KANformer: A Flexible Kolmogorov-Arnold Transformer For Power Load Forecasting

Yang Yu[1], Zheng Yang[1,2*], Shanshan Lin[1], and Yue Zhang[1,2]

[1] *School of information and engineering, Shenyang University of Technology, Shenyang 110023, China*
[2] *College of Information, Shenyang Institute of Engineering, Shenyang 110136, China*
[*] *Corresponding author. E-mail: yangzheng@sie.edu.cn*

Received: Aug. 03, 2025; Accepted: Sep. 18, 2025

With economic growth and improving living standards, electricity demand becomes more complex and volatile. As a key part of power system planning, operation, and management, power load forecasting is of great importance. Accurate forecasting enables grid dispatching departments to make reasonable generation plans and schedule equipment maintenance in advance. However, there are still exist two issues in current power load forecasting methods: (1) Current methods commonly utilize multilayer perceptrons to construct the overall network which is extremely difficult to interpret how these models arrive at specific predictions. (2) They commonly utilize the one-step generation paradigm with a customized forecasting head. Such a manner ignores the temporal dependencies in the forecasting series and needs to train separately for different prediction lengths. To this end, a novel interpretable Kolmogorov-Arnold networks (KAN)-based Transformer architecture (KANformer)is proposed as the backbone of the model to capture variation patterns of power load time-series data. Specifically, KANformer transforms the forecasting task into a standard language modeling task. It uses patching technology to project time series into patch-based representations. During training, an autoregressive optimization function replaces the traditional single-step generation scheme. This allows the model to effectively model the temporal dependencies within the prediction range at the patch level through autoregressive inference. It can also seamlessly adapt to various power grid load datasets with different prediction settings without any modifications. Experimental results on two real-world power grid load datasets show that KANformer has superior performance and generalization ability.

Keywords: Time series analysis, power load forecasting, Kolmogorov-Arnold network.

http://dx.doi.org/10.6180/jase.202608_31.043

1. Introduction

With economic growth and improving living standards, electricity demand becomes more complex and volatile. As a key part of power system planning, operation, and management, grid load forecasting is of great importance [1–5]. Accurate forecasting enables grid dispatching departments to make reasonable generation plans and schedule equipment maintenance in advance. This ensures grid stability and prevents power outages caused by sudden load changes. For long - term grid planning, reliable load forecasting offers scientific support for transmission line construction and substation layout. It helps optimize grid resource allocation and cut construction and operation costs [6, 7]. Moreover, in the context of large-scale renewable energy integration, load forecasting facilitates the effective utilization of renewable energy. It also helps achieve energy-saving and emission-reduction goals, improves energy efficiency, and promotes the development of a smarter, greener, and more reliable power grid.

Current power load forecasting methods fall into two main categories: traditional statistical and modern intelli-

gent approaches. Traditional statistical methods involve time-series analysis and regression analysis [8–12]. Time-series analysis, like ARIMA, differencing makes the series stationary, and uses autoregression and moving average for fitting and forecasting. Regression analysis treats grid load as the dependent variable, with factors such as temperature and humidity as independent variables, building a model to describe their quantitative relationship, including linear and polynomial regression. Modern intelligent methods cover machine learning and deep learning. Machine learning's neural networks have strong nonlinear mapping ability, automatically learning complex input-output relationships in historical load data. Training multilayer BP neural networks on such data adjusts weights and thresholds for load forecasting. Support vector machines, based on statistical learning theory, find optimal hyperplanes or regression functions for predictions and are suited for small-sample, nonlinear forecasting tasks. Deep learning excels in handling large, complex load data. For instance, RNN structures like LSTM and GRU process time-series grid load data, capturing long-term dependencies and dynamic changes. CNNs can also combine with RNNs: convolutional layers extract local load data features, then RNNs model the time series, enhancing forecasting accuracy [13, 14].

While deep learning-based power load forecasting has demonstrated strong predictive accuracy, its practical adoption in critical grid operations faces two significant barriers rooted in model design limitations. First, the prevalent reliance on Multi-Layer Perceptron (MLP) architectures creates an inherent interpretability deficit. These complex 'black-box' models, with their intricate structures and numerous parameters, fail to provide actionable insights into how specific forecasts are generated. This opacity is operationally unacceptable. Grid operators require transparent reasoning to trust forecasts, especially during critical events (e.g., sudden load spikes threatening stability) or when justifying decisions to stakeholders. Without understanding why a model predicts a certain load, operators cannot confidently act on its output or diagnose potential errors, severely hindering its utility for reliable grid management. Second, the prediction heads in these deep learning models are typically fixed once trained. Whether it is the output layer or a specific forecasting module, their configurations are determined during the training phase and lack flexibility for adjustments afterward. This rigidity limits the models' ability to adapt to dynamic power system requirements. For example, if there is a need to change the forecasting horizon to meet new operational standards, the models cannot easily accommodate these changes without extensive retraining. Moreover, this inflexibility also affects scalability. As power grids evolve, integrate more renewable energy sources, or expand into new regions, the forecasting systems need to be scalable to handle increased data volumes and more complex forecasting scenarios. However, the fixed prediction heads make it difficult to extend the models' capabilities without significant modifications to the underlying architecture.

To this end, we introduce KANformer, an innovative model architecture that integrates interpretable Kolmogorov-Arnold Networks (KAN) with Transformer technology. This advanced framework is specifically designed to capture and analyze the intricate variation patterns inherent in power load time-series data. By transforming the forecasting task into a standard language modeling task, KANformer leverages patching technology to project time series into patch-based representations. This approach not only enhances the model's ability to capture local features but also reduces the impact of noise on the data. During the training phase, KANformer employs an autoregressive optimization function, which replaces the conventional single-step generation scheme. This enables the model to effectively model the temporal dependencies within the prediction range at the patch level through autoregressive inference. Furthermore, the adaptive nature of KANformer's design allows it to seamlessly adapt to various power grid load datasets with different prediction settings, without requiring any modifications to the model's architecture. This flexibility makes KANformer a versatile tool for power load forecasting, capable of handling a wide range of scenarios and data conditions. The experimental results on two real-world power grid load datasets have demonstrated that KANformer not only achieves superior performance but also exhibits strong generalization ability. This makes it a promising solution for addressing the challenges of power load forecasting in practical applications.

The key contributions of KANformer include:

- A novel KAN-based transformer is proposed for forecasting power load in an autoregressive manner, which can be suitable for various forecasting settings.
- An adaptive residual learning is devised via transforming the forecasting process into an iterative manner, to fully mine time-series patterns hidden in representations.
- Plenty of experiments are carried out on two real-world datasets and the results show KANformer achieves the state-of-the-art performance on the power load forecasting task in comparison with seven methods.

The structure of KANformer is organized as follows: Section 2 describes KANformer in details. Section 3 reports experiment results. Finally, Section 4 presents the conclusion.

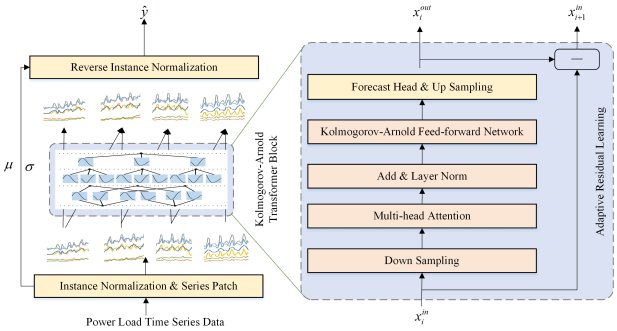

Fig. 1. The overall framework of KANfomer, containing time series patch partitioning, Kolmogorov-Arnold transformer block, adaptive residual learning.

2. Method

In this section, the problem description of the power load forecasting is firstly defined, and then the detailed architecture of KANfomer is introduced, containing time series patch partitioning, Kolmogorov-Arnold transformer block, adaptive residual learning, and the loss function. KANformer transforms the forecasting task into a standard language modeling task, leveraging patching technology to project time series into patch-based representations. During training, an autoregressive optimization function replaces the traditional single-step generation scheme, enabling the model to effectively capture temporal dependencies within the prediction range at the patch level. Furthermore, KANformer's adaptive design allows it to seamlessly adapt to various power grid load datasets with different prediction settings without modifications to the model's architecture.

2.1. Problem Description

Power load forecasting refers to predicting the future power load demand over a certain period using the historical data of multiple related variables. The input to this task typically includes historical electricity load data. These data can be represented as a multi-variable time series $X \in \mathbb{R}^{C\times L}$, where C denotes the number of variable types, and L represents the length of the time series. The output is the predicted electricity load values for the next H time steps $Y \in \mathbb{R}^{C\times H}$, where H indicates the number of future time steps to be forecasted.

2.2. Time Series Patch Partitioning

Power load time series data exhibit significant noise,e.g., measurement errors, anomalous consumption events, and sparse informativeness. Traditional point-wise autoregressive training incurs high computational costs and easily overfits noisy regions, compromising generalization. To effectively capture local load patterns and suppress noise, we partition power load time series data into a range of non-overlapping patches:

Specifically, given a historical power load sequence $X = \{x_1, x_2, \ldots, x_L\}$, where x_t denotes the load at time t, we partition it into T consecutive patches of length L', satisfying $T \times L' = L$. Each patch is denoted $\mathbf{x}^{(i)} \in \mathbb{R}^{C\times L'}, i = 1, 2, \ldots, T$:

$$\mathbf{x}^{(i)} = \left\{ x_{(i-1)L'+1}, x_{(i-1)L'+2}, \ldots, x_{iL'} \right\} \tag{1}$$

This reduces fine-grained noise exposure and forces the model to learn localized load dynamics.

Meanwhile, power load time series data exhibit severe distribution shifts stemming from multiple sources: (1) Scale heterogeneity arises as base/peak loads vary drastically across different substations (e.g., industrial vs. residential feeders may differ by orders of magnitude) and geographic regions; (2) Long-term trends and seasonality introduce baseline drift through economic growth, energy efficiency policies, and seasonal temperature variations that systematically alter consumption patterns; (3) Holiday effects create fundamental divergences between workday and rest-day load profiles, where industrial facilities may show near-zero demand on holidays while residential areas display inverted usage patterns. These shifts violate the standard independent and identically distributed (i.i.d.) assumption, necessitating specialized handling for robust forecasting. To enhance robustness, we apply instance normalization per load patch $\mathbf{x}^{(i)}$:

$$\tilde{\mathbf{x}}^{(i)} = \frac{\mathbf{x}^{(i)} - \mu^{(i)}}{\sigma^{(i)} + \epsilon}, \quad \epsilon > 0 \tag{2}$$

$$\mu^{(i)} = \frac{1}{L'} \sum_{t=1}^{L'} x_t^{(i)} \tag{3}$$

$$\sigma^{(i)} = \sqrt{\frac{1}{L'} \sum_{t=1}^{L'} \left(x_t^{(i)} - \mu^{(i)} \right)^2} \tag{4}$$

where ϵ is a small constant to prevent division by zero. This normalization step significantly boosts the model's robustness to different data distributions. The normalized segments are then used for feature extraction, enabling the model to more effectively capture local features and improving the overall accuracy and robustness of the forecasting.

2.3. Kolmogorov-Arnold Transformer Block

The Kolmogorov-Arnold Network (KAN) represents a paradigm shift from traditional MultiLayer Perceptrons (MLPs) by leveraging the Kolmogorov-Arnold representation theorem. This theorem establishes that any multivariate continuous function can be expressed as a finite composition of univariate functions and additions. Unlike MLPs that employ fixed activation functions at nodes, KANs parameterize learnable activation functions along edges, using spline-parameterized univariate functions. This architectural innovation provides superior approximation capabilities with enhanced interpretability. To this end, a novel KAN Transformer with is designed for feature extraction of power load time series data. The detailed description is shown as follows.

Given the input series patch $\tilde{\mathbf{x}}^{(i)}$, we utilize the position embedding layer and patch embedding layer the to map $\tilde{\mathbf{x}}^{(i)}$ into the hidden space of the transformer:

$$\mathbf{z}_i = \mathrm{POE}\left(\tilde{\mathbf{x}}^{(i)}\right) + \mathrm{PAE}\left(\tilde{\mathbf{x}}^{(i)}\right) \tag{5}$$

where $\mathrm{POE}(\cdot)$ and $\mathrm{PAE}(\cdot)$ denote the position embedding layer and patch embedding layer, respectively. Then, a multi-head self-attention mechanism with the residual connection is utilized to capture intra-patch and inter-patch long-range dependence:

$$\mathbf{h}_i^l = MHSA\left(\mathbf{z}_i^l\right) + \mathbf{z}_i^l, \tag{6}$$

where $\mathbf{h}_i^l$ denote the output of the multi-head self-attention $MHSA(\cdot)$ at the l-th Transformer block. Next, $\mathbf{h}_i^l$ is input into the feed-forward network that replaces standard MLP blocks with KAN layers to perform highly adaptive nonlinear transformation for information propagation:

$$\overline{\mathbf{h}}_i^l = \text{ KAN-FFN }\left(\mathbf{h}_i^l\right), \tag{7}$$

where KAN-FFN($\cdot$) is the Kolmogorov-Arnold feed-forward network:

$$\overline{\mathbf{h}}_i^l = \begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_{l-1}}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_{l-1}}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{l,n_l,1}(\cdot) & \phi_{l,n_l,2}(\cdot) & \cdots & \phi_{l,n_l,n_{l-1}}(\cdot) \end{pmatrix} \mathbf{h}_i^l \tag{8}$$

where n_l and n_{l-1} stand for neuron numbers of the l-th and $l-1$-th the Kolmogorov-Arnold feed-forward layers, respectively. $\phi_l(\cdot)$ denotes the learnable nonlinear mapping function and contains the spline function and the activation function

$$\phi_l\left(\mathbf{h}_i^l\right) = w_a\,\mathrm{Spline}_l\left(\mathbf{h}_i^l\right) + w_b\,\mathrm{SiLU}\left(\mathbf{h}_i^l\right), \tag{9}$$

$$\mathrm{Spline}\left(\mathbf{h}_i^l\right) = \sum_i c_i B_i\left(\mathbf{h}_i^l\right) \tag{10}$$

where $\mathrm{Spline}_l(\cdot)$ is the weighting sum of B-spline functions $B_i(\cdot)$.w_b and w_s denote learnable parameters. For time series forecasting, particularly in power load prediction, KANs offer compelling advantages over conventional MLPs:

- Adaptive feature extraction: KANs dynamically learn optimal activation shapes tailored to complex load patterns (e.g., non-linear holiday effects, temperature-load response curves), whereas MLPs are constrained by predefined activation functions like ReLU or GELU.
- Interpretable learning: The spline-based activations are visually interpretable, revealing domain relationships (e.g., sigmoidal temperature thresholds triggering cooling loads) that remain opaque in MLPs.

To translate the learned representations into future predictions, the output hidden states $\overline{\mathbf{h}}_i^{\text{out}}$ of KAN Transformer are processed by a dedicated forecasting head. This head is implemented as a linear projection layer, transforming the high-dimensional hidden states into a sequence of predicted values within the downsampled temporal resolution.

$$\mathbf{x}_i^{\text{out}} = U\left(KAN_p\overline{\mathbf{h}}_i^{\text{out}}\right). \tag{11}$$

where $U(\cdot)$ denotes the up-sampling operation for aligning the original input time steps and feature dimensionality. KAN_p denotes parameters of the forecasting head.

2.4. Adaptive Residual Learning

A novel adaptive residual learning is devised via transforming the forecasting process into an iterative manner, to fully mine time-series patterns hidden in representations exacted by the KAN Transformer.

Specifically, the input of the $i+1$-th block is the residual of input and output of the i-th block, which can be defined as follows:

$$\mathbf{x}_{i+1}^{in} = \mathbf{x}_i^{in} - \mathrm{ARL}\left(\mathbf{x}_i^{out}\right). \tag{12}$$

Within the auto-regressive training framework, the patch x_{out}^i serves as the prediction for the subsequent patch x_{in}^i. Zero-padding is used to initiate the regression for the first patch.

2.5. The loss function

Let L denote the number of blocks in KAN Transformer, the sum of the forecasting results is all blocks is viewed as the final forecasting result:

$$\hat{\mathbf{y}}_{\text{sum}} = \sum_{l=1}^{L} \mathbf{x}_l^{\text{out}}. \tag{13}$$

Meanwhile, the reversed instance normalization is utilized to recover the characteristics of the input time series data via de-normalization

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}_{\text{sum}} \cdot (\sigma + \epsilon) + \mu. \tag{14}$$

To fully capture temporal dependencies, we define a patch-wise auto-regressive loss function to optimize KAN Transformer:

$$L_{\text{overall}} = \text{MSE}(\hat{\mathbf{y}}, \text{Concat}(X[:, T:], \mathbf{Y})) \tag{15}$$

where $\mathbf{Y}$ denotes future true values.

When making inferences, the KAN Transformer, with its well-designed training architecture and channel-independence assumption, can theoretically conduct general-purpose predictions on any multivariate time series input, no matter the target prediction horizon. Its prediction process resembles that of a language model's decoding. For any given input sequence X, the model first predicts the next immediate patch. Then, this predicted patch is appended to the end of the input sequence. Subsequently, the model uses this updated sequence to predict the second patch, and so on. In this way, the KAN Transformer sidesteps the shortcomings of conventional prediction heads with fixed - length steps. It enhances the model's usability and flexibility, making it better equipped to handle various prediction scenarios and providing a more efficient and general solution for time-series prediction tasks.

3. Results and discussion

3.1. Dataset and Metric

Following [15–17], two common datasets are used as benchmarks for testing performance of the proposed method, i.e., Electricity and StateGridLoad Electricity encompasses the hourly electricity consumption 321 customers with sample number 26304, sampled on an hourly basis from 2012 to 2014. StateGridLoad contains the electricity consumption data of five regions in Northeast China, sampled every 15 minutes in 2020, with a total of 34659 samples.

To evaluate the performance on the two datasets, we employ two commonly used metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE is calculated as the average of the squared differences between the predicted and actual values:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{16}$$

where n is the number of samples, y_i is the actual value, and $\hat{y}_i$ is the predicted value. MAE is computed as the average of the absolute differences between the predicted and actual values:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{17}$$

These two metrics provide insights into the accuracy of the predictions. Lower values of MSE and MAE indicate better model performance. By using both metrics, we can comprehensively evaluate the performance of the models.

3.2. Implementation Details

KANformer is conducted via the PyTorch architecture on the Linux system with an NVIDIA GTX A100 GPU. KANformer contains four blocks. KANformer is optimized via ADAM with the learning rate range in $[\ 0.001, 0.00001\ ]$. The patch length is set as 48 . The batch size is set as 32. The look-back window length is uniformly set to 336 on two datasets while the prediction lengthes are $96, 192, 336$, and 720 , respectively.

3.3. Comparison analysis

Comparison methods: Seven the state-of-the-art time series prediction methods are viewed as baselines, i.e., PatchTFS[9], ARMD[10], Simmtm[11], PGTS[12], Dlinear [13], FMLP [14], and DFM [15].

Comparison results: In comparative experiments across multiple forecasting lengths and datasets, KANformer consistently outperformed other methods on both MSE and MAE metrics by significant margins. For instance, on the Electricity dataset with a prediction length of 96, KANformer achieves an MSE of 0.149 , marking a 0.09 reduction compared to PatchTFS, a 0.005 reduction versus ARMD, and a significant 0.053 decrease relative to DFM. These examples highlight KANformer's superior prediction accuracy across different datasets and prediction lengths compared to other methods.

The performance gains of KANformer can be attributed to several key designs and features.

First, the time series patch partitioning strategy effectively reduces noise interference and compels the model to learn local dynamic characteristics, enhancing robustness and generalization in noisy data. Second, the Kolmogorov-Arnold Transformer block, with spline-based learnable activation functions, dynamically adapts to complex nonlinear

Table 1. MSE results with different prediction lengths {96, 192, 336, 720} of comparison methods on the *Electricity* and *State Grid Load* datasets.

	Electricity				State Grid Load			
Method	96	192	336	720	96	192	336	720
PatchTFS	0.158	0.175	0.205	0.276	0.057	0.082	0.135	0.180
ARMD	0.154	0.170	0.197	0.275	0.055	0.077	0.129	0.187
Simmtm	0.175	0.189	0.218	0.297	0.070	0.098	0.157	0.201
PGTS	0.167	0.181	0.212	0.280	0.065	0.087	0.168	0.192
Dlinea	0.190	0.217	0.253	0.327	0.088	0.122	0.167	0.222
FMLP	0.153	0.173	0.208	0.280	0.058	0.085	0.125	0.192
DFM	0.202	0.224	0.251	0.305	0.107	0.158	0.226	0.240
KANformer	0.149	0.168	0.192	0.269	0.043	0.075	0.121	0.175

patterns in power load data, such as holiday effects and temperature-response curves, improving accuracy in capturing underlying patterns. Additionally, adaptive residual learning transforms the forecasting process into an iterative manner, gradually uncovering hidden patterns in the time series. Finally, the overall architecture of KANformer, including reverse instance normalization and summing predictions from multiple blocks, boosts prediction performance and flexibility, enabling better handling of diverse forecasting scenarios and time steps. These elements work together to make KANformer excel in power load forecasting tasks.

3.4. Ablation analysis

This section assesses the impact of different components in KANformer on the long-term forecasting. There exist three variants: (1) KANformer w/o MLP denotes the overall network is conducted via multilayer perceptrons. (2) KANformer w/o IN denotes the removal of the instance normalization. (3) KANformer w/o ARL denotes the removal of the adaptive residual learning. Ablation results are shown in Table 3. There are two observations: (1) The results consistently reveal that when any single component of KANformer is removed to form a variant, the forecasting performance experiences a noticeable decline. This clear deterioration in performance strongly demonstrates the crucial role and effectiveness of each individual component within the KANformer architecture. (2) KANformer consistently delivers the most accurate and reliable predictions across all tested prediction lengths. This consistent excellence in performance provides further validation for the rationality and robustness of KANformer's design.

3.5. Architecture analysis

In this section, we explore the impact of the Kolmogorov-Arnold Transformer block (KATB) count on the performance of KANformer for power load forecasting. This analysis aims to determine the optimal number of blocks for achieving the best forecasting accuracy. We conducted experiments by varying the number of KATBs in the model, specifically setting the block count to $2, 3, 4$, and 5 , while maintaining all other hyperparameters constant. The experimental results, as shown in Fig. 2, indicate that the forecasting performance of KANformer is closely related to the number of KATBs. When the number of blocks was increased from 2 to 4, both MSE and MAE decreased significantly, suggesting that a higher number of blocks can better capture the complex patterns in power load data. However, further increasing the block count to 5 led to a slight degradation in performance, which might be attributed to overfitting due to the increased model complexity. Specifically, for the Electricity dataset with a prediction length of 96, KANformer achieved the lowest MSE of 0.149 and MAE of 0.250 when the number of KATBs was set to 4 . This result highlights that four blocks are sufficient to model the temporal dependencies and nonlinear characteristics of power load data effectively. Beyond this point, adding more blocks does not necessarily improve performance and may even introduce unnecessary computational overhead. In summary, our analysis demonstrates that the architecture depth of KANformer plays a critical role in its forecasting performance. The optimal number of KATBs was found to be 4, providing a balance between model complexity and forecasting accuracy. This finding serves as a valuable guideline for configuring KANformer in practical power load forecasting applications.

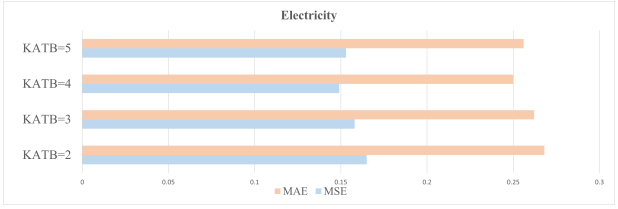

Fig. 2. Performance of KANformer with Different Block Counts on the Electricity Dataset (Prediction Length = 96).

Table 2. MAE results with different prediction lengths {96, 192, 336, 720} of comparison methods on the *Electricity* and *State Grid Load* datasets.

	Electricity				State Grid Load			
Method	96	192	336	720	96	192	336	720
PatchTFS	0.258	0.279	0.304	0.366	0.161	0.217	0.273	0.424
ARMD	0.255	0.272	0.299	0.371	0.159	0.209	0.268	0.416
Simmtm	0.280	0.311	0.348	0.401	0.183	0.228	0.315	0.430
PGTS	0.274	0.301	0.323	0.392	0.177	0.233	0.288	0.435
Dlinea	0.302	0.326	0.357	0.428	0.187	0.243	0.289	0.424
FMLP	0.255	0.273	0.300	0.366	0.158	0.209	0.269	0.407
DFM	0.312	0.335	0.378	0.422	0.203	0.288	0.326	0.450
DAUformer	0.250	0.268	0.293	0.358	0.155	0.206	0.262	0.399

Table 3. Ablation analysis of KANformer on the Electricity dataset

	MSE				MAE			
Variant	96	192	336	720	96	192	336	720
KANformer w/o MLP	0.152	0.173	0.197	0.270	0.253	0.270	0.297	0.361
KANformer w/o IN	0.187	0.204	0.256	0.302	0.286	0.310	0.337	0.399
DAUformer w/o ARL	0.157	0.178	0.201	0.272	0.256	0.274	0.298	0.362
KANformer	0.149	0.168	0.192	0.269	0.250	0.268	0.293	0.358

4. Conclusion

In this paper, we have proposed KANformer, a novel architecture that combines the KolmogorovArnold representation theorem with the Transformer framework for power load forecasting. KANformer addresses the interpretability and flexibility issues commonly encountered in existing deep learning approaches. By leveraging the Kolmogorov-Arnold theorem, KANformer dynamically learns complex nonlinear patterns in power load data, while the spline-based activation functions enhance approximation capabilities and provide a foundation for interpretability. The adaptive residual learning mechanism transforms the forecasting process into an iterative manner, allowing the model to extract and refine hidden time-series patterns, resulting in more accurate predictions. Our extensive experiments on two real-world power grid datasets demonstrate that KANformer surpasses seven state-of-the-art methods in terms of both MSE and MAE across different prediction horizons. For future research, several directions could be explored to further enhance KANformer. One promising avenue is the incorporation of additional data sources, such as weather forecasts, economic indicators, or smart meter readings, to enrich the model's input and potentially improve its forecasting accuracy. Another direction is the optimization of computational efficiency. This could involve developing more efficient training algorithms, leveraging hardware acceleration techniques, or exploring model compression methods to reduce resource consumption.

References

[1] Q. Ma, Z. Liu, Z. Zheng, Z. Huang, S. Zhu, Z. Yu, and J. T. Kwok, (2024) *"A survey on time-series pre-trained models"* **IEEE Transactions on Knowledge and Data Engineering**: DOI: 10.1109/TKDE.2024.3475809.

[2] Q. Deng, C. Wang, J. Sun, Y. Sun, J. Jiang, H. Lin, and Z. Deng, (2023) *"Nonvolatile CMOS memristor, reconfigurable array, and its application in power load forecasting"* **IEEE Transactions on Industrial Informatics *20***(4): 6130–6141. DOI: 10.1109/TII.2023.3341256.

[3] J. Gao, M. Liu, P. Li, J. Zhang, and Z. Chen, (2024) *"Deep Multiview Adaptive Clustering With Semantic Invariance"* **IEEE Transactions on Neural Networks and Learning Systems *35***(9): 12965–12978. DOI: 10.1109/TNNLS.2023.3265699.

[4] J. Gao, M. Liu, P. Li, A. A. Laghari, A. R. Javed, N. Victor, and T. R. Gadekallu, (2023) *"Deep Incomplete Multiview Clustering via Information Bottleneck for Pattern Mining of Data in Extreme-Environment IoT"* **IEEE Internet of Things Journal *11***(16): 26700–26712. DOI: 10.1109/JIOT.2023.3325272.

[5] Q. Xing, X. Huang, J. Wang, and S. Wang, (2024) *"A novel multivariate combined power load forecasting system based on feature selection and multi-objective intelligent optimization"* **Expert Systems with Applications *244***: 122970. DOI: 10.1016/j.eswa.2023.122970.

[6] G.-F. Fan, Y.-Y. Han, J.-W. Li, L.-L. Peng, Y.-H. Yeh, and W.-C. Hong, (2024) *"A hybrid model for deep learning short-term power load forecasting based on feature*

extraction statistics techniques" **Expert Systems with Applications *238***: 122012. DOI: 10.1016/j.eswa.2023.122012.

[7] Z. Zhan, X. Mao, H. Liu, and S. Yu, (2025) *"STGL: Self-Supervised Spatio-Temporal Graph Learning for Traffic Forecasting"* **Journal of Artificial Intelligence Research 2**(1): 1–8. DOI: 10.70891/JAIR.2025.040001.

[8] B. Li, Y. Zhao, S. Zhelun, and L. Sheng. "Danceformer: Music conditioned 3d dance generation with parametric motion transformer". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. ***36***. 2. 2022, 1272–1279. DOI: 10.1609/aaai.v36i2.20014.

[9] P. Tang and W. Zhang. "Unlocking the Power of Patch: Patch-Based MLP for Long-Term Time Series Forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. ***39***. 12. 2025, 12640–12648. DOI: 10.1609/aaai.v39i12.33378.

[10] J. Gao, Q. Cao, and Y. Chen. "Auto-regressive moving diffusion models for time series forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. ***39***. 16. 2025, 16727–16735. DOI: 10.1609/aaai.v39i16.33838.

[11] J. Dong, H. Wu, H. Zhang, L. Zhang, J. Wang, and M. Long, (2023) *"Simmtm: A simple pre-training framework for masked time-series modeling"* **Advances in Neural Information Processing Systems *36***: 29996–30025.

[12] T. Zhou, P. Niu, L. Sun, R. Jin, et al., (2023) *"One fits all: Power general time series analysis by pretrained lm"* **Advances in neural information processing systems *36***: 43322–43355.

[13] A. Zeng, M. Chen, L. Zhang, and Q. Xu. "Are transformers effective for time series forecasting?" In: *Proceedings of the AAAI conference on artificial intelligence*. ***37***. 9. 2023, 11121–11128. DOI: 10.1609/aaai.v37i9.26317.

[14] K. Yi, Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, N. An, D. Lian, L. Cao, and Z. Niu, (2023) *"Frequency-domain mlps are more effective learners in time series forecasting"* **Advances in Neural Information Processing Systems *36***: 76656–76679.

[15] A. Das, W. Kong, R. Sen, and Y. Zhou. "A decoder-only foundation model for time-series forecasting". In: *Forty-first International Conference on Machine Learning*. 2024.

[16] Y. Wang, Y. Hao, K. Zhao, and Y. Yao, (2025) *"Stochastic configuration networks for short-term power load forecasting"* **Information Sciences *689***: 121489. DOI: 10.1016/J.INS.2024.121489.

[17] J. Wang, M. Kou, R. Li, Y. Qian, and Z. Li, (2025) *"Ultra-short-term wind power forecasting jointly driven by anomaly detection, clustering and graph convolutional recurrent neural networks"* **Advanced Engineering Informatics *65***: 103137. DOI: 10.1016/J.AEI.2025.103137.