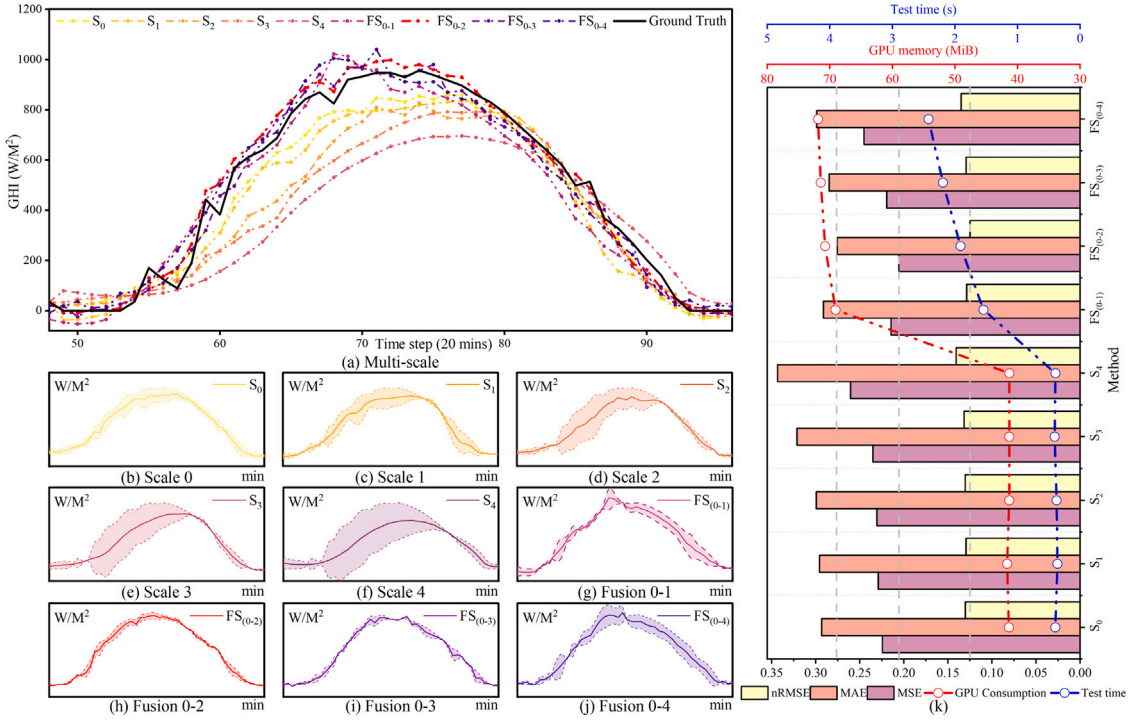

Fig. 6. Comparison of prediction results and error curves across different temporal scales.

trends and demonstrate enhanced capability in modeling fine-grained variations. In addition, this approach leads to a noticeable reduction in prediction error.

Fig. 6(k) illustrates the prediction accuracy and computational cost of each method on the solar irradiance dataset from an overall perspective. The results indicate that prediction error increases progressively from S_0 to S_4, while differences in computation time and GPU memory usage remain relatively small. With the adoption of the multi-scale data fusion strategy, the computational cost increases moderately, showing a gradual upward trend as the number of fused scales grows. In contrast, the prediction error exhibits a decreasing-then-increasing pattern, with $FS_{(0\text{-}2)}$ yielding the lowest error among all configurations. Based on these findings, a three-scale data fusion strategy, which incorporates the original resolution and its first and second downsampled versions (S_0, S_1, and S_2), was selected for the solar irradiance forecasting task. This configuration effectively balances the ability to capture overall temporal trends with the preservation of essential fine-grained variations.

5.2. Algorithm accuracy verification

Tables 2–3 present prediction results of various comparison methods across different datasets. The best evaluation metrics for each dataset are highlighted in bold. The results indicate that the proposed MDFM-AdaKAN method consistently outperforms other methods in terms of overall accuracy. Specifically, MDFM-AdaKAN ranked first on the ETTh1, ETTm1, and ETTm2 datasets, and secured second place on ETTh2, with only a marginal performance gap relative to the TiDE method. Compared to the average performance of baseline methods across four datasets, MDFM-AdaKAN improved MSE, MAE, nRMSE, and R^2 by 28.0%, 19.7%, 13.4%, and 16.4%, respectively. Relative to the best-performing baseline method PatchTST, MDFM-AdaKAN showed improvements of 3.6%, 3.9%, 1.8%, and 1.2%. On the solar irradiance datasets (Solar-all, Solar-cloudy, and Solar-sunny), MDFM-AdaKAN ranked first in all cases, with improvements in MSE, MAE, nRMSE, and R^2 over the average performance of comparison methods by 20.4%, 20.5%, 10.8%, and 9.1%, respectively. Compared to the best baseline method, TimesNet, MDFM-AdaKAN achieved gains of 1.4%, 5.6%, 0.7%, and 6.2%.

To demonstrate the performance of MDFM-AdaKAN under different data scenarios, we take solar irradiance as an example and plot stacked polar charts for various weather conditions, as shown in Fig. 7. Since the evaluation standards for the four metrics differ, with higher R^2 values indicating better performance and lower MSE, MAE, and nRMSE values being preferable, we applied reversal and scaling to MSE, MAE, and nRMSE values. This adjustment presents the results more intuitively while maintaining data proportions. The best-performing method is highlighted in the chart with a red dashed circle. It is evident from the result that our method demonstrated superior performance for MAE and R^2 metrics across all weather conditions. For MSE and nRMSE metrics, MDFM-AdaKAN achieved top performance under sunny conditions and ranked second in the other two datasets. These findings indicate that MDFM-AdaKAN performs consistently well across all metrics in different weather conditions.

Fig. 8 shows the prediction curves for the next day's GHI under both sunny and cloudy conditions. As depicted in Fig. 8, concurrently generating prediction results for the next day's GHI is a challenging task due to irregular fluctuations caused by factors such as cloud cover. These fluctuations lead to a generally low overlap between the predicted and actual values across all methods, with the effect being more pronounced under cloudy conditions. Despite these challenges, the proposed method demonstrates superior performance among all compared approaches.

To further assess the interpretability of the model, we applied DFT to transform the prediction results into the frequency domain. A comparison of the frequency spectra among different prediction methods provides an intuitive assessment of how effectively each method captures high-frequency and low-frequency signals. This helps determine whether the prediction methods capture the periodic characteristics of the solar irradiance data. The comparison results are presented in Fig. 9, where the horizontal axis denotes the normalized frequency, and the vertical axis represents the amplitude of each frequency component derived from the GHI data in the frequency domain. As depicted, all methods perform relatively well under sunny conditions, with the amplitude of each frequency component closely aligning with the actual

Table 2
Comparison of prediction accuracy on ETDataset.

Datasets	Methods	MSE	MAE	nRMSE	R^2	Metrics rank	Avg rank
ETTh1 Dataset	MDFM-AdaKAN	**0.3643**	**0.3880**	**0.0690**	**0.6592**	(1,1,1,1)	**1**
	TimeMixer	0.4178	0.4478	0.0739	0.6081	(8,10,8,8)	8
	Autoformer	0.4489	0.4573	0.0766	0.5815	(11,11,11,11)	11
	DLinear	0.3962	0.4108	0.0720	0.6350	(7,5,7,6)	7
	FEDformer	0.3771	0.4185	0.0702	0.6446	(2,7,2,3)	3
	FiLM	0.4384	0.4332	0.0757	0.5830	(10,8,10,10)	10
	Informer	0.9523	0.7735	0.1116	0.1159	(13,13,13,13)	13
	iTransformer	0.3950	0.4098	0.0718	0.6331	(6,4,6,7)	6
	LightTS	0.4351	0.4436	0.0754	0.5932	(9,9,9,9)	9
	PatchTST	0.3771	0.3969	0.0702	0.6497	(3,3,2,2)	2
	TiDE	0.3847	0.3925	0.0709	0.6432	(4,2,4,4)	3
	TSMixer	0.4941	0.5020	0.0804	0.5410	(12,12,12,12)	12
	TimesNet	0.3891	0.4120	0.0713	0.6361	(5,6,5,5)	5
ETTh2 Dataset	MDFM-AdaKAN	0.2903	0.3385	0.0672	**0.8067**	(3,2,3,1)	2
	TimeMixer	0.2902	0.3427	0.0672	0.8044	(2,3,2,4)	3
	Autoformer	0.3637	0.4000	0.0753	0.7567	(10,10,10,10)	10
	DLinear	0.3415	0.3953	0.0729	0.7767	(8,9,8,8)	8
	FEDformer	0.3507	0.3916	0.0739	0.7622	(9,8,9,9)	9
	FiLM	0.3223	0.3644	0.0708	0.7793	(6,6,6,6)	6
	Informer	2.8671	1.3369	0.2113	−0.9548	(13,13,13,13)	13
	iTransformer	0.3004	0.3496	0.0684	0.7982	(5,5,5,5)	5
	LightTS	0.4143	0.4510	0.0803	0.7269	(11,11,11,11)	11
	PatchTST	0.2906	0.3450	0.0673	0.8050	(4,4,4,3)	4
	TiDE	**0.2894**	**0.3382**	**0.0671**	0.8063	(1,1,1,2)	**1**
	TSMixer	1.0559	0.8065	0.1282	0.2948	(12,12,12,12)	12
	TimesNet	0.3370	0.3709	0.0724	0.7770	(7,7,7,7)	7
ETTm1 Dataset	MDFM-AdaKAN	**0.3084**	**0.3438**	**0.0625**	**0.6799**	(1,1,1,1)	**1**
	TimeMixer	0.3203	0.3581	0.0637	0.6729	(2,2,2,2)	2
	Autoformer	0.4667	0.4620	0.0769	0.5460	(11,11,11,11)	11
	DLinear	0.3460	0.3737	0.0662	0.6656	(6,5,6,4)	4
	FEDformer	0.3637	0.4105	0.0679	0.6299	(9,9,9,9)	9
	FiLM	0.3536	0.3709	0.0669	0.6583	(7,4,7,5)	6
	Informer	0.6219	0.5600	0.0887	0.4143	(13,13,13,13)	13
	iTransformer	0.3413	0.3764	0.0657	0.6495	(5,8,5,7)	7
	LightTS	0.3933	0.4133	0.0706	0.6072	(10,10,10,10)	10
	PatchTST	0.3279	0.3676	0.0644	0.6661	(3,3,3,3)	3
	TiDE	0.3550	0.3741	0.0670	0.6486	(8,6,8,8)	8
	TSMixer	0.4786	0.4699	0.0778	0.5271	(12,12,12,12)	12
	TimesNet	0.3345	0.3764	0.0651	0.6522	(4,7,4,6)	4
ETTm2 Dataset	MDFM-AdaKAN	**0.1741**	**0.2535**	**0.0514**	**0.8764**	(1,1,1,1)	**1**
	TimeMixer	0.1762	0.2583	0.0517	0.8744	(2,2,2,2)	2
	Autoformer	0.2362	0.3160	0.0598	0.8276	(11,10,11,11)	11
	DLinear	0.1934	0.2928	0.0541	0.8639	(9,9,9,8)	9
	FEDformer	0.1917	0.2812	0.0539	0.8615	(8,8,8,9)	8
	FiLM	0.1834	0.2662	0.0527	0.8687	(5,5,5,4)	4
	Informer	0.3735	0.4587	0.0752	0.7362	(13,13,13,13)	13
	iTransformer	0.1837	0.2668	0.0528	0.8681	(6,6,6,6)	6
	LightTS	0.2264	0.3260	0.0586	0.8412	(10,11,10,10)	10
	PatchTST	0.1829	0.2672	0.0526	0.8686	(4,7,4,5)	5
	TiDE	0.1821	0.2649	0.0525	0.8698	(3,3,3,3)	3
	TSMixer	0.2498	0.3661	0.0615	0.8266	(12,12,12,12)	12
	TimesNet	0.1888	0.2659	0.0535	0.8649	(7,4,7,7)	7

values, indicating that the methods effectively capture the time-series characteristics under sunny conditions.

However, under cloudy conditions, the overall performance of all methods declines, especially in high-frequency components. High-frequency components represent rapid fluctuations or short-term variability, which become more pronounced and challenging to predict under cloudy conditions due to atmospheric factors like cloud movement and rainfall. Such abrupt variations are challenging for models to capture accurately, resulting in larger discrepancies in the amplitude between the predicted and actual high-frequency signals. In contrast, low-frequency components, which reflect longer-term trends, are generally less affected by transient weather events, making them easier for models to predict. In summary, MDFM-AdaKAN demonstrates a stronger ability to capture both short-term fluctuations and long-term trends, underscoring its excellent interpretability.

5.3. Algorithm efficiency analysis

This section evaluates the computational efficiency of various algorithms on the solar irradiance dataset. The analysis focused on the computation time, GPU memory usage, parameter counts, and the corresponding MAE values during the testing process. Additionally, the performance improvements achieved by replacing KAN with AdaKAN within the MDFM-AdaKAN framework are analyzed, with KAN using the default hyperparameter settings from the official code repository. The comparative results are presented in Table 4 and Fig. 10, where the color of each orb indicates different models. The horizontal position of the orb's center corresponds to the test time, while the vertical position represents the associated MAE value. The diameter of each orb indicates the GPU memory usage for the respective model.

Table 3
Comparison of prediction accuracy on Solar irradiance datasets.

Datasets	Methods	MSE	MAE	nRMSE	R^2	Metrics rank	Avg rank
Solar-all Dataset	MDFM-AdaKAN	0.1610	**0.2159**	0.1138	**0.8161**	(2,1,2,1)	**1**
	TimeMixer	0.1732	0.2463	0.1180	0.7448	(7,4,7,9)	6
	Autoformer	0.1856	0.2874	0.1222	0.7420	(11,12,11,10)	12
	DLinear	0.1816	0.2623	0.1208	0.7362	(10,8,10,11)	10
	FEDformer	0.1683	0.2833	0.1164	0.7557	(5,11,5,7)	7
	FiLM	0.1870	0.2581	0.1227	0.7273	(13,6,13,13)	13
	Informer	0.1731	0.2718	0.1179	0.7926	(6,9,6,2)	5
	iTransformer	0.1611	0.2447	0.1138	0.7566	(3,3,3,6)	3
	LightTS	0.1614	0.2592	0.1140	0.7721	(4,7,4,4)	4
	PatchTST	0.1795	0.2796	0.1202	0.7541	(9,10,9,8)	9
	TiDE	0.1864	0.2570	0.1224	0.7301	(12,5,12,12)	11
	TSMixer	0.1791	0.3158	0.1200	0.7680	(8,13,8,5)	8
	TimesNet	**0.1580**	0.2311	**0.1127**	0.7837	(1,2,1,3)	2
Solar-cloudy Dataset	MDFM-AdaKAN	0.2056	**0.2754**	0.1249	**0.6806**	(2,1,2,1)	**1**
	TimeMixer	0.2159	0.3003	0.1280	0.5351	(6,2,6,12)	6
	Autoformer	0.2801	0.4241	0.1458	0.5637	(13,13,13,10)	13
	DLinear	0.2313	0.3214	0.1325	0.5847	(9,7,9,9)	9
	FEDformer	**0.2034**	0.3317	**0.1242**	0.6313	(1,11,1,3)	3
	FiLM	0.2414	0.3233	0.1353	0.5885	(11,9,11,7)	11
	Informer	0.2313	0.3205	0.1325	0.6532	(8,6,8,2)	5
	iTransformer	0.2067	0.3050	0.1252	0.5939	(3,5,3,5)	3
	LightTS	0.2494	0.3526	0.1375	0.5511	(12,12,12,11)	12
	PatchTST	0.2268	0.3254	0.1312	0.5863	(7,10,7,8)	8
	TiDE	0.2402	0.3232	0.1349	0.5929	(10,8,10,6)	9
	TSMixer	0.2097	0.3025	0.1262	0.4474	(5,4,5,13)	7
	TimesNet	0.2072	0.3023	0.1254	0.5960	(4,3,4,4)	2
Solar-sunny Dataset	MDFM-AdaKAN	**0.0113**	**0.0754**	**0.0368**	**0.9864**	(1,1,1,1)	**1**
	TimeMixer	0.0121	0.0800	0.0380	0.9851	(3,3,3,2)	3
	Autoformer	0.0315	0.1382	0.0612	0.9560	(12,12,12,12)	12
	DLinear	0.0171	0.0973	0.0451	0.9784	(8,8,8,8)	8
	FEDformer	0.0222	0.1181	0.0515	0.9711	(10,11,10,10)	10
	FiLM	0.0149	0.0858	0.0422	0.9804	(5,5,5,5)	5
	Informer	0.0414	0.1471	0.0702	0.9443	(13,13,13,13)	13
	iTransformer	0.0222	0.1137	0.0515	0.9697	(11,10,11,11)	11
	LightTS	0.0164	0.0937	0.0442	0.9790	(7,7,7,7)	7
	PatchTST	0.0214	0.1125	0.0505	0.9732	(9,9,9,9)	9
	TiDE	0.0152	0.0873	0.0426	0.9800	(6,6,6,6)	6
	TSMixer	0.0142	0.0845	0.0411	0.9827	(4,4,4,4)	4
	TimesNet	0.0120	0.0763	0.0378	0.9844	(2,2,2,3)	2

Experimental results indicate that AdaKAN significantly outperforms the original KAN in prediction accuracy, confirming the effectiveness of integrating AdaptiveLoss and GRBF. Additionally, AdaKAN exhibits superior computational efficiency, with lower GPU memory usage (102 MiB) and fewer parameters (56,159). The computation time on the test set was 1.894 s, substantially shorter than the 25.511 s required by KAN, highlighting the efficiency gains achieved by replacing B-splines with GRBF. These results demonstrate that AdaKAN offers a comprehensive performance advantage over KAN. Compared to other methods, the proposed approach demonstrates moderate computational efficiency, closely aligning with attention-based models. However, a notable gap remains compared to more lightweight models such as DLinaer, TSMixer, and LightTS, indicating the potential direction for further optimization in KAN-based models. In summary, the proposed method achieves significant improvements in computational efficiency over the original KAN model. Although the overall computational efficiency remains at a moderate level relative to other methods, this limitation is effectively offset by substantial gains in prediction accuracy, making the proposed approach an ideal choice for high-precision forecasting applications.

5.4. Ablation experiment

An ablation study was conducted to assess the effectiveness of each component within the MDFM-AdaKAN framework. The DFT, KAN(GRBF)-F, KAN(GRBF)-P, and AdaptiveLoss modules were sequentially added, and the resulting changes in prediction accuracy are presented in Table 5. The combination 1 represents the baseline without these modules, serving as a control for the ablation study. In this setup, DFT replaces average pooling for seasonal trend decomposition, KAN(GRBF)-F uses the GRBF enhanced KAN module for feature fusion instead of MLP, KAN(GRBF)-P applies the GRBF enhanced KAN module for time series prediction, and AdaptiveLoss replaces MSELoss with an adaptive loss function.

Fig. 11 illustrates the performance improvements achieved by incorporating each module. Experimental results show that each module independently enhances prediction accuracy. A comparison of combinations #2 with #1 shows that DFT, by converting time-domain data to the frequency domain, better captures periodic patterns and reduces high-frequency noise, flexibly handling multiple periodic signals while retaining more details, thus improving prediction performance. Comparing combination #3 with #2 reveals that the KAN(GRBF)-F module efficiently extracts complex time-series features from multi-scale fused data, resulting in higher prediction accuracy. Further analysis of combination #4 and #3 shows that combining KAN(GRBF)-F and KAN(GRBF)-P yields even greater improvements in prediction performance. Lastly, the comparison between combination #5 and #4 demonstrates that introducing the adaptive loss function enhances the model's robustness, effectively reducing the impact of outliers and accommodating various error distributions. This significantly improves prediction accuracy under different weather conditions. The combined use of all four modules achieves optimal predictive performance. Compared to the combination #1, the overall improvements in MSE, MAE, nRMSE, and R^2 are 7.0%, 12.3%, 3.6%, and 9.6%, respectively. These results confirm that the synergistic effects of the modules significantly enhance the model's overall predictive capabilities, validating the soundness of the proposed approach.

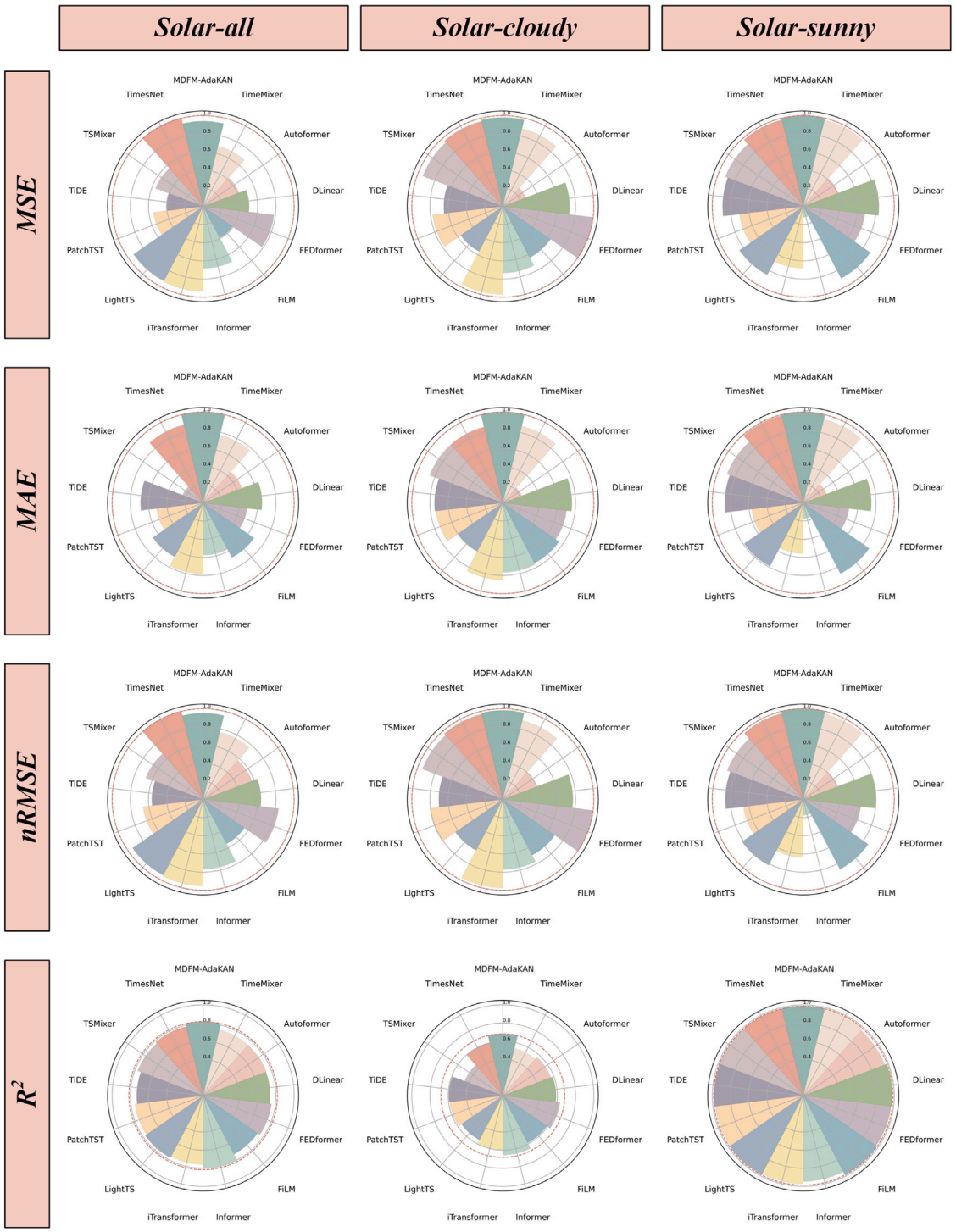

Fig. 7. Stacked polar chart of evaluation metrics under different weather conditions.

5.5. Robustness analysis

To assess the robustness of the proposed method, experiments were conducted with five different random seeds while keeping hyperparameter settings consistent. For each dataset, the results of four evaluation metrics are reported in Table 6 as mean ± standard deviation across the five runs. To quantitatively assess the model's stability under repeated trials, a metric termed Stability Score is defined in Eq. (21), where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and standard deviation of each evaluation metric across the repeated experiments. A higher Stability Score indicates better performance consistency and model robustness. As shown in the results, the proposed method achieves a Stability Score exceeding 95% on all datasets, demonstrating reliable stability.

$$\left(1-\frac{1}{4}\left(\frac{\sigma_{\mathrm{MSE}}}{\mu_{\mathrm{MSE}}}+\frac{\sigma_{\mathrm{MAE}}}{\mu_{\mathrm{MAE}}}+\frac{\sigma_{\mathrm{nRMSE}}}{\mu_{\mathrm{nRMSE}}}+\frac{\sigma_{R^2}}{\mu_{R^2}}\right)\right)\times 100\% \tag{21}$$

6. Conclusion

In this study, a robust and efficient MTS forecasting method named MDFM-AdaKAN is proposed, which demonstrates favorable predictive

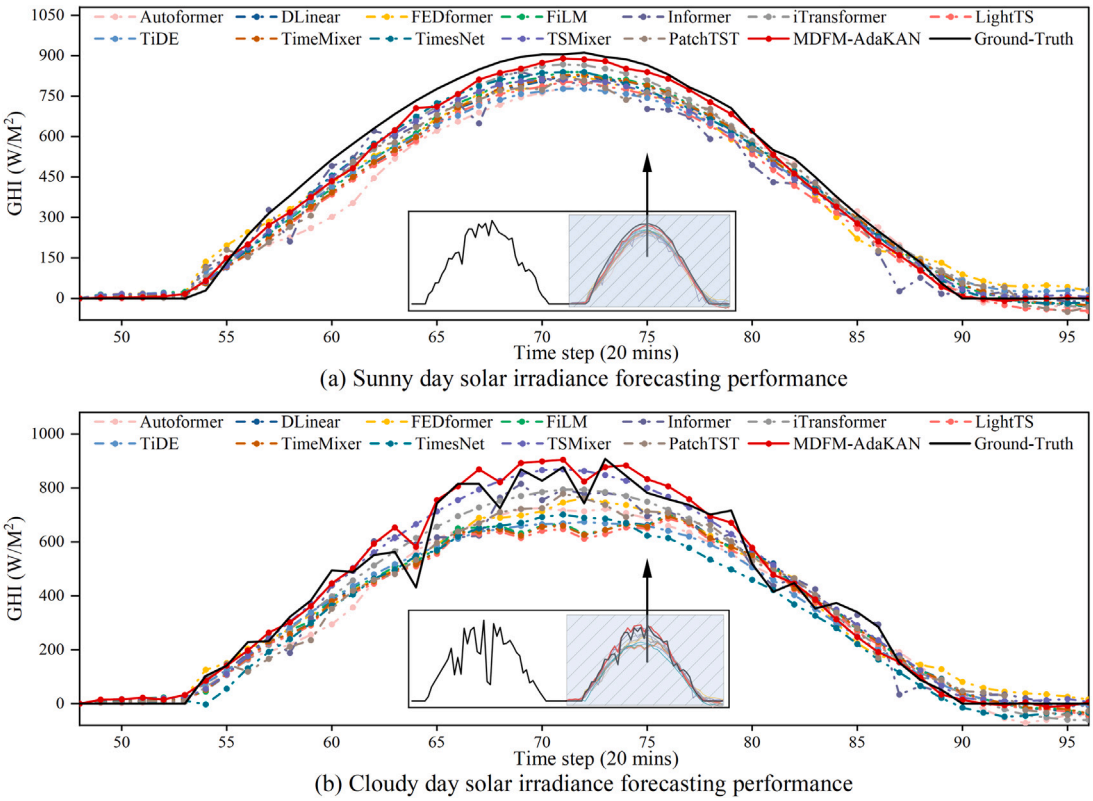

Fig. 8. Prediction curves for the next day under sunny and cloudy conditions.

Table 4
Computational efficiency analysis.

Method	MAE	GPU memory(MiB)	Parameter	Test Time (s)	Metrics rank
MDFM-AdaKAN	0.216	102	56 159	1.894	(1,7,6,10)
MDFM-KAN	0.259	350	137 126	25.511	(7,14,10,14)
Autoformer	0.287	148	89 497	1.362	(13,9,8,8)
DLinear	0.262	17	4704	0.091	(9,1,1,1)
FEDformer	0.283	148	179 609	3.532	(12,9,11,13)
FiLM	0.258	226	9437 238	2.670	(6,13,14,12)
Informer	0.272	149	98 313	0.582	(10,11,9,7)
iTransformer	0.245	146	44 432	0.211	(3,8,5,4)
LightTS	0.259	17.9	5868	0.131	(7,2,2,3)
PatchTST	0.280	23.4	57 680	0.227	(11,5,7,5)
TiDE	0.257	19.7	213 615	1.895	(5,4,12,11)
TSMixer	0.316	18.2	21 442	0.109	(14,3,3,2)
TimesNet	0.231	157	1183 929	1.462	(2,12,13,9)
TimeMixer	0.246	69.3	22 743	0.553	(4,6,4,6)

Table 5
Ablation study analysis: impact of each module on prediction accuracy.

Combination	DFT	KAN(GRBF)-F	KAN(GRBF)-P	AdaptiveLoss	MSE	MAE	nRMSE	R^2
#1	×	×	×	×	0.1732	0.2463	0.1180	0.7448
#2	✓	×	×	×	0.1711	0.2431	0.1173	0.7534
#3	✓	✓	×	×	0.1698	0.2397	0.1169	0.7658
#4	✓	✓	✓	×	0.1652	0.2268	0.1153	0.7943
#5	✓	✓	✓	✓	0.1610	0.2159	0.1138	0.8161

Table 6
Robustness analysis of different datasets.

Datasets	MSE	MAE	nRMSE	R^2	Stability score
Solar-all	0.161 ± 0.004	0.216 ± 0.006	0.114 ± 0.001	0.816 ± 0.009	98.8%
Solar-sunny	0.011 ± 0.000	0.075 ± 0.002	0.037 ± 0.001	0.986 ± 0.009	97.5%
Solar-cloudy	0.206 ± 0.003	0.275 ± 0.012	0.125 ± 0.001	0.581 ± 0.007	98.2%
ETTh1	0.364 ± 0.003	0.388 ± 0.002	0.069 ± 0.003	0.659 ± 0.003	99.4%
ETTh2	0.290 ± 0.012	0.338 ± 0.010	0.067 ± 0.013	0.807 ± 0.008	97.5%
ETTm1	0.308 ± 0.002	0.344 ± 0.003	0.062 ± 0.001	0.678 ± 0.001	99.2%
ETTm2	0.174 ± 0.002	0.254 ± 0.003	0.051 ± 0.001	0.876 ± 0.001	99.3%

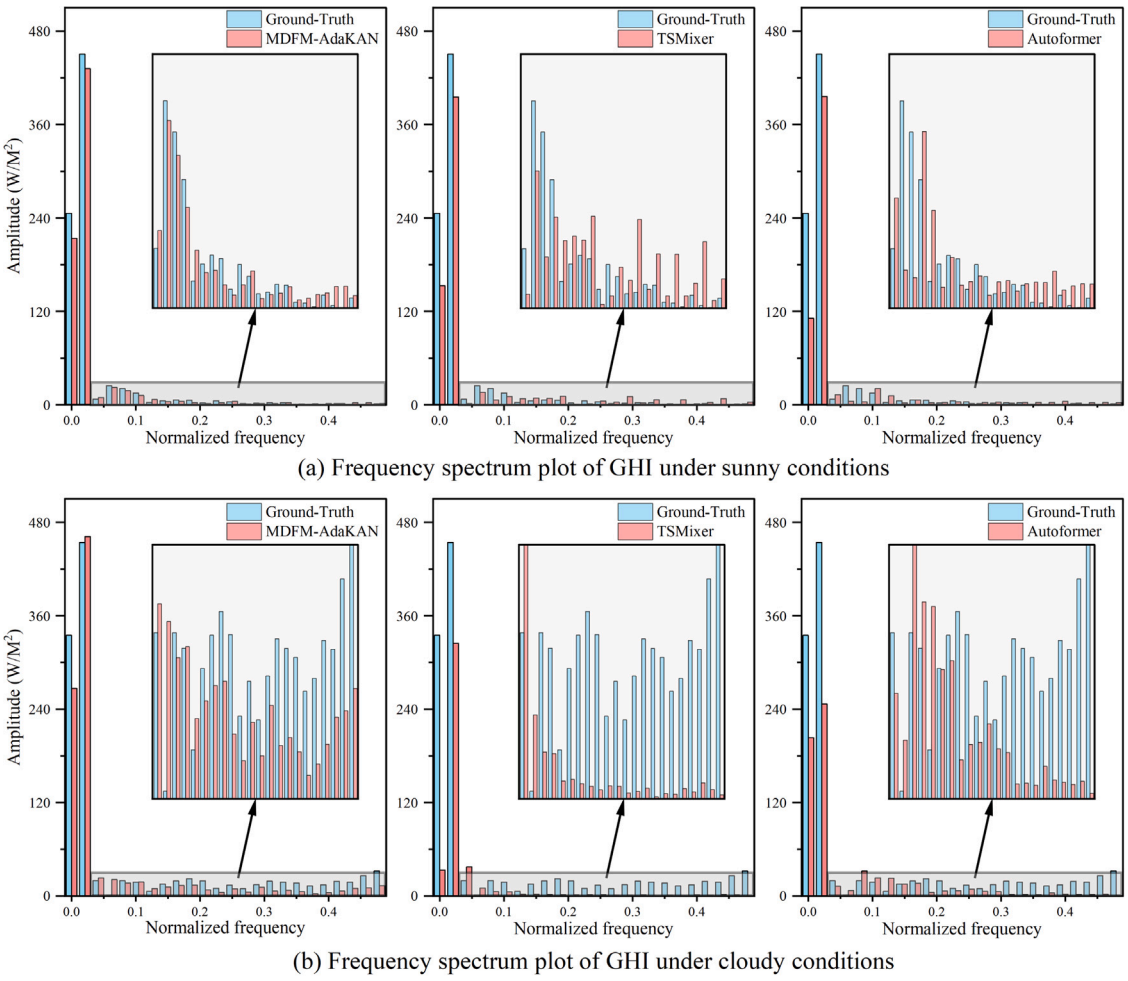

Fig. 9. Frequency spectra plot under different weather conditions.

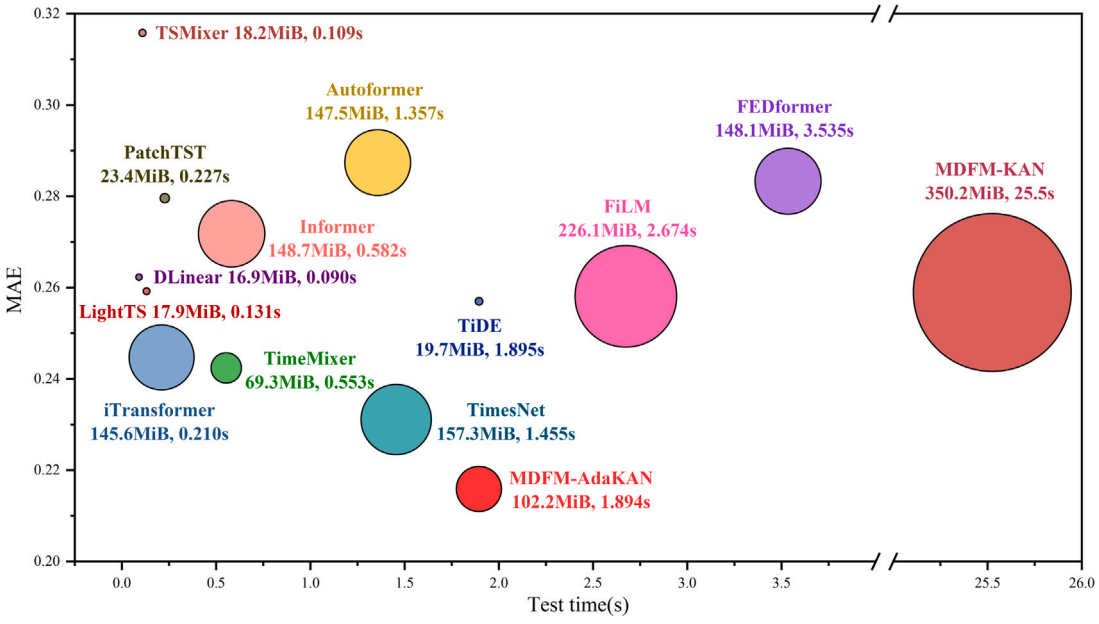

Fig. 10. Model efficiency comparison under Solar-all dataset.

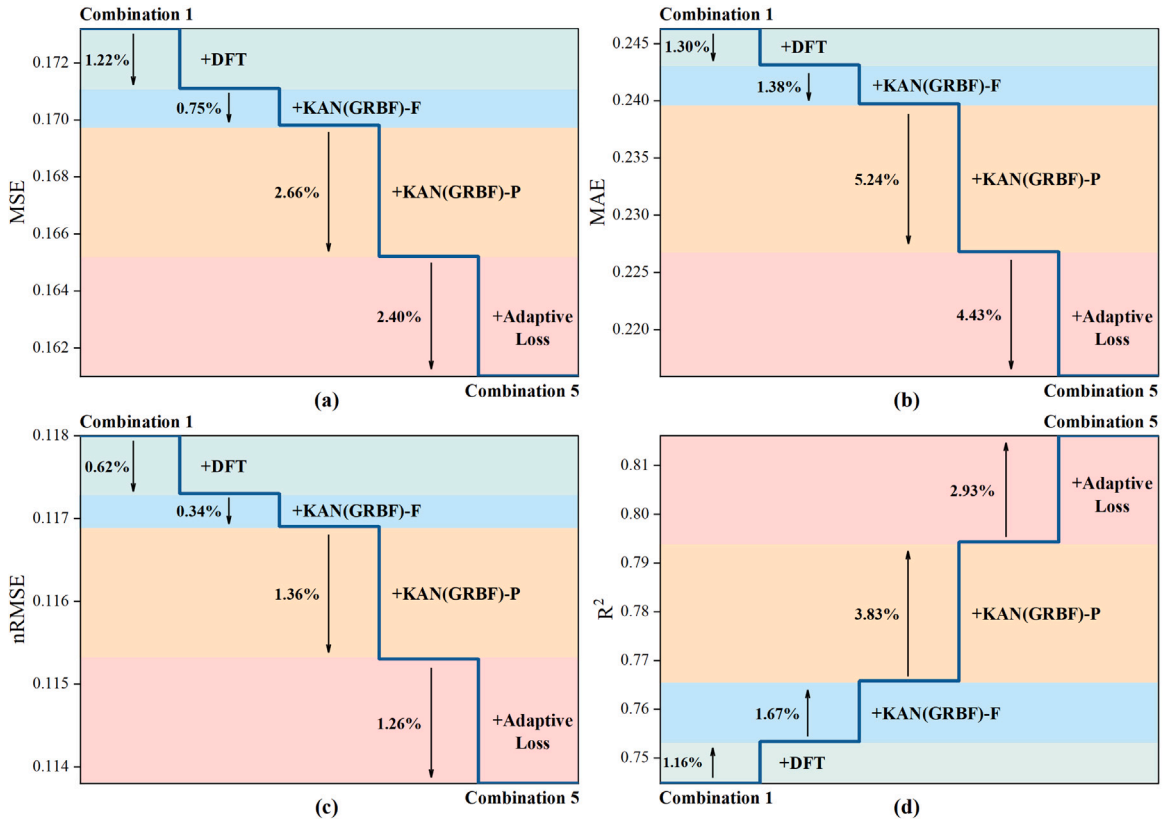

Fig. 11. Improvement achieved after adding each module.

performance under various data scenarios. The proposed method innovatively adopts different multi-scale data fusion strategies based on the importance of seasonal and trend components at different temporal scales. Improvements are made to the basis functions and loss function of the original KAN model, enhancing its capability in temporal feature extraction and prediction accuracy, thereby improving the real-time performance and robustness of the KAN model for large-scale tasks. Comprehensive comparative experiments were conducted on diverse datasets, evaluating the proposed model from the perspectives of data fusion, accuracy, interpretability, generalizability, and computational efficiency, highlighting the following conclusions:

Data Fusion. Effective data fusion methods contribute to improved prediction accuracy. MTS data exhibit distinct temporal characteristics at different temporal scales, which become more pronounced when decomposed into seasonal and trend components. We found that selecting appropriate downsampling scales and employing effective data fusion strategies enables the extraction of richer temporal features, leading to superior predictive performance. However, excessive downsampling may lead to the loss of essential details and increase computational complexity.

Accuracy. The proposed method demonstrated the highest overall accuracy. Four metrics were used to assess the algorithm's performance. We observed that MSE performed slightly weaker results on the solar-cloudy dataset. This can be attributed to the presence of numerous outliers caused by frequent cloud layer changes, as MSE is sensitive to such anomalies, resulting in larger error values. In contrast, MAE and R^2 provided stronger penalties for outliers, indicating that the proposed method demonstrates effective overall error control, albeit with minor limitations when dealing with outliers. Nonetheless, it consistently surpasses other baseline methods in performance.

Interpretability. The integration of Kolmogorov-Arnold theory significantly enhances the model's interpretability. As demonstrated by the experimental results in Section 5.2, comparing the amplitude of spectral components at different frequencies facilitates the evaluation of the model's ability to capture both low-frequency and high-frequency signals. Compared to other methods, the proposed approach not only effectively captures long-term trends but also more accurately detects high-frequency details, resulting in superior performance under abnormal data fluctuations conditions.

Generalizability. The generalizability of the model reflects its learning and adaptation capabilities. The proposed method demonstrated consistent optimal results under diverse data scenarios and proved applicable to other time-series forecasting tasks, highlighting strong generalization capabilities.

Computational Efficiency. The proposed method exhibited moderate computational efficiency. KAN-based models typically experience higher computational costs due to the significant overhead of learnable activation functions and limited parallel processing capabilities. However, this research found that appropriate tuning of the basis and loss functions could significantly improve computational efficiency while maintaining high accuracy.

In summary, the proposed method provides robust and efficient MTS forecasting results. It effectively addresses the instability and inefficiency resulting from the intermittent nature of time series data. This novel and effective solution offers significant potential for high-accuracy time series forecasting.

7. Limitations

We have identified some limitations in the current algorithm that need to be addressed in future work. Firstly, although AdaKAN has significantly improved computational efficiency compared to KAN and is now comparable to attention-based models, there remains a noticeable gap when compared to MLP-based frameworks. Secondly, the current data fusion approach, which integrates features from the first two downsampled scales, may not be suitable for all prediction interval

Table A.1
Evaluation of time series forecasting models under varying prediction horizons.

	Solar-sunny					ETTh1				
	Methods	MSE	MAE	nRMSE	R^2	Methods	MSE	MAE	nRMSE	R^2
Forecast half cycle	MDFM-AdaKAN	0.0085	0.0665	0.0318	0.9697	MDFM-AdaKAN	0.3299	0.3689	0.0657	0.6719
	iTransformer	0.0313	0.1424	0.0610	0.8438	TiDE	0.3434	0.3703	0.0670	0.6655
	Informer	0.0876	0.2155	0.1022	0.7095	PatchTST	0.3367	0.3760	0.0663	0.6739
	TimesNet	0.0110	0.0759	0.0363	0.9549	iTransformer	0.3494	0.3832	0.0676	0.6599
	TimeMixer	0.0105	0.0733	0.0353	0.9624	TimeMixer	0.3774	0.4148	0.0711	0.6365
Forecast one cycle	MDFM-AdaKAN	0.0113	0.0754	0.0368	0.9864	MDFM-AdaKAN	0.3643	0.3880	0.0690	0.6592
	iTransformer	0.0222	0.1137	0.0515	0.9697	TiDE	0.3847	0.3925	0.0709	0.6432
	Informer	0.0414	0.1471	0.0702	0.9443	PatchTST	0.3771	0.3969	0.0702	0.6497
	TimesNet	0.0120	0.0763	0.0378	0.9844	iTransformer	0.3950	0.4098	0.0718	0.6331
	TimeMixer	0.0121	0.0800	0.0380	0.9851	TimeMixer	0.4178	0.4478	0.0739	0.6081
Forecast two cycles	MDFM-AdaKAN	0.0204	0.1011	0.0494	0.9764	MDFM-AdaKAN	0.4139	0.4194	0.0735	0.6226
	iTransformer	0.0289	0.1300	0.0587	0.9617	TiDE	0.4389	0.4252	0.0757	0.5994
	Informer	0.0476	0.1612	0.0753	0.9431	PatchTST	0.4239	0.4278	0.0744	0.6151
	TimesNet	0.0268	0.1188	0.0565	0.9662	iTransformer	0.4486	0.4412	0.0766	0.5900
	TimeMixer	0.0216	0.1071	0.0507	0.9744	TimeMixer	0.4800	0.4891	0.0802	0.5661
Forecast three cycles	MDFM-AdaKAN	0.0268	0.1162	0.0565	0.9695	MDFM-AdaKAN	0.4412	0.4377	0.0759	0.5982
	iTransformer	0.0470	0.1665	0.0749	0.9420	TiDE	0.4716	0.4415	0.0785	0.5690
	Informer	0.0855	0.2163	0.1010	0.9021	PatchTST	0.4694	0.4544	0.0783	0.5703
	TimesNet	0.0394	0.1398	0.0685	0.9545	iTransformer	0.4852	0.4618	0.0796	0.5555
	TimeMixer	0.0300	0.1254	0.0598	0.9637	TimeMixer	0.5372	0.5238	0.0848	0.5212
Forecast four cycles	MDFM-AdaKAN	0.0355	0.1333	0.0651	0.9602	MDFM-AdaKAN	0.4603	0.4443	0.0775	0.5814
	iTransformer	0.0705	0.2028	0.0917	0.9128	TiDE	0.4876	0.4494	0.0798	0.5552
	Informer	0.0922	0.2246	0.1048	0.8983	PatchTST	0.4767	0.4593	0.0789	0.5667
	TimesNet	0.0393	0.1407	0.0684	0.9537	iTransformer	0.5037	0.4723	0.0811	0.5403
	TimeMixer	0.0436	0.1482	0.0721	0.9462	TimeMixer	0.5691	0.5461	0.0873	0.4916

lengths. In future work, an adaptive downsampling method will be developed to enable the model to automatically select the optimal number of downsampling scales based on dataset characteristics, enhancing prediction accuracy across diverse data scenarios and prediction interval lengths.

CRediT authorship contribution statement

Jian Liu: Methodology, Writing – original draft. **Fan Yang:** Data curation, Formal analysis. **Ke Yan:** Visualization, Writing – original draft, Project administration, Supervision.

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.

Acknowledgment

This work is supported by National Science Foundation of China Excellent Young Overseas Talents Programme under Grant No. Z2023043 92531 and The Science and Technology Innovation Program (Overseas Expert Program) of Hunan Province under Grant No. 2024RC9006.

Appendix A. Multi-horizon forecasting analysis

This section presents an experimental analysis across multiple forecasting horizons to evaluate the effectiveness of the proposed method. To ensure fairness and comparability, all experiments in this section are conducted using the same model architecture and hyperparameter settings as in the main experiment. In addition, a fixed historical input length of one full cycle is used for each task: 48 time steps for solar irradiance forecasting and 96 time steps for electricity transformer temperature forecasting. The forecasting horizons are set as 0.5, 1, 2, 3, and 4 multiples of the respective cycle lengths, covering a range of forecasting requirements from short-term to long-term. Experiments were conducted on the Solar-sunny and ETTh1 datasets, with the results summarized in Table A.1.

As presented in Table A.1, the prediction errors of all methods tend to increase with longer forecasting horizons. However, the proposed method consistently achieves better predictive accuracy across all forecasting horizons on both datasets, indicating robust adaptability to forecasting tasks of varying temporal horizons. Further analysis reveals that on the Solar-sunny dataset, forecasting over a half-cycle horizon generally results in slightly inferior performance across all methods relative to a full-cycle forecast. This phenomenon can be attributed to the strongly periodic nature of the Solar-sunny dataset. When forecasting a full cycle, the periodic patterns in the time series are more effectively preserved within the output window, enabling the model to better learn and generalize the underlying temporal regularities. In contrast, a horizon that does not span an entire cycle limits the model's ability to represent periodic components, thereby reducing prediction accuracy. In the case of the ETTh1 dataset, which exhibits weaker periodicity, the model is exposed to more complex and irregular temporal dynamics. Consequently, longer forecasting horizons lead to higher prediction errors compared to those observed under shorter horizons.

Appendix B. Analysis of the Gaussian radial basis function

This section presents an in-depth analysis of GRBF. Under consistent hyperparameter settings, predictive accuracy and computational efficiency were assessed for GRBF and compared with those of B-spline basis functions across multiple datasets. The experimental results are summarized in Table B.1. Due to an out-of-memory issue encountered during training with B-spline basis functions on the ETTh2 dataset, the corresponding result is represented as "–" to signify a missing entry. The results demonstrate that replacing B-spline basis functions with GRBF leads to substantial improvements in predictive accuracy. Moreover, the inference time on the test set and the number of model parameters are considerably reduced. These findings highlight the effectiveness of GRBF in enhancing prediction performance while reducing computational cost.

The configuration of the GRBF is primarily determined by three key hyperparameters: the number of center vectors N, and the distribution range of the center vectors, defined by GridMin and GridMax. The parameter N controls the resolution of the function approximation by specifying the number of Gaussian radial basis functions used per input

Table B.1
Performance comparison of GRBF and B-spline basis function across multiple datasets.

	Datasets	MSE	MAE	nRMSE	R^2	Test time	Parameter
GRBF	solar-all	0.1610	0.2159	0.1138	0.8161	1.8940	56 159
	solar-sunny	0.0113	0.0754	0.0368	0.9864	0.3968	59 903
	solar-cloudy	0.2056	0.2754	0.1249	0.6806	1.8885	85 303
	ETTh1	0.3643	0.3880	0.0690	0.6592	1.5244	305 473
	ETTh2	0.2903	0.3385	0.0672	0.8067	3.4225	417 473
	ETTm1	0.3084	0.3438	0.0625	0.6799	6.2198	305 473
	ETTm2	0.1741	0.2535	0.0514	0.8764	4.5129	249 457
B-spline	solar-all	0.2066	0.2792	0.1289	0.7353	25.511	137 126
	solar-sunny	0.0194	0.1047	0.0480	0.9761	2.1786	89 798
	solar-cloudy	0.2370	0.3015	0.1341	0.5655	12.665	127 246
	ETTh1	0.3883	0.4004	0.0712	0.6448	10.774	432 536
	ETTh2	–	–	–	–	–	–
	ETTm1	0.3149	0.3471	0.0631	0.6751	44.143	432 536
	ETTm2	0.2236	0.3029	0.0582	0.8397	32.473	376 552

Fig. B.1. Heatmaps of evaluation metrics on the Solar-sunny dataset under different GRBF hyperparameter settings.

dimension. GridMin and GridMax define the lower and upper bounds for the distribution of the N center vectors, which specify the effective region of the basis functions in the input space. To intuitively demonstrate the impact of these hyperparameters on model performance, the Solar-sunny dataset was selected as a case study. Corresponding heatmaps of multiple evaluation metrics under varying hyperparameter settings are shown in Fig. B.1.

The experimental results exhibit a non-monotonic trend in prediction error with respect to N: as N increases, the error initially decreases, reaching a minimum at $N = 8$, followed by an increase at higher resolutions. This behavior suggests that using a moderate number of radial basis functions improves the model's ability to capture nonlinear relationships, while an excessive number can lead to overfitting and reduced generalization. Moreover, the inference time on the test set increases monotonically with N, indicating a consistent rise in computational cost associated with increased model complexity.

Under the condition of a fixed N, the effects of the interval boundaries defined by GridMin and GridMax on model performance were further investigated. The results indicate that MAE and MSE are more sensitive to changes in the grid range, whereas R^2 and test-time remain comparatively stable. None of the evaluated metrics exhibit a

Table C.1
Comparison of AdaptiveLoss and MSELoss across multiple datasets.

	Datasets	alpha(α)	Scale(c)	MSE	MAE	nRMSE	R^2
AdaptiveLoss	solar-all	0.2909	0.0744	0.1610	0.2159	0.1138	0.8161
	solar-sunny	1.9864	0.1637	0.0113	0.0754	0.0368	0.9864
	solar-cloudy	0.1821	0.1186	0.2056	0.2754	0.1249	0.6806
	ETTh1	0.9904	0.3283	0.3643	0.3880	0.0690	0.6592
	ETTh2	1.1483	0.3355	0.2903	0.3385	0.0672	0.8067
	ETTm1	0.3675	0.1705	0.3084	0.3438	0.0625	0.6799
	ETTm2	0.2081	0.1118	0.1741	0.2535	0.0514	0.8764
MSELoss	solar-all	–	–	0.1652	0.2268	0.1153	0.7943
	solar-sunny	–	–	0.0173	0.0958	0.0454	0.9791
	solar-cloudy	–	–	0.2189	0.3122	0.1289	0.5230
	ETTh1	–	–	0.3736	0.3986	0.0699	0.6507
	ETTh2	–	–	0.4480	0.4489	0.0835	0.6856
	ETTm1	–	–	0.3257	0.3667	0.0642	0.6600
	ETTm2	–	–	0.1924	0.2777	0.0540	0.8644

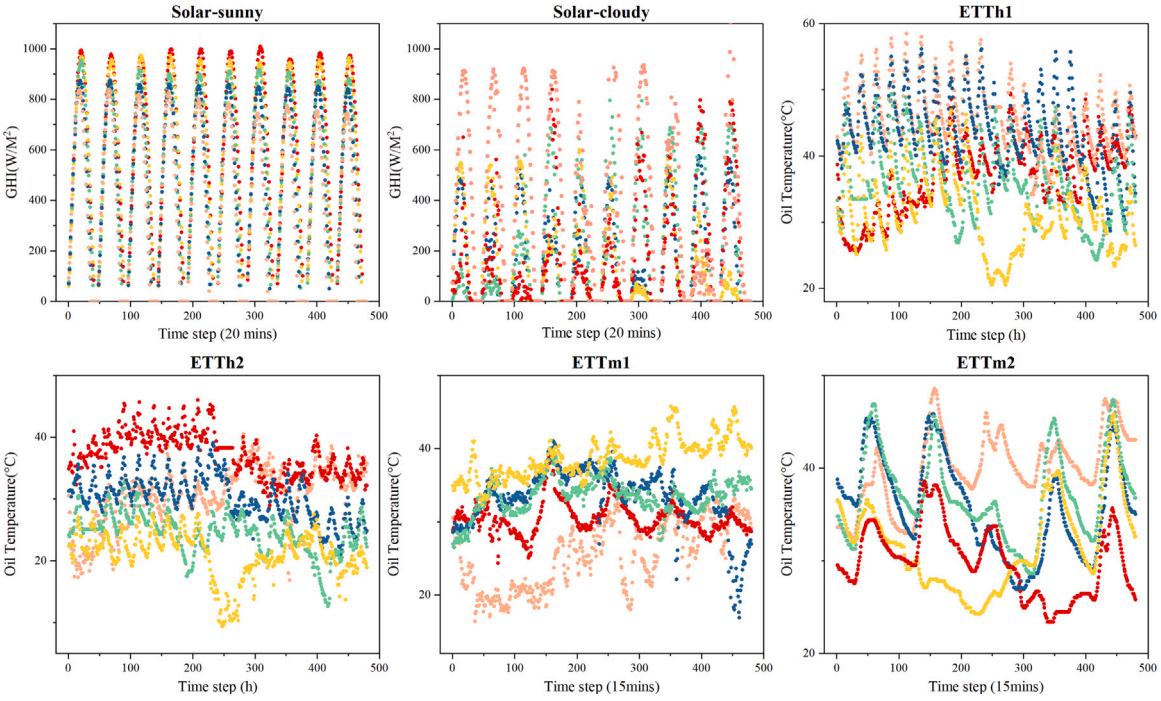

Fig. C.1. Visualization of target variable distributions in different datasets. To illustrate the distributional characteristics of the target variables, the data are organized into sample groups, each consisting of 480 consecutive observations. Due to the large volume of data, five sample groups are randomly selected for visualization, with each sequence plotted in a distinct color to provide a clearer visual presentation.

monotonic relationship with respect to variations in the interval. Therefore, when applying GRBF to different datasets, it is crucial to ensure that the majority of the input data distribution falls within the grid range defined by GridMin and GridMax. Under this condition, hyperparameter tuning can be conducted to identify an optimal configuration that strikes a balance between improved predictive accuracy and manageable computational cost.

Appendix C. Analysis of the adaptive loss function

This section presents a comprehensive analysis of AdaptiveLoss, including its impact on predictive performance across different datasets and the tendency of its learned tunable parameters under varying data distributions. Notably, the scale parameter c is inherently influenced by the magnitude and statistical distribution of the residuals. In the absence of global normalization across datasets, the learned values of c are not directly comparable. Furthermore, since the shape parameter α and the scale parameter c are jointly optimized during training, a coupling relationship exists between them. Consequently, rather than interpreting the absolute values of α, it is treated as a relative indicator of the model's robustness adjustment under varying data conditions.

Table C.1 summarizes the experimental results obtained by replacing the MSELoss with the adaptive loss function across various datasets. Since the MSELoss does not involve learnable parameters α and c, the corresponding entries in the table are denoted by "–". Compared to MSELoss, AdaptiveLoss consistently achieves improved performance across all datasets, demonstrating its effectiveness in modeling diverse data distributions. Fig. C.1 presents the visualization of target variables from different datasets, which highlight their respective distributional characteristics. A joint analysis of these visualizations and the corresponding learned α values provides deeper insights into how AdaptiveLoss adaptively modulates the shape of its loss function in response to varying data distributions, thereby enhancing model robustness or fitting precision as appropriate.

In the solar irradiance forecasting task, the Solar-sunny dataset exhibits a relatively stable temporal pattern with minimal outliers. Under such low-noise conditions, the model tends to learn a relatively large α value that is close to the standard L2 loss. This indicates that the model preserves a stronger capacity for data fitting in order to achieve

improved predictive accuracy. Conversely, the Solar-cloudy dataset contains a substantial number of outliers. In this case, the learned α value is relatively small and aligns more closely with the properties of the Cauchy loss, thereby effectively attenuating the influence of outliers on gradient updates and enhancing model robustness.

A similar trend is observed in the electricity transformer temperature forecasting task. The ETTm1 and ETTm2 datasets, characterized by higher temporal resolution, exhibit frequent short-term disturbances and irregular fluctuations. To mitigate the impact of such fluctuations on performance, the model tends to learn relatively small values of α. In contrast, the ETTh1 and ETTh2 datasets possess smoother temporal dynamics due to their lower resolution, with the resulting learned α values approximating the Charbonnier loss. This behavior reflects a favorable balance between robustness and predictive accuracy. These observations align with the theoretical analysis in Section 3.2.3 and further support the effectiveness of AdaptiveLoss in enhancing model robustness and adaptability to varying data distributions in multivariate time series forecasting.

Data availability

The source code and testing data are publicly available in https://github.com/liujian123223/MDFM-AdaKAN.

References

[1] V. Riabchuk, L. Hagel, F. Germaine, A. Zharova, Utility-based context-aware multi-agent recommendation system for energy efficiency in residential buildings, Inf. Fusion 112 (2024) 102559, http://dx.doi.org/10.1016/j.inffus.2024.102559.

[2] J.-H. Syu, J.C.-W. Lin, P.S. Yu, Multi-head learning models for power consumption prediction of unmanned ground vehicles, Inf. Fusion 118 (2025) 102895, http://dx.doi.org/10.1016/j.inffus.2024.102895.

[3] J. Liu, F. Yang, K. Yan, L. Jiang, Household energy consumption forecasting based on adaptive signal decomposition enhanced iTransformer network, Energy Build. 324 (2024) 114894, http://dx.doi.org/10.1016/j.enbuild.2024.114894.

[4] N. Jin, F. Yang, Y. Mo, Y. Zeng, X. Zhou, K. Yan, X. Ma, Highly accurate energy consumption forecasting model based on parallel LSTM neural networks, Adv. Eng. Inf. 51 (2022) 101442, http://dx.doi.org/10.1016/j.aei.2021.101442.

[5] S. Almaghrabi, M. Rana, M. Hamilton, M. Saiedur Rahaman, Multivariate solar power time series forecasting using multilevel data fusion and deep neural networks, Inf. Fusion 104 (2024) 102180, http://dx.doi.org/10.1016/j.inffus.2023.102180.

[6] H. Wen, Y. Du, X. Chen, E. Lim, H. Wen, L. Jiang, W. Xiang, Deep learning based multistep solar forecasting for PV ramp-rate control using sky images, IEEE Trans. Ind. Inf. 17 (2) (2021) 1397–1406, http://dx.doi.org/10.1109/TII.2020.2987916.

[7] F. Mo, X. Jiao, X. Li, Y. Du, Y. Yao, Y. Meng, S. Ding, A novel multi-step ahead solar power prediction scheme by deep learning on transformer structure, Renew. Energy 230 (2024) 120780, http://dx.doi.org/10.1016/j.renene.2024.120780.

[8] V. Plakandaras, T. Papadimitriou, P. Gogas, Forecasting transportation demand for the U.S. market, Transp. Res. Part A Policy Pr. 126 (2019) 195–214, http://dx.doi.org/10.1016/j.tra.2019.06.008.

[9] A. Babii, E. Ghysels, J. Striaukas, Machine learning time series regressions with an application to nowcasting, J. Bus. Econom. Statist. 40 (3) (2022) 1094–1106, http://dx.doi.org/10.1080/07350015.2021.1899933.

[10] M. Langtry, V. Wichitwechkarn, R. Ward, C. Zhuang, M.J. Kreitmair, N. Makasis, Z. Xuereb Conti, R. Choudhary, Impact of data for forecasting on performance of model predictive control in buildings with smart energy storage, Energy Build. 320 (2024) 114605, http://dx.doi.org/10.1016/j.enbuild.2024.114605.

[11] D. Zhuang, V.J. Gan, Z. Duygu Tekler, A. Chong, S. Tian, X. Shi, Data-driven predictive control for smart HVAC system in IoT-integrated buildings with time-series forecasting and reinforcement learning, Appl. Energy 338 (2023) 120936, http://dx.doi.org/10.1016/j.apenergy.2023.120936.

[12] N. Bhoj, R. Singh Bhadoria, Time-series based prediction for energy consumption of smart home data using hybrid convolution-recurrent neural network, Telemat. Inform. 75 (2022) 101907, http://dx.doi.org/10.1016/j.tele.2022.101907.

[13] R. Hou, Q. Liu, P. He, Y. Liu, Y. Huang, J. Xie, Y. Xie, T. Dai, DPEM: Dual-perspective enhanced mamba for multivariate time series forecasting, Inf. Fusion 123 (2025) 103250, http://dx.doi.org/10.1016/j.inffus.2025.103250.

[14] M. Germán-Morales, A. Rivera-Rivas, M. del Jesus Díaz, C. Carmona, Transfer learning with foundational models for time series forecasting using low-rank adaptations, Inf. Fusion 123 (2025) 103247, http://dx.doi.org/10.1016/j.inffus.2025.103247.

[15] Q. Guo, L. Fang, R. Wang, C. Zhang, Multivariate time series forecasting using multiscale recurrent networks with scale attention and cross-scale guidance, IEEE Trans. Neural Netw. Learn. Syst. 36 (1) (2025) 540–554, http://dx.doi.org/10.1109/TNNLS.2023.3326140.

[16] R. Zhang, Y. Hao, Time series prediction based on multi-scale feature extraction, Math. 12 (7) (2024) 973, http://dx.doi.org/10.3390/math12070973.

[17] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, M. Long, TimesNet: Temporal 2D-variation modeling for general time series analysis, in: 11th Int. Conf. Learn. Represent., 2023, http://dx.doi.org/10.48550/arXiv.2210.02186.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, 2017, pp. 6000–6010.

[19] C. Zhu, X. Song, Y. Li, S. Deng, T. Zhang, A spatial-frequency dual-domain implicit guidance method for hyperspectral and multispectral remote sensing image fusion based on Kolmogorov–Arnold network, Inf. Fusion 123 (2025) 103261, http://dx.doi.org/10.1016/j.inffus.2025.103261.

[20] A.N. Kolmogorov, On the Representation of Continuous Functions of Several Variables by Superpositions of Continuous Functions of a Smaller Number of Variables, Am. Math. Soc., 1961.

[21] A.N. Kolmogorov, On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition, in: Dokl. Akad. Nauk, vol. 114, (5) Russian Academy of Sciences, 1957, pp. 953–956.

[22] B.C. Koenig, S. Kim, S. Deng, KAN-ODEs: Kolmogorov-Arnold network ordinary differential equations for learning dynamical systems and hidden physics, Comput. Methods Appl. Mech. Engrg. 432 (2024) 117397, http://dx.doi.org/10.1016/j.cma.2024.117397.

[23] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljacic, T.Y. Hou, M. Tegmark, KAN: Kolmogorov–Arnold networks, in: ICLR 2025, 2025, http://dx.doi.org/10.48550/arXiv.2404.19756.

[24] A. Jamali, S.K. Roy, D. Hong, B. Lu, P. Ghamisi, How to learn more? Exploring Kolmogorov-Arnold networks for hyperspectral image classification, Remote. Sens. 16 (21) (2024) http://dx.doi.org/10.3390/rs16214015.

[25] B. Lim, S. Zohren, Time-series forecasting with deep learning: a survey, Philos. Trans. R. Soc. A 379 (2194) (2021) 20200209, http://dx.doi.org/10.1098/rsta.2020.0209.

[26] P. Sen, M. Roy, P. Pal, Application of ARIMA for forecasting energy consumption and GHG emission: A case study of an Indian pig iron manufacturing organization, Energy 116 (2016) 1031–1038, http://dx.doi.org/10.1016/j.energy.2016.10.068.

[27] W. Zhong, D. Zhai, W. Xu, W. Gong, C. Yan, Y. Zhang, L. Qi, Accurate and efficient daily carbon emission forecasting based on improved ARIMA, Appl. Energy 376 (2024) 124232, http://dx.doi.org/10.1016/j.apenergy.2024.124232.

[28] C. Chen, Q. Chen, S. Yao, M. He, J. Zhang, G. Li, Y. Lin, Combining physical-based model and machine learning to forecast chlorophyll-a concentration in freshwater lakes, Sci. Total Environ. 907 (2024) 168097, http://dx.doi.org/10.1016/j.scitotenv.2023.168097.

[29] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297, http://dx.doi.org/10.1007/BF0099401.

[30] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32, http://dx.doi.org/10.1023/A:1010933404324.

[31] G. Liu, K. Zhong, H. Li, T. Chen, Y. Wang, A state of art review on time series forecasting with machine learning for environmental parameters in agricultural greenhouses, Inf. Process. Agric. 11 (2) (2024) 143–162, http://dx.doi.org/10.1016/j.inpa.2022.10.005.

[32] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324, http://dx.doi.org/10.1109/5.726791.

[33] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366, http://dx.doi.org/10.1016/0893-6080(89)90020-8.

[34] P. Wang, K. Wang, Y. Song, X. Wang, AutoLDT: a lightweight spatio-temporal decoupling transformer framework with AutoML method for time series classification, Sci. Rep. 14 (1) (2024) 29801, http://dx.doi.org/10.1038/s41598-024-81000-1.

[35] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: AAAI, 2021, pp. 11106–11115, http://dx.doi.org/10.1609/aaai.v35i12.17325.

[36] J. Yan, L. Mu, L. Wang, R. Ranjan, A.Y. Zomaya, Temporal convolutional networks for the advance prediction of ENSO, Sci. Rep. 10 (1) (2020) 8055, http://dx.doi.org/10.1038/s41598-020-65070-5.

[37] B. Qin, X. Gao, T. Ding, F. Li, D. Liu, Z. Zhang, R. Huang, A hybrid deep learning model for short-term load forecasting of distribution networks integrating the channel attention mechanism, IET Gener. Transm. Distrib. 18 (9) (2024) 1770–1784, http://dx.doi.org/10.1049/gtd2.13142.

[38] R. Rick, L. Berton, Energy forecasting model based on CNN-LSTM-AE for many time series with unequal lengths, Eng. Appl. Artif. Intell. 113 (2022) 104998, http://dx.doi.org/10.1016/j.engappai.2022.104998.

[39] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J.Y. Zhang, J. ZHOU, TimeMixer: Decomposable multiscale mixing for time series forecasting, in: Proc. 12th Int. Conf. Learn. Represent., 2024, http://dx.doi.org/10.48550/arXiv.2405.14616.

[40] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting? Proc. the AAAI Conf. Artif. Intell. 37 (9) (2023) 11121–11128, http://dx.doi.org/10.1609/aaai.v37i9.26317.

[41] Y. Chen, F. Ding, L. Zhai, Multi-scale temporal features extraction based graph convolutional network with attention for multivariate time series prediction, Expert Syst. Appl. 200 (2022) 117011, http://dx.doi.org/10.1016/j.eswa.2022.117011.

[42] F. Granata, S. Zhu, F. Di Nunno, Advanced streamflow forecasting for central European rivers: The cutting-edge Kolmogorov-Arnold networks compared to transformers, J. Hydrol. 645 (2024) 132175, http://dx.doi.org/10.1016/j.jhydrol.2024.132175.

[43] J.T. Barron, A general and adaptive robust loss function., in: CVPR, 2019, pp. 4331–4339.

[44] I.E. Livieris, C-KAN: A new approach for integrating convolutional layers with Kolmogorov-Arnold networks for time-series forecasting, Math. 12 (19) (2024) http://dx.doi.org/10.3390/math12193022.

[45] T. Quanwei, X. Guijun, X. Wenju, Cakformer: Transformer model for long-term heat load forecasting based on Cauto-correlation and KAN, Energy 324 (2025) 135460, http://dx.doi.org/10.1016/j.energy.2025.135460.

[46] H.T. Pedro, D.P. Larson, C.F. Coimbra, A comprehensive dataset for the accelerated development and benchmarking of solar forecasting methods, J. Renew. Sustain. Energy 11 (3) (2019) http://dx.doi.org/10.1063/1.5094494.

[47] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, Adv. Neural Inf. Process. Syst. 34 (2021) 22419–22430.

[48] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting, CoRR (2022) arXiv:2201.12740.

[49] T. Zhou, Z. Ma, X. Wang, Q. Wen, L. Sun, T. Yao, W. Yin, R. Jin, FiLM: Frequency improved Legendre memory model for long-term time series forecasting, in: NeurIPS, 2022.

[50] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, M. Long, iTransformer: Inverted transformers are effective for time series forecasting, CoRR (2023) http://dx.doi.org/10.48550/arXiv.2310.06625, arXiv:2310.06625.

[51] D. Campos, M. Zhang, B. Yang, T. Kieu, C. Guo, C.S. Jensen, LightTS: Lightweight time series classification with adaptive ensemble distillation, Proc. ACM Manag. Data 1 (2) (2023) 171:1–171:27, http://dx.doi.org/10.1145/3589316.

[52] Y. Nie, N.H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: Long-term forecasting with transformers, in: 11th Int. Conf. Learn. Represent., 2023, http://dx.doi.org/10.48550/arXiv.2211.14730.

[53] A. Das, W. Kong, A. Leach, S. Mathur, R. Sen, R. Yu, Long-term forecasting with TiDE: Time-series dense encoder, CoRR (2023) http://dx.doi.org/10.48550/arXiv.2304.08424, arXiv:2304.08424.

[54] S.-A. Chen, C.-L. Li, S.O. Arik, N.C. Yoder, T. Pfister, TSMixer: An all-MLP architecture for time series forecast-ing, Trans. Mach. Learn. Res. (2023) http://dx.doi.org/10.48550/arXiv.2303.06053.

[55] D. Chicco, M.J. Warrens, G. Jurman, The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation, Peerj Comput. Sci. 7 (2021) e623, http://dx.doi.org/10.7717/peerj-cs.623.