scientific reports

Check for updates

OPEN

Self-adaptive evolutionary neural networks for high-precision short-term electric load forecasting

Muhammad Abbas[1], Yanbo Che[1], Sarmad Maqsood[2], Muhammad Zain Yousaf[3,4]✉, Mustafa Abdullah[5], Wajid Khan[1], Saqib Khalid[6], Mohit Bajaj[7,8] & Mohammad Shabaz[9]✉

Reliable short-term electric load forecasting (STLF) is essential for enhancing grid stability, optimizing energy distribution, and minimizing operational costs in modern power systems. However, existing forecasting models, including statistical approaches and deep learning architectures such as multi-layer perceptron (MLP), struggle to capture complex nonlinear load variations while maintaining computational efficiency. To overcome these limitations, a self-adaptive Kolmogorov–Arnold network (SADE-KAN), an optimized forecasting framework that combines the power of Kolmogorov–Arnold networks (KAN) with self-adaptive differential evolution (SADE) is introduced to enhance both predictive accuracy and computational efficiency. Unlike conventional MLP models, KAN replaces fixed activation functions with spline-based learnable functions that offers greater flexibility in capturing temporal dependencies. However, these learnable activation functions introduce a new set of hyperparameters that require careful optimization to ensure efficient training and manage network complexity. To address this, SADE dynamically tunes these hyperparameters, ensuring an optimal balance between accuracy, complexity, and training efficiency. The proposed SADE-KAN model is validated on ISO-NE hourly load data (2019–2023, ~1 million observations) across multiple forecasting horizons (24, 48, 96, and 168 h). Experimental results demonstrate that SADE-KAN reduces mean absolute percentage error (MAPE) by up to 35% and root mean squared error (RMSE) by 38% compared to MLP models, while requiring 35% fewer learnable parameters. Despite a slightly higher training time, SADE-KAN significantly enhances generalization and robustness, capturing rapid load fluctuations more effectively than MLP, conventional KAN and other recently published advanced models. These findings establish SADE-KAN as a computationally efficient and highly accurate forecasting framework, offering a robust solution for real-time power system applications, demand response strategies, and energy market operations.

Keywords Electric load forecasting, Kolmogorov–Arnold networks, Machine learning, Optimization, Power systems, Self-adaptive differential evolution, Smart grids, Time series prediction

Abbreviations

ARIMA	Auto-regressive integrated moving average
Bi-LSTM	Bi-directional long short-term memory
CNN	Convolutional neural network
DCNN	Dilated convolutional neural network
DE	Differential evolution
ES	Exponential smoothing
FFN	Feed-forward network

[1]Key Laboratory of Smart Grid of Ministry of Education, School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China. [2]Centre of Real Time Computer Systems, Faculty of Informatics, Kaunas University of Technology, 51386 Kaunas, Lithuania. [3]Center for Research on Microgrids (CROM), Huanjiang Laboratory, Zhuji, Shaoxing 311800, Zhejiang, China. [4]Department of Electrical and Electronic Engineering Technology, University of Johannesburg, Johannesburg, South Africa. [5]Electric Vehicles Engineering Department, Faculty of Engineering, Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, Jordan. [6]School of Electrical Engineering, The University of Lahore, Lahore, Pakistan. [7]Department of Electrical Engineering, Graphic Era (Deemed to be University), Dehradun 248002, India. [8]Graphic Era Hill University, Dehradun 248002, India. [9]Model Institute of Engineering and Technology, Jammu, Jammu and Kashmir, India. ✉email: zain.yousaf@huat.edu.cn; mohammad.shabaz@amu.edu.et

GRU	Gated recurrent unit
ISO-NE	Independent system operator—New England
KAN	Kolmogorov–Arnold network
KART	Kolmogorov–Arnold representation theorem
KF	Kalman filtering
KNN	K nearest neighbors
LGBM	Light gradient boosting machine
LR	Linear regression
LSTM	Long short-term memory
LTLF	Long-term load forecasting
MAE	Mean absolute error
MAPE	Mean absolute percentage error
ML	Machine learning
MLP	Multi-layer perceptron
MTLF	Medium-term load forecasting
NNs	Neural networks
RMSE	Root mean squared error
RNNs	Recurrent neural networks
SADE	Self-adaptive differential evolution
STLF	Short-term load forecasting
SVR	Support vector regressor
VSTLF	Very short-term load forecasting
XGB	Extreme gradient boosting

Indices and parameters

$X\mathbb{R}^{m.n}$	Input feature matrix where m is the number of time steps and n is features
x_{ij}	Value of the jth feature at the ith timestamp
$\underset{k}{t}$	Feature vector at the time step k, $\underset{k}{\overrightarrow{t}} = [\quad x_{k1}, \quad x_{k2}, \quad \ldots, \quad x_{kn} \quad]$
$\hat{Y}$	Predicted output load sequence
y_{m+k}	Load value at the $m+k$th timestamp
m	Length of historical observation window
T	Forecast horizon (total number of prediction steps)
N_{out}	Number of output nodes in KAN
N_{in}	Number of input nodes (flattened features passed to KAN)
k	Order of the B-spline basis function
G	Number of grid intervals of B-spline
Φ_l	Function matrix applied at layer
$\phi_{l,j,i}\,(x)$	Univariate function on the edge from neuron i in layer to neuron j in layer $l+1$
$x_{l+1,j}$	Activation of the jth neuron in layer $l+1$
$\hat{x}_{l,j,i}$	Post-activation output from neuron i in layer to neuron j in layer $l+1$
nl	Number of neurons in layer
$b\,(x)$	Basis function
$B_i\,(x)$	B-spline basis function of order k
C_i	Trainable coefficients for the B-spline basis functions
$\mathrm{KAN}\,(X)$	Output of the Kolmogorov–Arnold Network model
$X_i = G_i, D_i, W_i$	Solution vector comprising KAN hyperparameters
X_i^g	ith Individual (solution) in generation g
N_P	Population size in SADE
F_i	Mutation factor for individual i, $F_i \in [0.4,0.9]$
C_i	Crossover rate for individual i, $C_i \in [0.1,0.9]$
V_i^{g+1}	Mutant vector for individual i at generation $g+1$
U_i^{g+1}	Trial vector after crossover
$f\,(.)$	Fitness function evaluating solution quality
F_{mean}	Mean of successful mutation factors
σF	Standard deviation used in sampling mutation factor
C_{mean}	Mean of successful crossover rates
σC	Standard deviation used in sampling crossover rate

Modern power systems aim to reduce peak load and maintain a balance between power generation and consumption to ensure safe, reliable, and cost-effective operations[1]. Utilities engaged in generation, transmission, and distribution rely on load forecasting to facilitate effective resource allocation, generation planning, integration of renewable energy sources, and stability of the grid[2]. Given its critical role, load forecasting has been a long-standing area of research. Despite numerous advancements, the demand for highly accurate models continues to rise as forecast precision directly influences system efficiency and financial outcomes. For instance, Hobbs et al.[3] showed that even a 1% improvement in forecast accuracy could translate into substantial economic savings for utilities.

Recent advancements in energy forecasting have seen a growing reliance on artificial intelligence and hybrid machine learning models to improve prediction accuracy across various energy domains. Al-Suod et al.[4] employed artificial neural networks to forecast energy consumption in mining plants, showcasing the

adaptability of ANNs to industrial loads with high nonlinearity. Building upon the potential of machine learning in complex systems, Singh et al.[5] explored energy management and power forecasting in grid-connected microgrids comprising multiple distributed energy resources, integrating multiple data sources to support real-time decision-making. Extending the use of neural models, Alharbi et al.[6] combined multilayer perceptrons with a waterwheel plant algorithm for hyperparameter tuning to forecast energy efficiency in buildings, highlighting the role of hybrid optimization in enhancing model performance. Guermoui et al.[7] advanced photovoltaic (PV) power forecasting by employing multi-scale fusion techniques across diverse case studies, emphasizing the value of spatio-temporal data integration. Similarly, Sharma et al.[8] addressed the challenges of solar forecasting under varying weather conditions by utilizing gradient descent and Levenberg–Marquardt-based ANN models, demonstrating the benefit of adaptive training algorithms. Focusing on regional applications, Mfetoum et al.[9] designed a multilayer perceptron for solar irradiance forecasting in Central Africa, incorporating meteorological insights to account for climatic variability. In the hydrological forecasting domain, Reihanifar et al.[10] developed a multi-objective genetic programming model for drought prediction using meteorological data, illustrating the versatility of evolutionary algorithms in time series forecasting. Forecasting irradiance with hybrid deep learning, Mouloud et al.[11] assessed bidirectional architectures for short-term global horizontal irradiance (GHI), demonstrating improved accuracy via deep temporal modeling. Pushing the boundary of renewable energy forecasting, Alhussan et al.[12] targeted green hydrogen production with a hybrid dynamic optimization ensemble, reflecting growing interest in clean fuel systems. Al-qaness et al.[13] proposed ResInformer, a residual transformer-based architecture for PM2.5 forecasting, bringing attention to the potential of transformer models in high-dimensional environmental time series. Wind power forecasting was explored by Saeed et al.[14] using an enhanced Al-Biruni Earth radius metaheuristic algorithm, showing notable improvements in prediction quality through metaheuristic-tuned models. Similarly, Molu et al.[15] utilized a hybrid deep learning framework with Bayesian optimization for short-term solar irradiance forecasting, underlining the effectiveness of probabilistic search in hyperparameter tuning. Seasonal forecasting of GHI was tackled by Kheldoun et al.[16] using a combined CNN-BiGRU model, indicating the benefits of combining spatial and temporal neural networks for grid-connected PV applications. Finally, Khelifi et al.[17] introduced a hybrid TVF-EMD-ELM model for short-term PV power forecasting, integrating signal decomposition and extreme learning machines to effectively capture non-stationary features in solar data. Collectively, these studies illustrate a clear trend toward hybrid, adaptive, and domain-specific forecasting models, often merging deep learning, signal processing, and evolutionary optimization techniques to address the growing complexity of energy systems.

Generally, load forecasting processes are categorized into two primary groups: short-term load forecasting (STLF) and long-term load forecasting (LTLF). Further, these processes can be subdivided into very short-term load forecasting (VSTLF) and medium-term load forecasting (MTLF) based on their time horizons[18]. Among these, STLF is crucial for making effective real-time decisions and managing system operations such as day-ahead and hour-ahead scheduling, energy trading and demand response management[19]. To perform STLF, both machine learning (ML) techniques and non-ML or statistical techniques have been adopted. The non-ML techniques include auto-regressive integrated moving average (ARIMA)[20], exponential smoothing (ES)[21], linear regression (LR)[22] and Kalman filtering (KF)[23]. These techniques traditionally address the time series forecasting problem, utilizing statistical methods to forecast future load based on the historical data. ARIMA is particularly known among the non-ML techniques, as it combines auto-regression, integration, and moving averages to predict future load. Its variants, such as seasonal ARIMA (SARIMA), have been developed to accommodate seasonality in data series[24]. ES are another widely used set of non-ML forecasting techniques, known for their simplicity and efficiency in dealing with data that exhibit trends and seasonality[25]. While these models are widely used for their efficiency and interpretability, they have limitations in capturing the dynamics of nonlinear systems, complicated seasonal patterns, and long-term dependencies[26].

On the other hand, ML techniques, particularly neural networks (NNs) offer greater flexibility in modeling nonlinearity, handling large data volumes, and learning from historical patterns[27]. The simple multi-layer perceptron (MLP) was one of the earliest NNs used for STLF, showing considerable promise in modeling the nonlinear data[27]. Based on the foundation of MLP, more complex architectures have been developed to enhance the capabilities of NNs. Wide and deep NNs stand out for their ability to capture both linear and non-linear relationships within large datasets, handling multiple input features and learning hierarchical patterns effectively[28]. Advanced architectures like convolutional neural networks (CNNs)[29], recurrent neural networks (RNNs)[30], long short-term memory (LSTM) networks[31], and gated recurrent units (GRUs)[32] have evolved from the simple MLP and are widely used in STLF. These models effectively handle temporal sequences, retaining information over longer periods while mitigating the vanishing gradient problem, making them highly suitable for load forecasting tasks[33].

Researchers have leveraged these advanced NN techniques in various innovative ways to improve load forecasting accuracy and efficiency. For example, Kong et al.[34] developed an LSTM-based model for short-term residential load forecasting, demonstrating its superiority over traditional methods in capturing complex temporal patterns. However, the model's training time is significantly longer due to its complexity and requires substantial computational resources. Smyl et al.[35] introduced a hybrid method combining ES with RNNs for forecasting electricity demand, showing improved accuracy and robustness. Despite its effectiveness, its interpretability is limited, making it difficult for stakeholders to understand the model's decision-making process. Wu et al.[36] proposed a CNN-LSTM hybrid model, utilizing CNNs for feature extraction and LSTMs for sequence prediction. This model outperformed standalone models in accuracy and computational efficiency. However, the model's performance heavily depends on the quality of the input features, and it may require extensive hyperparameter tuning. Yang et al.[37] applied a dilated CNN (DCNN) for day-ahead electricity price forecasting, capturing long-term dependencies more effectively than conventional RNN-based models. The primary limitation is the need for large datasets to train the DCNN effectively, which may not always be available. Marino

et al.[38] utilized deep NNs with stacked LSTM layers for multi-step ahead forecasting, significantly improving performance over shallow networks. However, the deep architecture increases the risk of overfitting, especially when the training dataset is not sufficiently large. Hong et al.[39] demonstrated that LSTM models incorporating weather forecasts outperformed those using only historical load data. However, the reliance on accurate weather data can be a limitation, as errors in weather forecasting can directly impact the accuracy of electricity demand forecasts. Unfortunately, most of the techniques, being primarily based on the MLP architecture, often suffer from limitations such as lack of interpretability, reduced accuracy in complex scenarios, high computational complexity, and significant data dependency. These shortcomings raise the need for further research and innovation in developing more efficient and accurate forecasting models[19,24,26,27,33].

Surprisingly, in a recent study[40], Kolmogorov–Arnold Network (KAN) is introduced as a novel NN architecture potentially supplanting the traditional MLP. Inspired by the Kolmogorov–Arnold representation theorem (KART), KAN leverages this theorem to create a unique architecture. They revolutionize by replacing the linear weights with spline-based univariate functions as learnable activation functions along the network edges. This unique architecture improves both accuracy and interpretability of the networks while significantly reducing the number of parameters needed (138000 in KAN compared to 394,000 in MLP)[41]. Several studies have explored the theoretical and practical aspects of KAN. For instance, the authors in[41] examined KAN in time-series forecasting and highlighted that KAN consistently outperformed MLP, achieving lower error metrics while using fewer computational resources. However, the study also highlighted the importance of optimizing parameters such as node count and grid size to improve performance. Similarly, the authors in[42] examined the use of KAN for time-series forecasting highlighted that, although KAN provided improved accuracy over MLP, the training time was significantly longer due to the need to learn coefficients for the spline functions. Further, in[43] the Graph KAN variant faced the same challenge of slow training speed, and optimizing training time was left as a goal for future work. Moreover, a comprehensive survey in[44] explored the use of various polynomial functions as basis functions in KAN, the study pointed out the need for further research into model complexity, parameter tuning, and training time, particularly on more complex datasets. In[45], KAN models were found to be more effective than MLP models in federated learning tasks, but the high computational cost of spline-based functions was noted as an area for future optimization. Table 1 presents summary of the recent works, their contributions and limitations. These findings emphasize the importance of careful optimization of spline-based KAN parameters to balance model complexity, efficiency, and computational demands. To obtain accurate load forecasts, the following challenges in KAN need to be addressed:

- The increased training time due to the complexity of learning spline-based coefficients, particularly in complex architectures.
- The need to optimize model architecture and spline grid settings as KAN introduce new hyperparameters.
- A trade-off between complexity, training time, and the risk of overfitting when selecting these hyperparameters.

References	Forecasting model	Ke contributions	Identified limitations
[24,26,27,33]	MLP-based techniques	Provided foundational frameworks for STLF using standard neural networks	Suffer from interpretability issues, reduced accuracy in complex scenarios, and high data/computational demands
Kong et al.[34]	LSTM-based model	Demonstrated superiority over traditional methods in capturing complex temporal patterns for residential load forecasting	Long training time and high computational resource requirements
Smyl et al.[35]	Hybrid Exponential Smoothing and RNN	Improved forecasting accuracy and robustness for electricity demand	Limited interpretability of the model, challenging stakeholder understanding
Wu et al.[36]	CNN-LSTM hybrid model	Combined CNNs for feature extraction and LSTMs for sequence prediction, outperforming standalone models	Performance is heavily dependent on feature quality and requires extensive hyperparameter tuning
Yang et al.[37]	Dilated CNN (DCNN)	Effectively captures long-term dependencies in day-ahead electricity price forecasting	Requires large datasets for effective training, which may not always be available
Marino et al.[38]	Deep LSTM network	Achieved better multi-step ahead forecasting accuracy using stacked LSTM layers	Increased risk of overfitting with deep architecture, especially on small datasets
Hong et al.[39]	LSTM with weather forecasts	Showed enhanced performance using weather data alongside historical load	Accuracy heavily depends on the reliability of external weather forecasts
Liu et al.[40]	Kolmogorov–Arnold Network (KAN)	Novel architecture replacing linear weights with spline-based univariate functions, improving interpretability and parameter efficiency	Requires careful optimization of node count and grid size
Abbas et al.[41]	KAN for short term load forecasting	Outperformed MLP in accuracy and resource efficiency	The need to optimize model architecture and spline grid settings
VacaRubio et al.[42]	KAN	Achieved higher accuracy than MLP for time-series data	Noted longer training time due to learning spline coefficients
Kiamari et al.[43]	Graph-KAN	Extended KAN for graph-based data with consistent performance	Training speed was slow, with optimization deferred to future work
Seydi et al.[44]	KAN with polynomial basis functions	Surveyed basis function choices and emphasized model complexity challenges	Called for more research into parameter tuning and efficient training
Zeydan et al.[45]	KAN in federated learning	Demonstrated effectiveness over MLP in distributed learning	High computational cost of spline-based functions remained a challenge

Table 1. Summary of the recent works, their contributions and limitations.

To address the above challenges, this study introduces KAN optimized with self-adaptive differential evolution (SADE) as a novel approach for STLF, a method not previously explored in the existing literature. The adaptive spline-based activation functions of KAN enable the capture of complex and nonlinear patterns in load data, while SADE optimization enhances efficiency, mitigates overfitting and reduce training time. By auto-tuning the crucial hyperparameters, our approach addresses the challenge of model architecture and spline grid settings without compromising predictive accuracy. The primary objective is to asses the practical applicability of our proposed model by examining its accuracy and efficacy with respect to their trainable parameters, training time, data dependency, and capability to capture temporal dependencies. We validate the efficacy of our proposed model on the latest real-world ISO-NE-CA load data from Jan 2019 to Dec 2023, using performance metrics such as root mean squared error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) across various prediction horizons. Moreover, a detailed comparative analyses are conducted against the conventional models and advanced frameworks to assess its competitiveness in the field of STLF. Conceptual diagram of the proposed SADE-KAN model is presented in Fig. 1.

Contributions of this work are summarized as follows:

- This paper introduces the KAN optimized with SADE as a novel approach for multi-horizon STLF, aiming to achieve higher accuracy by leveraging its adaptive spline-based activation functions.
- KAN have recently emerged and its application for short term electric load forecasting remains unexplored, particularly on large datasets with high dimensionality and numerous dependent variables. This study fills that gap by evaluating KAN on a real-world large-scale dataset having significant variability.
- By optimizing the hyperparameters introduced by the spline-based activation functions using SADE, this study overcomes training and complexity challenges of KAN.
- Performance of the proposed SADE-KAN is comprehensively evaluated in terms of accuracy, parameters count, several error metrics, training time, and is compared against the conventional models and advanced frameworks.

The rest of the paper is organized as follows:

Section "Problem formulation" provides the problem formulation, KAN's fundamental background and the proposed SADE-KAN. Section "Dataset and preprocessing" presents the dataset and preprocessing steps used in this study. Section "Experimental setup and results" details the experimental setup and presents the simulation results, analyzing the performance of SADE-KAN. Finally, Section "Conclusion" offers concluding remarks, followed by references.

Problem formulation

We aim to predict future electricity load values based on historical time-series data. Given a historical observation window of length m, the goal is to forecast the load sequence over a specified horizon $T-m$, represented as:

$$\hat{Y} = [\quad y_{m+1}, \quad y_{m+2}, \quad \ldots, \quad y_T \quad] \tag{1}$$

where y_{m+k} corresponds to the load value at the $m+k$th timestamp for $k\,[1, T-m]$. The input data for forecasting is represented by a multivariate time series matrix X of historical load and its associated features, where each row corresponds to a specific time step, and each column represents a feature, expressed as:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} = \begin{bmatrix} \overrightarrow{t}_1 \\ \overrightarrow{t}_2 \\ \vdots \\ \overrightarrow{t}_m \end{bmatrix} \tag{2}$$

where x_{ij} represents the value of the jth feature at the ith timestamp. The vector $\overrightarrow{t}_k = [\quad x_{k1}, \quad x_{k2}, \quad \ldots, \quad x_{kn} \quad]$ represents the feature vector at time step k for $k\,[1, m]$, and n is the number of features, which may include variables such as system load, temperature, electricity pricing, and other relevant factors. The problem is framed as a supervised learning task, where the objective is to map the historical feature matrix X to the future load sequence $\hat{Y}$, thereby enabling accurate prediction of load values at future timestamps.

Kolmogorov Arnold network background

Contrary to MLP, which are deeply rooted in the universal approximation theorem stating that "a feed-forward network (FFN) with a single hidden layer and a finite number of neurons can approximate any continuous function on compact subsets of $\mathbb{R}^n$", KAN derive their foundation from the KART theorem. KART states that any continuous function f dependent on multiple variables, such as $X = [x_1, x_2, \ldots, x_n]$ within a bounded domain, can be decomposed into a finite set of simpler continuous functions[23]. In more formal terms, a real and smooth multivariate continuous function $f\,(X) : [0,1]^n \to \mathbb{R}$ can be expressed as a composition of univariate functions and the addition operations as

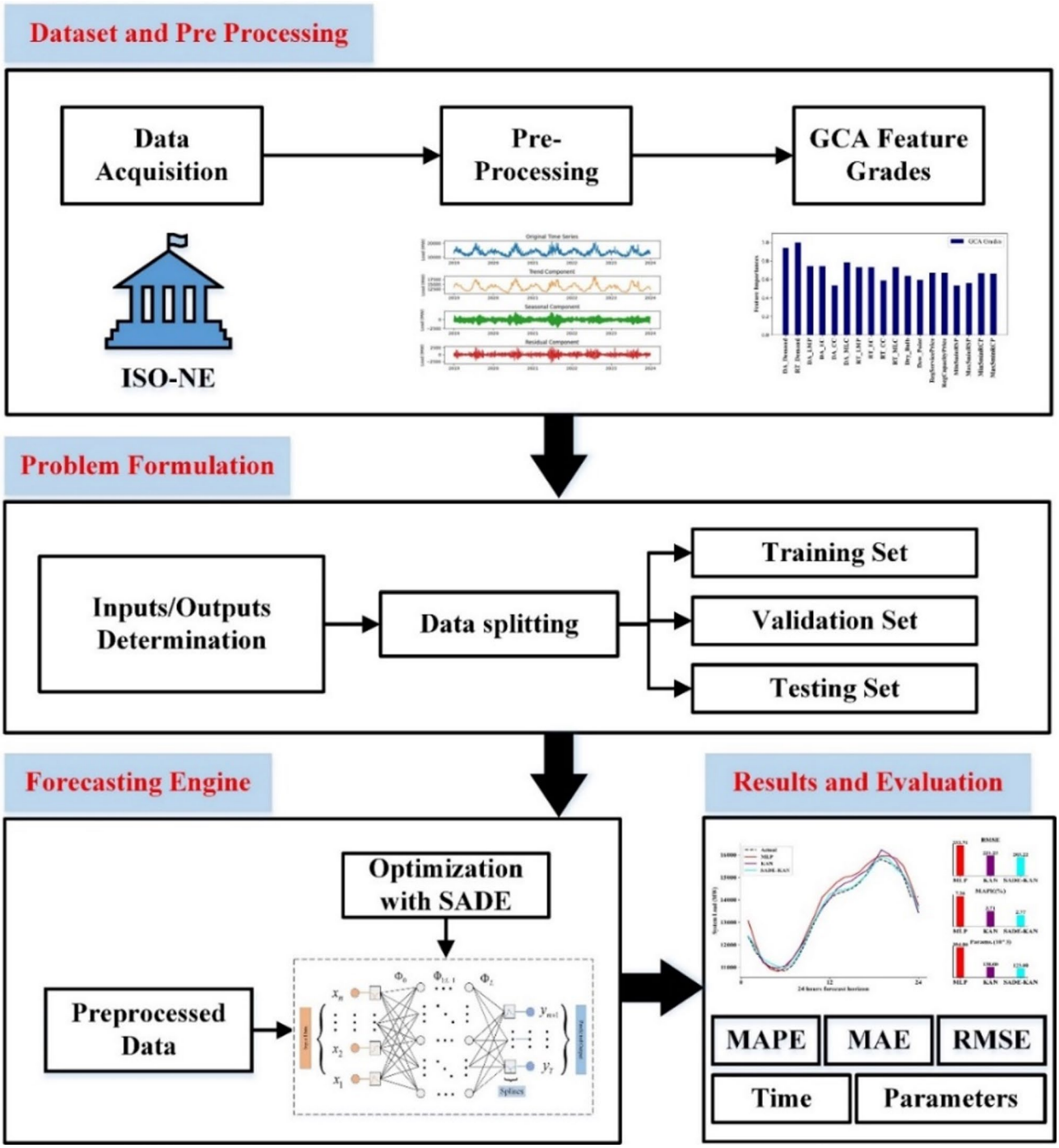

Fig. 1. Conceptual diagram of the proposed SADE-KAN model for STLF.

$$f(X) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \Phi_{q,p}(x_p) \right) \tag{3}$$

where $\Phi_{q,p}$ are the univariate functions that map each input variable x_p, specifically $\Phi_q : \mathbb{R} \to \mathbb{R}$ are the outer functions and $\Phi_{q,p} : [0,1] \to \mathbb{R}$ are the inner functions.

This implies that instead of directly approximating a complex multivariate function f, we can decompose it into a combination of one-dimensional simpler functions. This decomposition approach initially seems highly beneficial for machine learning, as it simplifies the challenge of learning a high-dimensional function into learning a smaller number of one-dimensional functions. Theoretically, this could enhance the learning process and make it more efficient and manageable. However, in practice, these one-dimensional functions $\Phi_{q,p}$

can be very complicated and exhibit non-smooth characteristics hence making them difficult to learn using traditional machine learning techniques. Consequently, despite its theoretical advantages, the KART has seen limited application in the field of machine learning.

Interestingly, recent theoretical developments as presented in[40] suggest that the KART could inspire new types of neural network architectures. These architectures have the potential to leverage this theorem in a way that is practical and effective for machine learning tasks. The authors in [41] propose a two-layer neural network model described by Eq. (3) having two layers of non-linearities, with $2n+1$ terms in the middle layer. Thus, the key task is to identify appropriate functions $\Phi_{q,p}$ and Φ_q that map the input variable x_p effectively.

Kolmogorov Arnold Network

In MLP, a layer includes a linear transformation followed by nonlinear operations, and the network can be made deeper by adding more layers [46]. Let us define an MLP network as a sequence of linear transformations and non-linear activations applied to the input X as

$$\mathrm{MLP}(X) = (W_{L-1} \circ \sigma \circ \ldots \circ W_1 \circ \sigma \circ W_0) X \tag{4}$$

where $W_0, \ldots, W_{L-1}$ are the learnable weights on the edges and σ is the fixed activation function such as ReLu applied at each node. The KAN replace these fixed activation functions and express $f(X)$ as a composition of inner and outer function matrices applied to the input vector.

Mathematically, the output of a KAN model is a composition of layers, where each layer applies an activation function to its inputs. These layers, build on each other, transform the output progressively. The input to the KAN is passed through successive layers, where the activation value of a neuron in the $l+1$ layer is computed as the sum of all incoming post-activations from the layer :

$$x_{l+1,j} = \sum_{i=1}^{nl} \hat{x}_{l,j,i} = \sum_{i=1}^{nl} \phi_{l,j,i}(x_{l,j}) \tag{5}$$

Here, $\phi_{l,j,i}(\cdot)$ denotes the activation function for the connection between the i^{th} neuron of layer and the j^{th} neuron in layer $l+1$. Similarly, $\hat{x}_{l,j,i}$ and nl are the post activation value and number of neurons in the l^{th} layer respectively. It can be presented in the matrix form as:

$$x_{l+1} = \underbrace{\begin{bmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_l}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n_l}(\cdot) \end{bmatrix}}_{\Phi_l} x_l \tag{6}$$

Here Φ_l is the function matrix representing a set of transformation applied by the layer in the network, with ranging from 0 to $L-1$. The composition of these transformations across all layers results in the output as:

$$\mathrm{KAN}(x) = (\Phi_{L-1} \circ \ldots \circ \Phi_1 \circ \Phi_0) x \tag{7}$$

The input data x is transformed by each function matrix ϕ_l in the l^{th} layer. This architecture includes B-splines as the core activation function for these transformations, where each $\phi_{l,j,i}(x)$ is parameterized as a B-spline such that:

$$\phi_{l,j,i}(x) = \omega \left(b(x) + \sum_{i=0}^{k+Grid-1} C_i B_i(x) \right) \tag{8}$$

where $b(x) = \frac{x}{1+e^{-x}}$ is a basis function, $B_i(x)$ is the B-spline function of order k with trainable coefficients C_m that determines the shape of the B-spline and $Grid$ refers to the number of intervals. The B-splines are used to approximate the non-linear relationship between features in the data. The KAN is particularly suited for this task due to its ability to capture complex relationships within the data through a series of nonlinear transformations [40]. KANs are visualized similarly to fully connected layers in traditional neural networks, with the distinction that each edge represents a one-dimensional function. This design simplifies the interpretation of complex networks by directly representing function layers [40]. Consequently, KAN offers a more interpretable connection between input features and output, showcasing its potential for improved transparency.

We have framed the load forecasting problem as a supervised learning framework, using a training dataset having input–output pairs and represented as $[\ x_{k1},\ x_{k2},\ \ldots,\ x_{kn}\ ]$. We aim to find f that approximates $[\ y_{m+1},\ y_{m+2},\ \ldots,\ y_T\ ]$. More formally:

$$f([\ y_{m+1},\ y_{m+2},\ \ldots,\ y_T\ ]) \approx x_{k1},\ x_{k2},\ \ldots,\ x_{kn} \tag{9}$$

Our model employs KAN structured as $[N_{in}, n, N_{out}]$ with n as the count of neurons in the hidden layer. The relevant features are selected by grey correlation analysis (GCA) and flattened to serve as input to the KAN. This architecture is designed to produce forecasts for various time horizons: the next day (24 h), the next two days

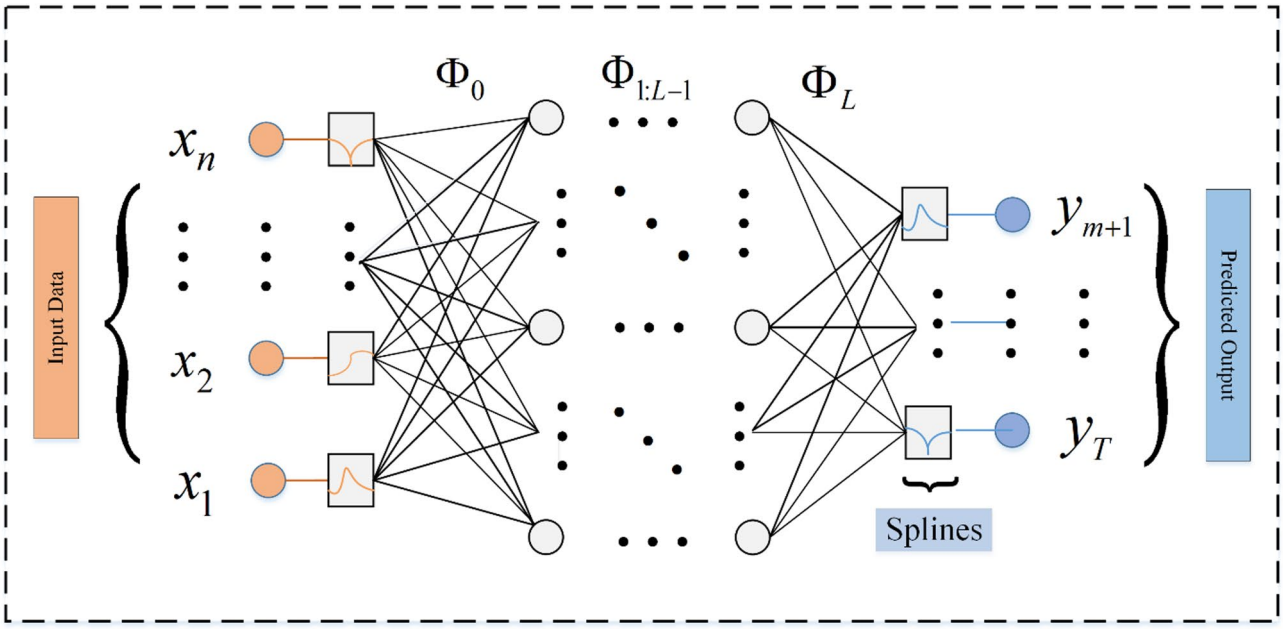

Fig. 2. Kolmogorov–Arnold Network representation with L number of layers.

Model	Forecast horizon	Configuration	Other parameters	Selected value
MLP	24 h	[RFx24, 300, 300, 24]	Epochs	200
	48 h	[RFx48, 300, 300, 48]	Optimizer	Adam
	96 h	[RFx96, 300, 300, 96]	Batch Size	32
	168 h	[RFx168, 300, 300, 168]	Activation	Fixed Relu
KAN	24 h	[RFx24, 30, 24]	Epochs	200
	48 h	[RFx48, 30, 48]	Optimizer	Adam
	96 h	[RFx96, 30, 96]	Batch Size	32
	168 h	[RFx168, 30, 168]	Activation	B-Spline, k = 3, G = 5

Table 2. Hyperparameters and configuration of the conventional models.

(48 h), the next four days (72 h) and the next week (168 h). A 24-step window is utilized, and the number of outputs N_{out} are comprised of nodes corresponding to the total amount of time steps in Eq. (1). Our KAN is represented by composition of these layers as

$$\mathrm{KAN}(X) = (\Phi_2 \circ \Phi_1 \circ \Phi_0) X \tag{10}$$

where Φ_2 transforms the outputs from the hidden layer into final predictions (N_{out}). Figure 2 illustrates a generic representation of a KAN network for any arbitrary number of layers L. The number of inputs N_{in} and number of outputs N_{out} are comprised of nodes corresponding to the total amount of time steps in Eq. (2) and (1) respectively. A detailed configuration and parameter settings of KAN and MLP models are provided in Table 2.

Optimization of KAN with SADE

The performance of KAN is highly sensitive to the choice of several key hyperparameters, particularly the grid size of the B-spline (G), the number of hidden layers (D), and the number of neurons per layer (W). Optimizing these parameters is essential to balancing model expressiveness with computational efficiency, especially in the context of STLF, where overfitting and poor generalization are common challenges. To simplify the optimization process, we fix certain settings to reduce the complexity of the search and focus on the most impactful parameters. These parameters control the expressive power and computational complexity of the model. Traditional methods such as gradient descent or cross-validation, often carry computational complexity and may not converge efficiently [47]. Therefore, a more robust SADE algorithm is used to optimize these parameters.

SADE extends the classical Differential Evolution (DE) framework by enabling the control parameters such as the mutation factor F and crossover rate C to adapt dynamically during the optimization process. This self-adaptive mechanism reduces the need for manual parameter calibration and allows the optimizer to adjust its behavior based on the historical success of mutations and recombinations, thereby improving convergence

Algorithm 1: SADE for parameter optimization of KAN

1. Initialize the population

- Creat population $P = X_1, X_2, \ldots, X_{Np}$, where each $X_i = G_i, D_i, W_i$
- Sample $G_i \sim U(3\text{-}12)$, $D_i \sim U(1\text{-}3)$, $W_i \sim U(30,40,50,60)$
- Initialize control parameters: $F_i \in [0.4,0.9]$, $C_i \in [0.1,0.9]$

2. **For each generation** g

- For each individual X_i^g in P
  - **Mutaion:**
    $V_i^{g+1} = X_{r1}^g + F_i(X_{r2}^g - X_{r3}^g)$
    Adapt $F_i = clip(F_{mean} + N(0, \sigma F))$
  - **Crossover**
    $U_i^{g+1} = crossover(X_i^g, V_i^{g+1}, C_i)$
    Adapt $C_i = clip(C_{mean} + N(0, \sigma C))$
  - **Selection**
    If $f(U_i^{g+1}) \leq f(X_i^g)$, set $X_i^{g+1} = U_i^{g+1}$
- Update F_{mean} and C_{mean}

3. Repeat until convergence or max generations
4. Return the best solution for G_i, D_i, W_i

Table 3. Implementation steps of SADE.

Model	Hyperparameters	Tuning range/value	Selected value
SADE-KAN	Grid size—G	[3–12]	10
	Hidden layer (Depth)	[1,2,3]	2
	Neurons/hidden layer (Width)	[30,40,50,60]	50
	Epochs	Fixed	200
	Optimizer	Fixed	Adam
	Batch Size	Fixed	32
	Activation	B-Spline	k = 3, G = 10

Table 4. Optimized hyperparameters of KAN with SADE.

and robustness. The optimization objective is to minimize the mean squared error (MSE) on the validation set by identifying the most effective configuration for the KAN model. The parameter search space was defined according to Table 3, where the grid size G was constrained to integer values between 4 and 12, the network depth D could take values in the discrete set [1–3], and the network width W was allowed to vary over [30,40,50,60]. To reduce complexity, other settings such as the spline type (fixed as cubic), the number of epochs (200), and knot positioning (uniform) were held constant to reduce the dimensionality of the search space and enhance convergence reliability.

The SADE algorithm was initialized with a population size of 30. For each individual X_i, the associated control parameters such as the mutation factor F_i and crossover rate C_i were independently initialized by sampling from the ranges $F_i \in [0.4,0.9]$ and $C_i \in [0.1,0.9]$. Unlike classical DE, where F and C are typically fixed for all individuals, SADE treats these as self-adaptive parameters. During the evolutionary process, successful values of F_i and C_i are tracked, and their distributions are updated using Gaussian perturbations centered around their historically successful means. This allows the algorithm to adaptively fine-tune its exploration and exploitation strategies across generations without external parameter tuning.

At each generation, SADE applies mutation and crossover operators to produce new candidate solutions. For each individual in the population, a mutant solution is created by combining information from multiple randomly selected individuals in the current population. This mutant is then combined with the original solution to produce a trial solution through a crossover operation. The trial solution is evaluated using the same validation MSE metric. If it demonstrates better performance than the original solution, it replaces it in the next generation. This selection mechanism ensures that only equal or superior solutions are retained, guiding the population toward optimal hyperparameter configurations over successive iterations [48]. The process iterates until a stopping criterion is met, either a maximum number of generations or no improvement over several consecutive generations. The full SADE implementation for KAN optimization is summarized in Table 3.

The final optimized KAN configuration selected by SADE consisted of a grid size of 10, network depth of 2 hidden layers, and 50 neurons per layer. These settings, summarized in Table 4, were used for all subsequent forecasting experiments reported in this study. By leveraging the adaptive mechanisms of SADE, the optimization

achieved a favorable trade-off between model accuracy and computational overhead of KAN. Figure 3 shows the flow chart of the SADE-KAN model for STLF.

Dataset and preprocessing

The dataset utilized in this study is collected from the ISO New England Control Area (ISO-NE-CA) publicly available data. It comprises hourly samples of weather data, system load, and other relevant information for all the wholesale load zones of ISO-NE-CA. The dataset is sampled every hour and covers a period of five years from Jan 2019 to Dec 2023, resulting in approximately one million samples. The data is arranged month wise, i.e. the data is structured in a month wise sequence, where observations are grouped by calendar month across all years. Such data arrangement facilitates temporal analysis, allowing for the examination of seasonal variations and trends within individual months. The data attributes are detailed in Table 5.

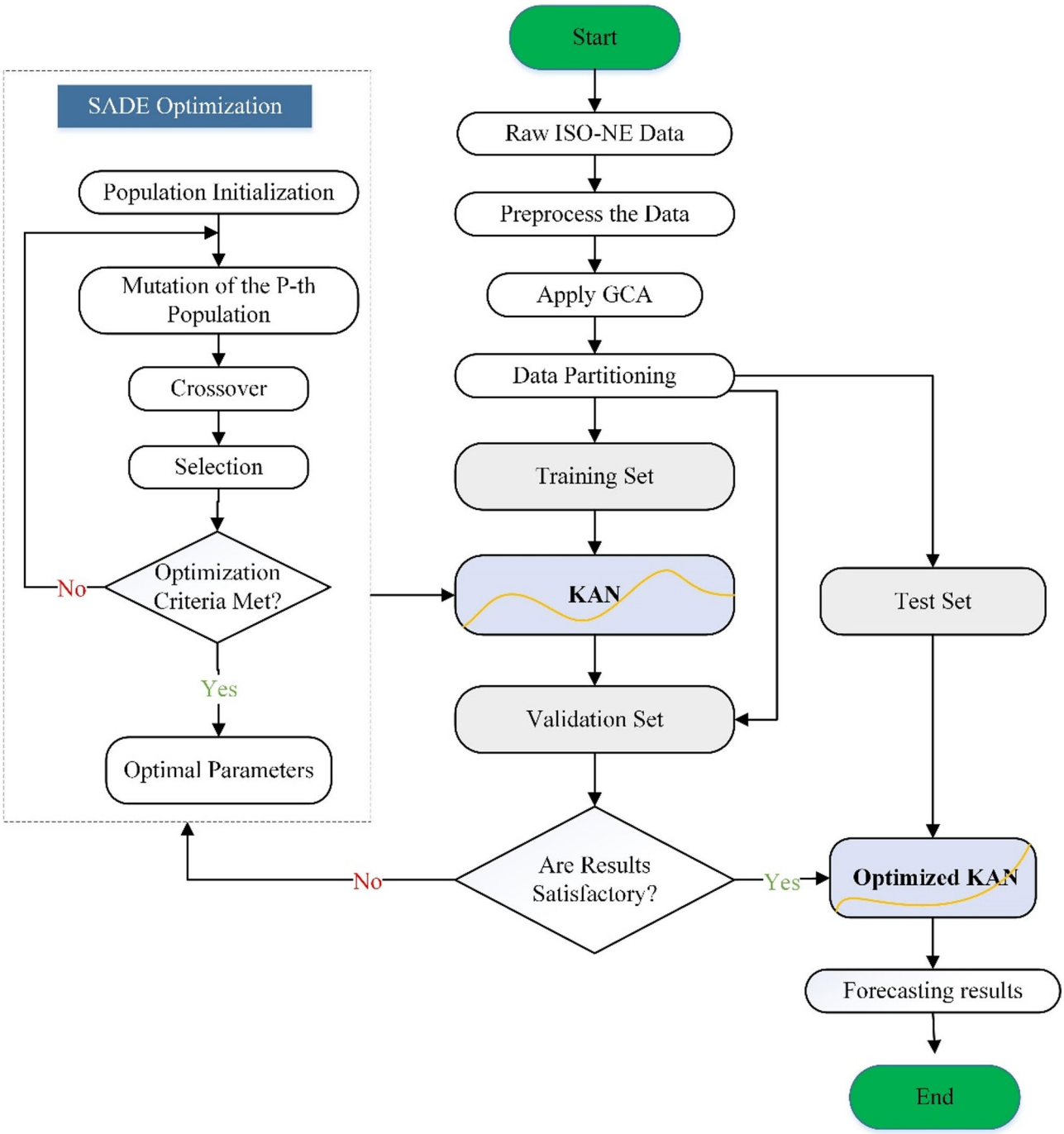

Fig. 3. Flowchart of the proposed SADE-KAN STLF method.

Feature	Description
Date	The calendar date of data
Hr_End	Observation at hour ending
DA_Demand	Day-ahead cleared demand in MW
RT_Demand	Real-time demand in MW
DA_LMP	Day-ahead locational marginal price (LMP) in $/MWh
DA_EC	Energy component of day-ahead LMP in $/MWh
DA_CC	Congestion component of day-ahead LMP in $/MWh
DA_MLC	Marginal loss component of day-ahead LMP in $/MWh
RT_LMP	Real-time locational marginal price (LMP) in $/MWh
RT_EC	Energy component of real-time LMP in $/MWh
RT_CC	Congestion component of real-time LMP in $/MWh
RT_MLC	Marginal loss component of real-time LMP in $/MWh
Dry_Bulb	The dry-bulb temperature in °F for the weather stations
Dew_Point	The dewpoint temperature in °F for the weather stations
System_Load	The actual New England system load in MW
Reg_SP	Regulation market service clearing price in $/MWh
Reg_CP	Regulation market capacity clearing price in $/MWh
Min_5min_RSP	The lowest pool level 5-min regulation service price in $/MWh
Max_5min_RSP	The highest pool level 5-min regulation service price in $/MWh
Min_5min_RCP	The lowest pool level 5-min regulation capacity price in $/MWh
Max_5min_RCP	The highest pool level 5-min regulation capacity price in $/MWh

Table 5. Description of the data attributes.

To broadly understand the system load which is our target variable, we decomposed it into trend, seasonal, and residual components, as shown in Fig. 4. The trend component represents the long-term behavior of the load, capturing underlying changes. Meanwhile, the seasonal component shows recurring patterns that repeat over shorter periods, often daily or weekly and the residual component highlights deviations from the trend and seasonal patterns, indicating irregularities. This decomposition helps in identifying the underlying patterns and variations in our target variable.

Input data preparation

To efficiently implement our proposed model, proper data preparation is essential. Data preparation plays a critical role in the performance of both ML and non-ML models. This is particularly crucial when employing the proposed SADE-KAN model that incorporates unique learnable activation functions and are sensitive to the scale of input data. Due to the sensitivity of the model to the scale of the data and the varying degrees of influence that different features exert on the final predicted load, we applied two standard preprocessing steps to our dataset.

First, we normalize our data by implementing MinMax scaling technique. This scaler transforms data to lie within a specified range, typically [0, 1], ensuring that each feature is scaled uniformly. This approach is particularly advantageous for neural networks, which are sensitive to variations in the magnitude of input data. Unlike standard normalization, which adjusts data to have a mean of zero, MinMax scaling retains the original zero reference point of the series. This is commonly achieved by dividing each value by the maximum observed value in the training dataset, as:

$$x_{ij}\prime = \frac{x_{ij} - min\left(x_j\right)}{max\left(x_j\right) - min\left(x_j\right)}, \forall i\left[1, m\right], j\left[1, n\right] \tag{11}$$

where $x_{ij}\prime$ is the normalized value of the jth feature at the ith timestamp, and $min\left(x_j\right)$ and $max\left(x_j\right)$ are the minimum and maximum values of the jth feature respectively. This method maintains consistent scaling across input features, preventing problems such as activation function saturation and numerical instability during computations.

Second, the GCA technique is applied to compute the grey correlation coefficients for each feature. These coefficients quantify the similarity between the feature sequences and the target load sequence, even amidst noise or complex interactions. The grey coefficient is calculated as [1]

$$\xi_{ij} = \frac{\Delta_{min} + \rho\Delta_{max}}{\Delta_{ij} + \rho\Delta_{max}}, \forall i\left[1, m\right], j\left[1, n\right] \tag{12}$$

where, Δ_{ij} is the absolute difference between the target sequence and the jth feature at timestamp i, Δ_{min} and Δ_{max} are the minimum and maximum differences across all time points, and ρ is the distinguishing coefficient

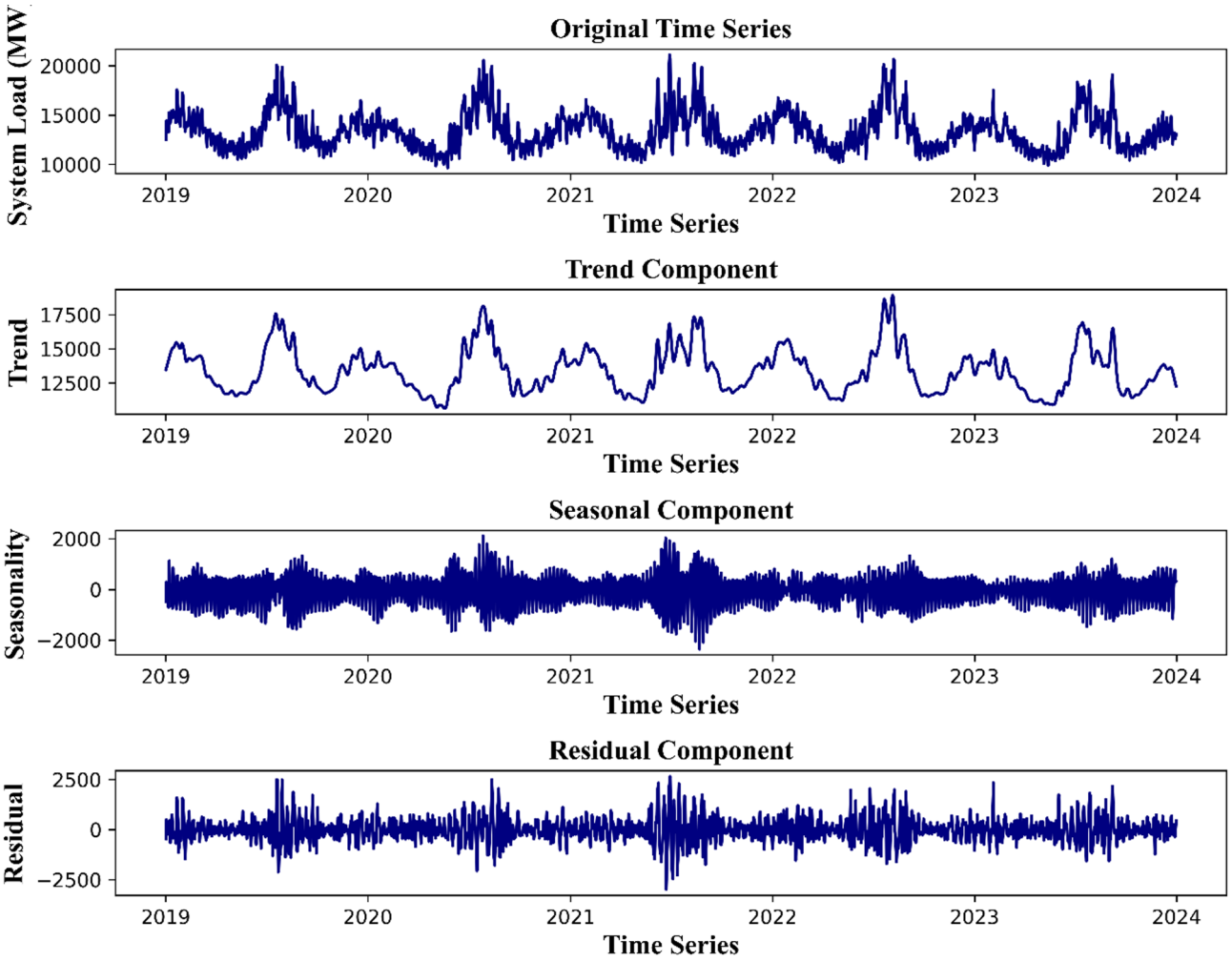

Fig. 4. The original load sequence and its components from Jan 2019 to Dec 2023.

usually set to 0.5 [1]. Finally, the grey correlation grades for each feature can be computed by averaging the grey correlation coefficients across all time points as

$$G_j = \frac{1}{m}\sum_{i=1}^{m}\xi_{ij}, \forall j\,[1,n] \tag{13}$$

GCA helps in focusing on the most relevant features by assessing the similarity degree between the feature sequences and the target variable, thereby reducing data dimensionality and eliminating irrelevant features. The grey correlation grades for each feature are calculated and visualized in Fig. 5. This visualization underscores the importance of each feature and its direct relationship with the system load, providing a clear rationale for their inclusion in the model. Features with higher grey correlation grades are considered more relevant to the target variable, while those with lower grades are deemed less important and are dropped from the analysis. After filtering out the irrelevant features, the selected features are fed into our KAN model for further processing. We evaluated the performance of our KAN model using different sets of features selected based on varying grey correlation grade thresholds. The optimal results were achieved with a grey correlation grade of 0.7. This threshold guarantees that only the most significant features are retained, resulting in improved accuracy and efficiency of the proposed SADE-KAN model.

Experimental setup and results

To evaluate the effectiveness of the proposed SADE-KAN model, a simulation environment was developed in Python, following the architectural specifications outlined in Sections "Kolmogorov arnold network" and "Optimization of KAN with SADE". All experiments were executed using Python version 3.12.0 on a Windows 11 platform, running on a 2.60 GHz 64-bit Intel Core i5-4210U processor with 12 GB RAM and a 500 GB hard drive. The hourly load data from ISO-NE, preprocessed as described in Section "Dataset and preprocessing", served as the input to the forecasting model. Data partitioning details are provided in Table 6. The performance

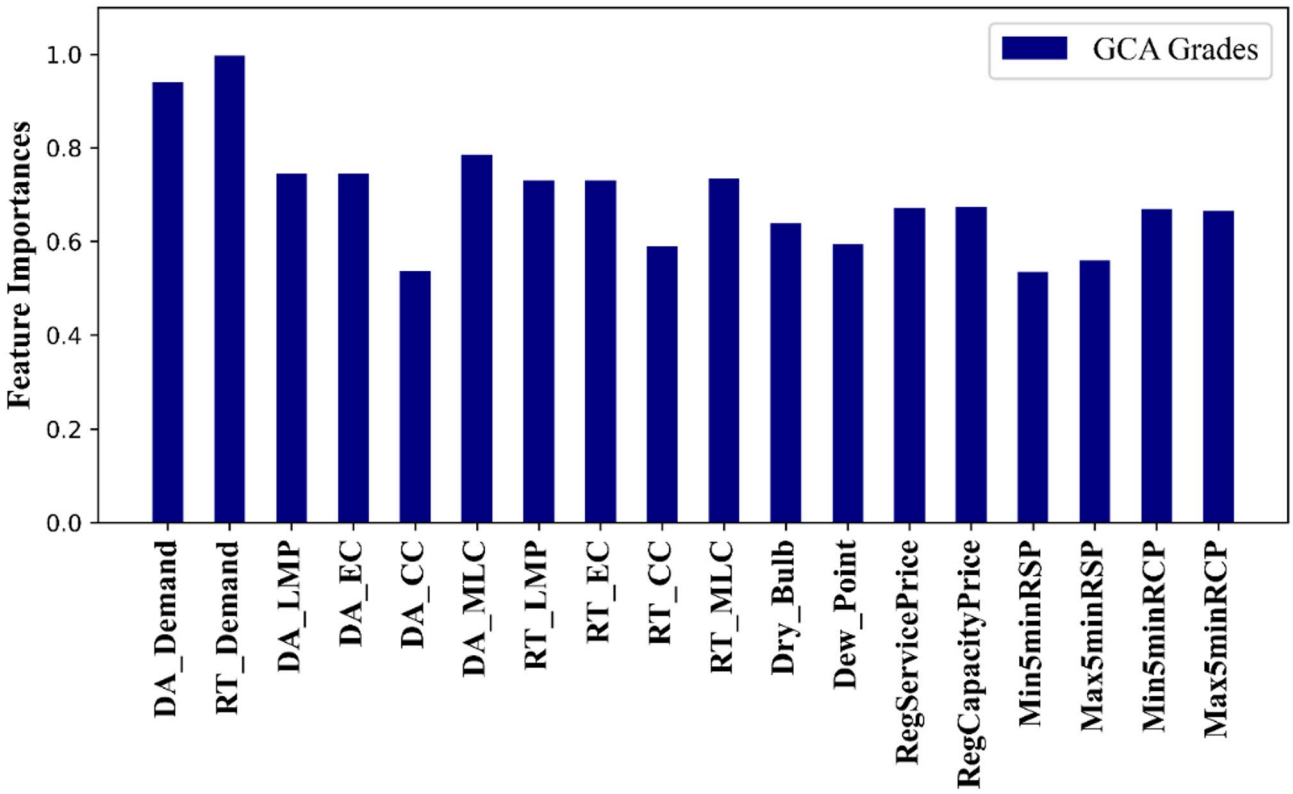

Fig. 5. Grey correlation grades given to each feature.

Dataset	Training set	Validation set	Testing set
From	01/01/2019 01:00	01/01/2023 01:00	01/07/2023 01:00
To	31/12/2022 00:00	30/06/2023 00:00	31/12/2023 00:00

Table 6. Division of training, validation and testing sets of the data.

of SADE-KAN was benchmarked against conventional MLP, the original KAN, and several recently proposed state-of-the-art models. Evaluation was based on key performance indicators, including mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), training time, and model complexity (measured by the number of parameters), across multiple forecasting horizons.

The experiments were structured to systematically compare the performance of these models with differing architectures, in order to comprehensively assess their impact on forecasting accuracy, training efficiency, and model complexity. Each model is trained for 200 epochs using the Adam optimizer with a learning rate of 0.001 and mean squared error (MSE) as the training and validation loss function calculated as [49]

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{Y} - Y\right)^2 \tag{14}$$

The evaluation metrics used in this experiment to assess the performance of these networks are MAE, RMSE, and MAPE, which are calculated as [49]

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\hat{Y} - Y\right|}{Y} \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{Y} - Y\right)^2} \tag{16}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\hat{Y} - Y\right|}{Y} \times 100\% \tag{17}$$

where N denotes number of samples, $\hat{Y}$ and Y represents the predicted and corresponding actual load respectively. By averaging the squared differences between observed and predicted values, the MSE offers a precise evaluation of model performance. The squaring step penalizes larger errors more heavily, making the metric suitable for addressing outliers and significant deviations. The MAPE quantifies the average deviation between the predicted values and the actual values, expressed as a percentage. The MAE, on the other hand, calculates the average of the absolute differences between predicted and actual values, providing a direct and interpretable measure of the average forecasting error. In contrast, the RMSE measures the standard deviation of the differences between the predicted and actual values, providing insight into the magnitude of prediction errors. Lower values of MAE, MAPE and RMSE indicate superior model performance, showing more accurate and reliable forecasts.

Results and comparative assessment

The evaluation results of the SADE-KAN model reveal significant insights into its performance in comparison to the traditional MLP, the original KAN and recently published advanced models. The results indicate that the predictions obtained using our proposed model better approximate the actual system load across all forecasting horizons, demonstrating robust generalization to unseen data. This is particularly evident in Fig. 6 which depicts the training and validation loss curves across all forecast horizons. The curves reveal a rapid convergence and robust learning of the underlying data patterns without overfitting. The training loss demonstrates a steady decrease with each epoch, while the validation loss follows a similar trajectory which shows robust generalization to unseen data. This stability in the learning process proves the robustness and reliability of the proposed SADE-KAN.

Additionally, Fig. 7 illustrates the 24-h forecasting performance, where SADE-KAN captures rapid changes in system load more effectively than both the KAN and MLP models. While the original KAN and MLP show occasional over-prediction or under-predictions, especially in the middle hours of the forecast horizon, SADE-KAN adapts more quickly to sudden shifts in load demand. This enhanced adaptability demonstrates suitability of the SADE-KAN for dynamic load forecasting, which is a critical aspect of effective load management.

Further, the capability of SADE-KAN to capture dynamic load variations is evident in the load forecasts for the next 48 h, 96 h and 168 h as illustrated in Figs. 8, 9, and 10 respectively. SADE-KAN consistently outperforms both the original KAN and MLP models in capturing rapid load fluctuations, especially at longer forecast

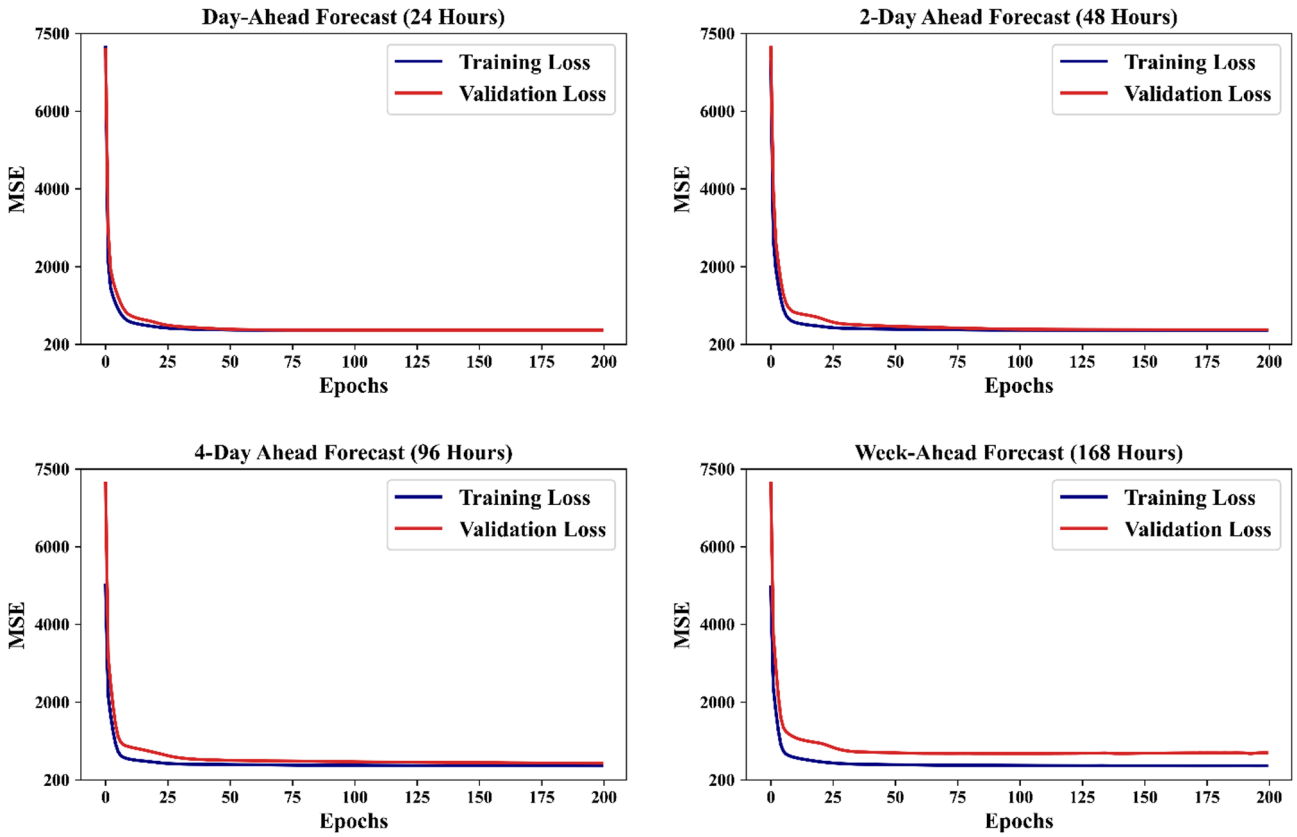

Fig. 6. KANs learning evaluation on ISO-NE-CA hourly data.

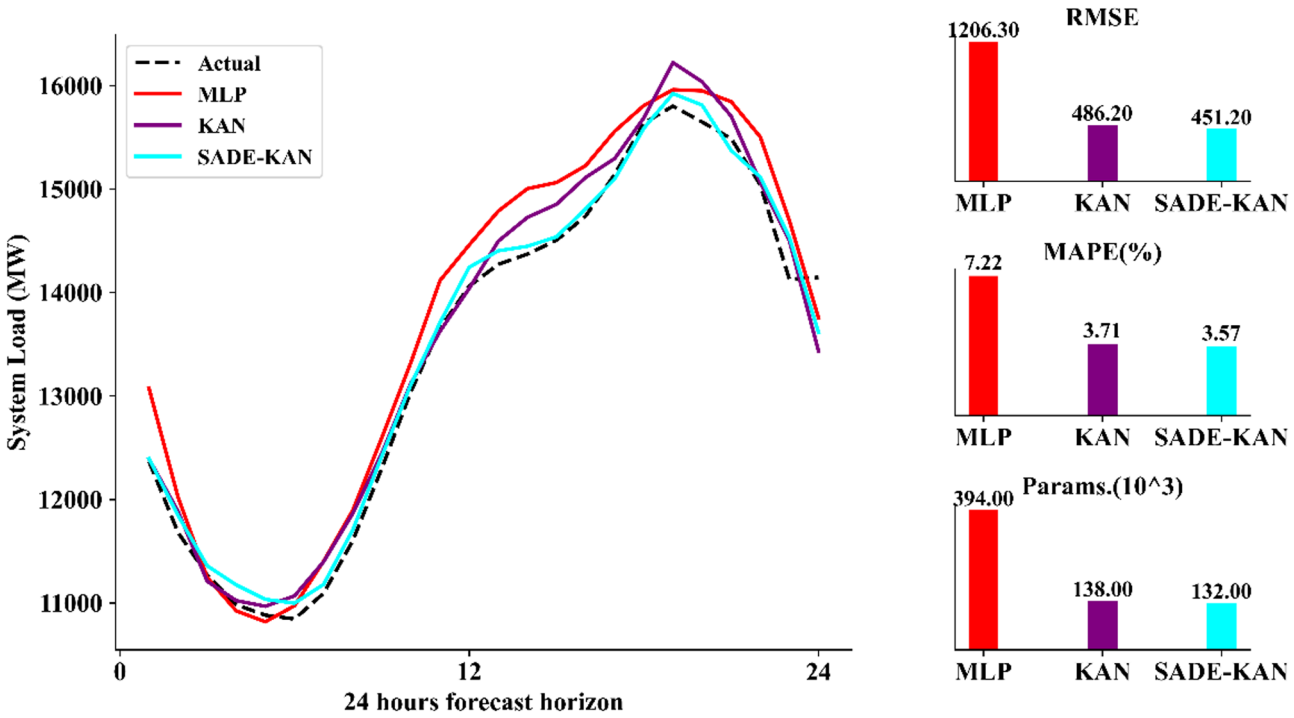

Fig. 7. One day ahead forecasted load with hour resolution using MLP, KAN and SADE-KAN.

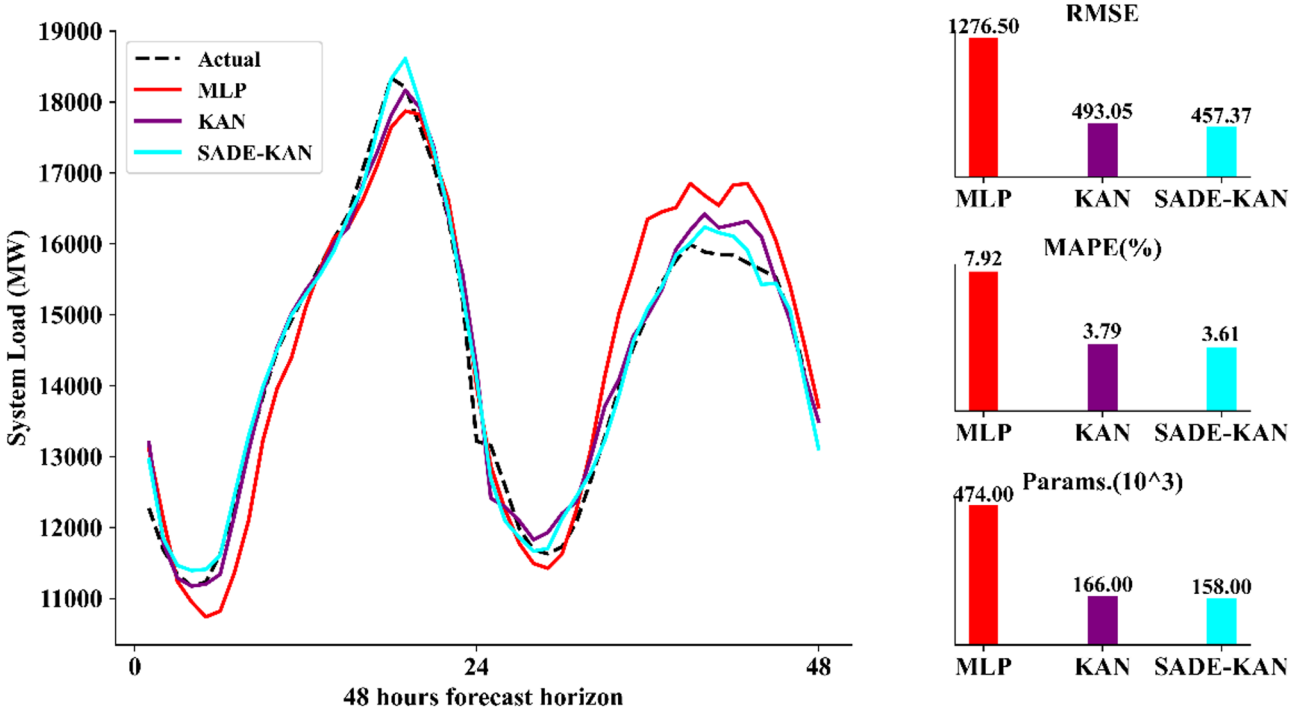

Fig. 8. Two days ahead forecasted load with hour resolution using MLP, KAN and SADE-KAN.

horizons. While KAN demonstrates a robust capacity to model load variations, it shows some limitations in tracking abrupt changes compared to the enhanced adaptability of SADE-KAN. In contrast, the MLP model exhibits a noticeable lag in responding to these variations, particularly over the longer horizons, where it struggles to capture immediate shifts in load. The ability of SADE-KAN to adjust quickly and accurately to these dynamic changes across longer forecasting horizons not only demonstrates its robustness but also makes it a more reliable and effective model for STLF.

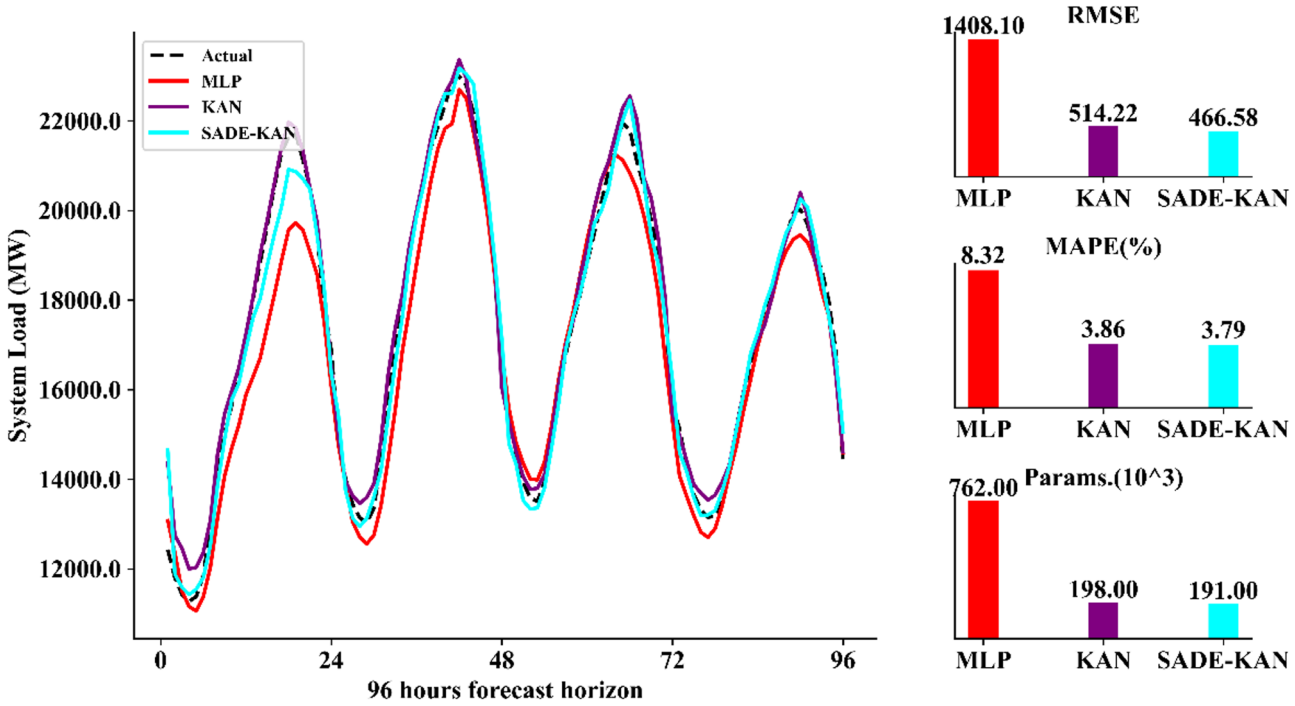

Fig. 9. Four days ahead forecasted load with hour resolution using MLP, KAN and SADE-KAN.

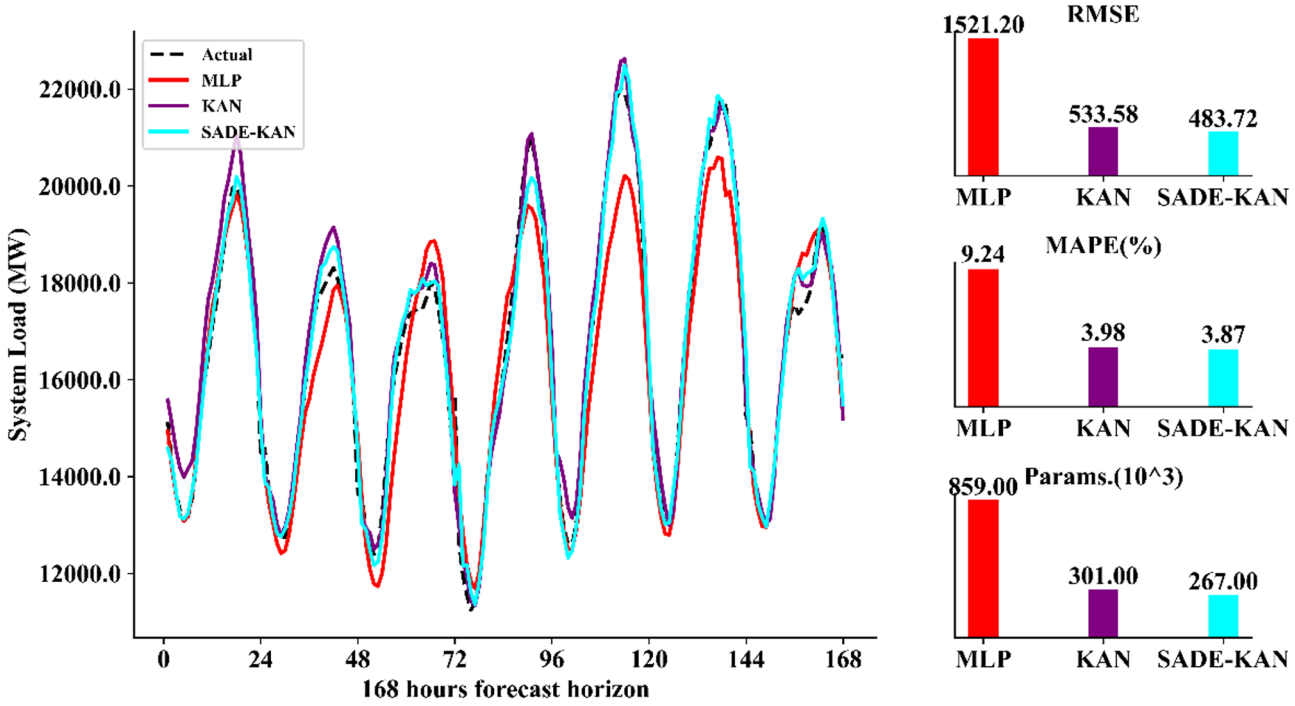

Fig. 10. Week ahead forecasted load with hour resolution using MLP, KAN and SADE-KAN.

To comprehensively quantify the performance of our proposed model, a thorough comparison with MLP and KAN is presented in Table 7. The table displays several key performance metrics including, MAPE, MAE, RMSE, training time, and the count of trainable parameters across four forecasting horizons, 24, 48, 96, and 168 h. Across all horizons, SADE-KAN consistently achieves the lowest error values, with a MAPE ranging from 3.57% (24-h horizon) to 3.87% (168-h horizon), compared to 3.71–3.98% for KAN and a significantly higher 7.22–9.24% for MLP. Similarly, MAE and RMSE values follow the same trend. For instance, at the 168-

Model	Horizon	MAPE	MAE	RMSE	Training time (Sec)	Parameters (10^3)
MLP	24 h	7.22	957.58	1206.3	147.8	394
	48 h	7.92	986.24	1276.5	176.9	474
	96 h	8.32	1055.7	1408.1	224.7	762
	168 h	9.24	1176.2	1521.2	382.2	859
KAN	24 h	3.71	393.25	486.21	1113.7	138
	48 h	3.79	402.12	493.05	1590.2	166
	96 h	3.86	416.27	514.22	1862.6	198
	168 h	3.98	435.75	533.58	2697.2	301
SADE-KAN	24 h	3.57	353.69	451.22	1519.9	132
	48 h	3.61	366.37	457.37	1712.4	158
	96 h	3.79	379.29	466.58	1819.9	191
	168 h	3.87	397.16	483.72	2307.2	267

Table 7. SADE-KAN evaluation results on the testing set compared to MLP and Conventional KAN.

GCA Grade	0.5	0.6	0.7	0.74	0.9	1.0
RMSE (MW)	1296.78	755.23	451.22	492.18	583.29	N/A
MAE (MW)	989.65	523.96	353.69	386.92	421.84	N/A
MAPE (%)	6.9	4.02	3.57	3.87	6.52	N/A
No. of input features	18	13	8	5	2	0

Table 8. Performance of the SADE-KAN in varying input features.

h horizon, SADE-KAN achieves an RMSE of 483.72 MW and an MAE of 397.16 MW, outperforming KAN (533.58 MW RMSE, 435.75 MW MAE) and substantially surpassing MLP (1521.2 MW RMSE, 1176.2 MW MAE). This consistent accuracy across increasing horizons underlines robustness and generalization capability of the SADE-KAN. Particularly, while MLP suffers a sharp degradation in performance as the horizon lengthens, SADE-KAN maintains stable accuracy with only marginal increases in error metrics.

In terms of computational cost, SADE-KAN incurs higher training times than MLP but remains comparable to or slightly more efficient than the original KAN. For example, at the 96-h horizon, SADE-KAN completes training in 1819.9 s, whereas KAN requires 1862.6 s. Moreover, SADE-KAN achieves this improved performance with a slightly reduced number of trainable parameters compared to KAN, further highlighting the efficiency of the proposed model.

A significant advantage of SADE-KAN is its performance in both accuracy and parameter efficiency. With only 132,000 trainable parameters for the 24-h forecasting horizon, SADE-KAN significantly reduces model complexity compared to the MLP, which requires 394,000 parameters for the same task. Across all forecasting horizons, SADE-KAN utilizes, on average, approximately 35% of the parameters used by the MLP, while also improving upon parameter count of the original KAN. This substantial reduction in model size indicates that SADE-KAN can deliver superior forecasting performance with lower computational overhead which is a critical attribute for deployment in resource-constrained environments.

In terms of performance, SADE-KAN outperforms MLP significantly across all forecast horizons, delivering lower error metrics. Moreover, SADE-KAN enhances the original KAN by refining its training process and boosting its ability to capture complex load dynamics. These improvements ensure that our proposed model provides more reliable and precise forecasts.

In terms of accuracy, SADE-KAN consistently outperforms the MLP across all forecast horizons, achieving significantly lower MAPE, MAE, and RMSE values. Furthermore, it enhances the original KAN by optimizing its training process and improving its capacity to model complex load patterns. These enhancements result in more precise and reliable forecasts, particularly for extended horizons where traditional models often degrade in performance.

While the proposed model achieves these gains, it still requires longer training time than MLP. This remains the only area where MLP holds an advantage. However, compared to the original KAN, SADE-KAN slightly reduces the training time, offering improvement in computational efficiency. Despite the longer training duration relative to MLP, the substantial improvements in forecast accuracy and reduced error rates make SADE-KAN a much more effective model for STLF practices.

Input features sensitivity analysis

The performance of the SADE-KAN model varies significantly with the number of input features used, as illustrated in Table 8. In this table, the GCA grade represents the proportion of features retained, and its effect on model performance is analyzed across all forecasting horizons. Initially, with 18 input features with GCA grade 0.5, the model exhibits relatively high error metrics, with an RMSE of 1296.78 MW, MAE of 989.65 MW, and MAPE of 6.9%. However, as the number of features is reduced, its performance improves, reaching optimal

Forecast Horizon	MAPE (%)	MAE (MW)	RMSE (MW)	Run. Time (Sec)
24 Hours	3.57	353.69	451.22	1519.9
48 Hours	3.61	366.37	457.37	1712.4
96 Hours	3.79	379.29	466.58	1819.9
168 Hours	3.87	397.16	483.72	2307.2

Table 9. Performance of the SADE-KAN in extended horizons.

Algorithm	RMSE (MW)	MAE (MW)	MAPE (%)	Run. Time (Sec)
Bi-LSTM [50]	562.25	408.14	4.62	N/A
CNN-LSTM [50]	483.21	370.51	4.24	N/A
XGBoost [51]	943.98	668.32	12.7	3643.24
LGBM [51]	700.67	495.06	3.85	3664.78
XGB + GBR + SVR + KNN [52]	2920.7	2282.1	3.69	1234.12
SADE-KAN	451.22	353.69	3.57	1519.9

Table 10. Performance of the SADE-KAN compared to recently published advanced models.

accuracy at GCA grade 0.7 with 8 features. At this level, the RMSE drops to 451.22 MW, MAE to 353.69 MW, and MAPE to 3.57%, indicating that a moderate reduction in input features enhances its ability to forecast load with greater precision.

Further reduction in the number of features, as seen at GCA grades 0.74 and 0.9, leads to a slight deterioration in performance. The RMSE increases to 492.18 MW and 583.29 MW, respectively, while MAPE rises to 3.87% and 6.52%. These results suggest that while reducing input features can improve model performance up to a point, there is a threshold beyond which accuracy declines due to the loss of critical information that is necessary for effective forecasting.

Performance in longer forecast horizons

The forecasting capability of SADE-KAN across extended time horizons is presented in Table 9. This evaluation aims to assess its temporal generalization ability and its capacity to maintain predictive accuracy as the forecast window is stretched from 24 to 168 h.

At the 24-h horizon, SADE-KAN achieves a MAPE of 3.57%, MAE of 353.69 MW, and RMSE of 451.22 MW, establishing a strong baseline performance. As the forecasting horizon extends to 48, 96, and 168 h, the model exhibits a gradual increase in error, which is expected due to the cumulative uncertainty associated with longer temporal predictions. However, the degradation is minimal and consistent, indicating its robustness. Specifically, at 168 h, the MAPE increases only to 3.87%, with MAE and RMSE rising moderately to 397.16 MW and 483.72 MW, respectively. This marginal increase in forecasting error over a 7-day horizon is a noteworthy result. It suggests that SADE-KAN is capable of learning complex temporal dependencies and preserving stability over extended prediction intervals. Additionally, while the runtime increases with the horizon, growing from 1519.9 s for 24 h to 2307.2 s for 168 h, the computational cost remains within practical limits.

Importantly, the model avoids the steep performance degradation typically seen in conventional architectures such as MLP when extrapolated to multi-day forecasts. These results affirm that SADE-KAN not only delivers high accuracy for immediate forecasts but also sustains performance across longer-term horizons, making it suitable for diverse STLF applications with varying planning requirements.

Performance compared to advanced models

To further establish the efficacy and competitiveness of the proposed SADE-KAN model, its forecasting performance was benchmarked against several recently published advanced machine learning and deep learning approaches. The comparison, summarized in Table 10, includes state-of-the-art models such as Bi-directional LSTM (Bi-LSTM) [50], CNN-LSTM hybrid [50], extreme gradient boosting (XGBoost) [51], light gradient boosting machine (LightGBM) [51], and an ensemble of XGB, gradient boosting regressor (GBR), support vector regressor (SVR) and K nearest neighbors (KNN) reported in [52].

SADE-KAN consistently outperforms all compared models across key accuracy metrics. It achieves the lowest MAPE of 3.57%, indicating superior relative error performance. In terms of absolute metrics, SADE-KAN attains an RMSE of 451.22 MW and an MAE of 353.69 MW, both of which are the lowest among all models presented. While the ensemble method reports a comparable MAPE of 3.69%, its extremely high RMSE of 2920.7 MW and MAE of 2282.1 MW expose significant absolute deviation, suggesting poor generalization despite a favorable percentage error.

Compared to Bi-LSTM and CNN-LSTM, SADE-KAN offers clear gains in accuracy. It reduces MAPE by 22.7% and 15.8% relative to Bi-LSTM and CNN-LSTM, respectively, while also improving RMSE and MAE. Particularly, SADE-KAN also demonstrates a more efficient runtime profile than gradient boosting models such as XGBoost and LGBM, which require over 3600 s for training. SADE-KAN completes training in 1519.9 s, offering a practical balance between accuracy and computational cost.

In summary, SADE-KAN surpasses the accuracy of deep learning and ensemble-based models, while maintaining computational feasibility. These results underscore its effectiveness and suitability for real-world STLF applications where both precision and efficiency are critical.

Conclusion

In summary, this study proposed the Kolmogorov–Arnold network optimized with Self-Adaptive Differential Evolution (SADE-KAN) for short-term load forecasting (STLF). By substituting the static linear weights of conventional MLP with adaptive spline-based activation functions, the KAN architecture effectively captures nonlinear dependencies in power system load data. Its optimization using SADE further enhances learning efficiency, mitigates overfitting through dynamic parameter tuning, and reduces complexity.

The experimental results revealed that SADE-KAN consistently delivered superior accuracy across all tested forecasting horizons (24, 48, 96, and 168 h), significantly outperforming both MLP and the original KAN in terms of RMSE, MAE, and MAPE. Particularly, SADE-KAN achieves the lowest MAPE values, ranging from 3.57% at the 24-h horizon to 3.87% at the 168-h horizon, compared to 3.71–3.98% for KAN and 7.22–9.24% for MLP. The same trend is evident in MAE and RMSE values. This consistent performance across increasingly long horizons demonstrates robustness and superior generalization ability of the proposed model, particularly where traditional models like MLP show noticeable performance degradation. In addition to accuracy, the model demonstrates remarkable parameter efficiency. For example, SADE-KAN requires only 132 k trainable parameters at the 24-h horizon, substantially fewer than the 394 k used by MLP, which implies reduced computational complexity and resource demand. Furthermore, the proposed model was benchmarked against several recently published advanced methods, including Bi-LSTM, CNN-LSTM, XGBoost, and ensemble hybrid models. SADE-KAN outperformed all of them in accuracy, achieving a MAPE of 3.57%, MAE of 353.69 MW, and RMSE of 451.22 MW, while also demonstrating more favorable computational efficiency than many of the ensemble and gradient boosting approaches.

While SADE-KAN demonstrates strong forecasting accuracy and parameter efficiency, several limitations remain. First, the model exhibits sensitivity to input feature, this implies that feature selection is crucial as excessive input dimensionality introduces redundancy that impairs learning. Second, SADE-KAN incurs higher training times compared to simpler models like MLP due to its spline-based activation functions. Lastly, the current evaluation of SADE-KAN is limited to ISO-NE load data, broader validation across diverse geographic regions, varying weather conditions, and different load patterns is essential to confirm its generalizability and robustness in real-world applications.

To address the limitations identified in this study, several technical enhancements can be pursued. Given its sensitivity to input dimensionality, advanced feature selection methods could be employed to retain only the most informative inputs. Alternatively, attention-based mechanisms or dimensionality reduction techniques may help reduce redundancy. To mitigate the relatively high training time, future work could explore simplifying the spline functions by reducing the number of knots or opting for less computationally intensive spline types such as linear or quadratic splines. This would reduce the number of trainable parameters and result in faster training times. To improve generalizability across regions and datasets, future research should focus on evaluating SADE-KAN across diverse datasets from different regions and power systems.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 18 March 2025; Accepted: 5 June 2025
Published online: 01 July 2025

References

1. Wang, K., Xu, C., Zhang, Y., Guo, S. & Zomaya, A. Y. Robust big data analytics for electricity price forecasting in the smart grid. *IEEE Trans. Big Data* **5**(1), 34–45 (2017).
2. Hong, T. et al. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. *Int. J. Forecast.* **32**(3), 896–913 (2016).
3. Hobbs, B. F. et al. Analysis of the value for unit commitment of improved load forecasts. *IEEE Trans. Power Syst.* **14**(4), 1342–1348 (1999).
4. Al-Suod, M. M. et al. Forecasting energy consumption of a mining plant using artificial neural networks. *IEEE Access* (2025).
5. R. Singh, A., Kumar, R. S., Bajaj, M., Khadse, C. B. & Zaitsev, I. Machine learning-based energy management and power forecasting in grid-connected microgrids with multiple distributed energy sources. *Sci. Rep.* **14**(1), 19207 (2024).
6. Alharbi, A. H. et al. Forecasting of energy efficiency in buildings using multilayer perceptron regressor with waterwheel plant algorithm hyperparameter. *Front. Energy Res.* **12**, 1393794 (2024).
7. Guermoui, M. et al. An analysis of case studies for advancing photovoltaic power forecasting through multi-scale fusion techniques. *Sci. Rep.* **14**(1), 6653 (2024).
8. Sharma, N. et al. Solar power forecasting beneath diverse weather conditions using GD and LM-artificial neural networks. *Sci. Rep.* **13**(1), 8517 (2023).
9. Mfetoum, I. M. et al. A multilayer perceptron neural network approach for optimizing solar irradiance forecasting in Central Africa with meteorological insights. *Sci. Rep.* **14**(1), 3572 (2024).
10. Reihanifar, M. et al. A new multi-objective genetic programming model for meteorological drought forecasting. *Water* **15**(20), 3602 (2023).
11. Mouloud, L. A. et al. Assessment of hybrid bidirectional deep learning models for short-term global horizontal irradiance forecasting. In *AIP Conference Proceedings* vol. 3214, no. 1 (AIP Publishing, 2024).
12. Alhussan, A. A. et al. Green hydrogen production ensemble forecasting based on hybrid dynamic optimization algorithm. *Front. Energy Res.* **11**, 1221006 (2023).

13. Al-qaness, M. A. et al. ResInformer: Residual transformer-based artificial time-series forecasting model for PM2.5 concentration in three major Chinese cities. *Mathematics* **11**(2), 476 (2023).
14. Saeed, M. A. et al. Forecasting wind power based on an improved al-Biruni Earth radius metaheuristic optimization algorithm. *Front. Energy Res.* **11**, 1220085 (2023).
15. Molu, R. J. J. et al. Advancing short-term solar irradiance forecasting accuracy through a hybrid deep learning approach with Bayesian optimization. *Results Eng.* **23**, 102461 (2024).
16. Kheldoun, A. et al. Seasonal forecasting of global horizontal irradiance for grid-connected PV plants: a combined CNN-BiGRU approach. In 2024 *3rd International conference on Power Electronics and IoT Applications in Renewable Energy and its Control (PARC)* 169–173 (IEEE, 2024).
17. Khelifi, R. et al. Short-term PV power forecasting using a hybrid TVF-EMD-ELM strategy. *Int. Trans. Electr. Energy Syst.* **2023**(1), 6413716 (2023).
18. Hong, T., & Shahidehpour, M. Load forecasting case study. *EISPC, US Department of Energy* (2015).
19. Mienye, I. D., Swart, T. G. & Obaido, G. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information* **15**(9), 517 (2024).
20. Arora, S. & Taylor, J. W. Rule-based autoregressive moving average models for forecasting load on special days: A case study for France. *Eur. J. Oper. Res.* **266**, 259–268 (2018).
21. Charlton, N. & Singleton, C. A refined parametric model for short term load forecasting. *Int. J. Forecast.* **30**(2), 364–368 (2014).
22. Dudek, G. Pattern-based local linear regression models for short term load forecasting. *Electric Power Syst. Res.* **130**, 139–147 (2016).
23. Takeda, H., Tamura, Y. & Sato, S. Using the ensemble Kalman filter for electricity load forecasting and analysis. *Energy* **104**, 184–198 (2016).
24. Biswas, A. K., Ahmed, S. I., Bankefa, T., Ranganathan, P., & Salehfar, H. Performance analysis of short and mid-term wind power prediction using ARIMA and hybrid models. In *2021 IEEE Power and Energy Conference at Illinois (PECI)* 1–7 (IEEE, 2021).
25. Smyl, S., Dudek, G. & Pełka, P. ES-dRNN: A hybrid exponential smoothing and dilated recurrent neural network model for short-term load forecasting. *IEEE Trans. Neural Netw. Learn. Syst*. (2023).
26. Benidis, K. et al. Deep learning for time series forecasting: Tutorial and literature survey. *ACM Comput. Surv.* **55**(6), 1–36 (2022).
27. Dudek, G. Neural networks for pattern-based short-term load forecasting: A comparative study. *Neurocomputing* **205**, 64–74 (2016).
28. Chen, K. et al. Short-term load forecasting with deep residual networks. *IEEE Trans. Smart Grid* **10**(4), 3943–3952 (2019).
29. Sadaei, H. J., de Lima e Silva, P. C., Guimarães, F. G. & Lee, M. H. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy* **175**, 365–377 (2019).
30. Ullah, K. et al. Short-term load forecasting: A comprehensive review and simulation study with CNN-LSTM hybrids approach. *IEEE Access* (2024).
31. Wang, S., Wang, X., Wang, S. & Wang, D. Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting. *Int. J. Electr. Power Energy Syst.* **109**, 470–479 (2019).
32. Abumohsen, M., Owda, A. Y. & Owda, M. Electrical load forecasting using LSTM, GRU, and RNN algorithms. *Energies* **16**(5), 2283 (2023).
33. Eren, Y. & Küçükdemiral, İ. A comprehensive review on deep learning approaches for short-term load forecasting. *Renew. Sustain. Energy Rev.* **189**, 114031 (2024).
34. Kong, W. et al. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans. Smart Grid* **10**(1), 841–851 (2017).
35. Smyl, S. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *Int. J. Forecast.* **36**(1), 75–85 (2020).
36. Wu, Q., Guan, F., Lv, C. & Huang, Y. Ultra-short-term multi-step wind power forecasting based on CNN-LSTM. *IET Renew. Power Gener.* **15**(5), 1019–1029 (2021).
37. Yang, Z., Ce, L. & Lian, L. Electricity price forecasting by a hybrid model, combining wavelet transform, ARMA and kernel-based extreme learning machine methods. *Appl. Energy* **190**, 291–305 (2017).
38. Marino, D. L., Amarasinghe, K. & Manic, M. Building energy load forecasting using deep neural networks. In *IECON 2016–42nd Annual Conference of the IEEE Industrial Electronics Society* 7046–7051 (2016).
39. Hong, W. C., Dong, Y., Zhang, W. Y., Chen, L. Y. & Panigrahi, B. K. Cyclic electric load forecasting by seasonal SVR with chaotic genetic algorithm. *Int. J. Electr. Power Energy Syst.* **44**(1), 604–614 (2013).
40. Liu, Z. et al. Kan: Kolmogorov–Arnold networks. Preprint http://arxiv.org/abs/2404.19756 (2024).
41. Abbas, M., Che, Y. & Zafar, A. Utilizing the new Kolmogorov-Arnold network for short-term electric load forecasting. *J. Renew. Sustain. Energy* **17**(2), 025501 (2025).
42. Vaca-Rubio, C. J., Blanco, L., Pereira, R., & Caus, M. Kolmogorov-arnold networks (KANs) for time series analysis. Preprint at http://arxiv.org/abs/2405.08790 (2024).
43. Kiamari, M., Kiamari, M., & Krishnamachari, B. GKAN: Graph Kolmogorov-Arnold networks. Preprint at http://arxiv.org/abs/2406.06470 (2024).
44. Seydi, S. T. (2024). Exploring the potential of polynomial basis functions in Kolmogorov-Arnold networks: A comparative study of different groups of polynomials. *arXiv preprint* arXiv:2406.02583.
45. Zeydan, E. et al. F-KANs: Federated Kolmogorov-Arnold networks. Preprint at http://arxiv.org/abs/2407.20100 (2024).
46. Askari, M. & Keynia, F. Mid-term electricity load forecasting by a new composite method based on optimal learning MLP algorithm. *IET Gener. Transm. Distrib.* **14**(5), 845–852 (2020).
47. Tang, X., Dai, Y., Wang, T. & Chen, Y. Short-term power load forecasting based on multi-layer bidirectional recurrent neural network. *IET Gener. Transm. Distrib.* **13**(17), 3847–3854 (2019).
48. Wang, Z., Ku, Y. & Liu, J. The power load forecasting model of combined SaDE-ELM and FA-CAWOA-SVM based on CSSA. *IEEE Access* **12**, 41870–41882 (2024).
49. Yadav, A., Bareth, R., Kochar, M., Pazoki, M. & Sehiemy, R. A. E. Gaussian process regression-based load forecasting model. *IET Gener. Transm. Distrib.* **18**(5), 899–910 (2024).
50. Zhang, G., Wei, C., Jing, C. & Wang, Y. Short-term electrical load forecasting based on time augmented transformer. *Int. J. Comput. Intell. Syst.* **15**(1), 67 (2022).
51. Massaoudi, M. et al. A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for short-term load forecasting. *Energy* **214**, 118874 (2021).
52. Junior, M. Y. et al. Optimized hybrid ensemble learning approaches applied to very short-term load forecasting. *Int. J. Electr. Power Energy Syst.* **155**, 109579 (2024).

Author contributions

Muhammad Abbas, Yanbo Che, Sarmad Maqsood, Muhammad Zain Yousaf: Conceptualization, Methodology, Software, Visualization, Investigation, Writing—Original draft preparation. Saqib Khalid, Wajid Khan: Data curation, Validation, Supervision, Resources, Writing—Review & Editing. Mustafa Abdullah, Mohammad Shabaz, Mohit Bajaj: Project administration, Supervision, Resources, Writing—Review & Editing.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to M.Z.Y. or M.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.