

Article

Bi-Mamba-Based Net-Load Forecasting Method with Multidimensional Temporal Information Fusion

Guodong Guo ¹, Ke Zhang ², Zhidong Wang ¹, Fan Li ¹, Jinju Huang ³ and Xiuming Bao ^{3,*}

¹ State Grid Economic and Technological Research Institute Co., Ltd., Beijing 100031, China; guogudong@chinasperi.sgcc.com.cn (G.G.); wangzhidong@chinasperi.sgcc.com.cn (Z.W.); lifan@chinasperi.sgcc.com.cn (F.L.)

² State Grid Corporation of China, Beijing 100031, China; kezhang@sgcc.com.cn

³ School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 071003, China; jinjuhuang163@163.com

* Correspondence: bxm18270192607@163.com; Tel.: +86-18270192607

Abstract

With the rise in small-scale distributed photovoltaic (PV) power generation technology, the behind-the-meter PV problem has greatly increased the difficulty of power system regulation and management and accurate net-load forecasting is of great significance to the economic and stable operation of the power system. The timing features of the net-load sequence are complex due to a variety of factors. In order to improve the extraction effect of the timing model on the timing features of the net-load sequence and to increase the accuracy of the net-load prediction, a net-load prediction method considering multidimensional timing information is proposed. A Mamba module is introduced into the model to filter the input data, retaining some of the effective contextual information while improving the operational efficiency of the model. The structure of Bi-Mamba is used to construct a bidirectional time-series feature extraction model, which fuses the date attributes and the positive and negative time-series features of the net load to improve the stability and accuracy of the model prediction. The results of the validation algorithms show that the proposed method can reduce the normalized Mean Absolute Error (nMAE) by 17.72% and the normalized Root Mean Squared Error (nRMSE) by 21.51% compared with the temporal convolutional network (TCN) baseline. Furthermore, the model exhibits robust stability across different seasons and day types, providing a reliable reference for scheduling decisions in power systems with high PV penetration.

Keywords: net-load forecast; behind-the-meter photovoltaic; Bi-Mamba; multi-dimensional temporal information



Academic Editor: José Matas

Received: 9 January 2026

Revised: 10 March 2026

Accepted: 18 March 2026

Published: 5 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

With the worsening of global warming, China has proposed the goals of “carbon peaking and carbon neutrality” and an energy transition that is “clean and low-carbon, safe and efficient”. Centralized and distributed on-site development are now the two main modes of PV integration. According to the International Energy Agency, global distributed-PV capacity will reach 530 GW in 2024, exceeding 45% of total PV capacity, with small distributed systems dominating future growth [1]. The rise in small-scale distributed PV has spurred interest in localized generation in residential buildings, and deployments of customer-side distributed-energy systems for on-site consumption are increasing. However, owing to costs, policies and other factors, most distributed PV systems are installed behind

the meter on the user side. Such systems can cause issues for load allocation and grid interconnection [2]; moreover, the invisibility of PV output leads to marked changes in load characteristics, increasing the difficulty of system control and operation [3]. User behavior exhibits strong periodicity and seasonality, with special patterns on holidays and dynamic changes under certain meteorological conditions. These traits impose higher requirements on data preprocessing and feature extraction for modeling. Specifically, distributed PV systems primarily utilize monocrystalline or polycrystalline silicon panels, whose power output is highly sensitive to irradiance and temperature changes. Unlike centralized PV plants, behind-the-meter systems lack real-time monitoring, causing the net-load curve to exhibit a “duck curve” characteristic with deep troughs during midday. This superimposes weather-driven stochasticity onto the traditional rigidity of user loads, creating complex, non-linear coupling effects that conventional models struggle to capture.

1.1. Conventional Methods for Net-Load Forecasting

In conventional time-series forecasting, extrapolation primarily relies on the temporal characteristics of historical data. For net-load forecasting, beyond fully exploiting historical net-load data, multiple exogenous factors may be introduced as covariates. From a modeling perspective, approaches can be broadly grouped into statistical, machine learning and deep learning methods [4]. The autoregressive integrated moving average (ARIMA) model, a representative statistical method, has been widely used for time-series forecasting [5,6]. Its seasonal variant, SARIMA, targets time series with pronounced seasonality [7,8]. Machine learning methods are commonly categorized as supervised [9] and unsupervised. In supervised learning, eXtreme Gradient Boosting (XGBoost) [10] and Random Forest (RF) [11] are ensemble tree-based methods that improve accuracy by aggregating multiple decision trees. Reference [12] introduces a sliding-window mechanism into XGBoost to improve feature extraction, and [13] combines XGBoost with RF to enhance feature selection and extraction. Support Vector Regression (SVR) extends the support vector machine (SVM) to regression tasks [14,15]. Unsupervised learning analyzes and models data without explicit labels [16], typically including dimensionality reduction [17] and clustering [18]. The K-means algorithm is a canonical unsupervised clustering method [19]. In [20], users are clustered by similarity in electricity-use behavior via K-means, and correlations across clusters are exploited to improve cluster-level forecasting. For dimensionality reduction, self-organizing maps (SOMs) [21] and principal component analysis (PCA) [22] are common; in practice, they are applied to high-dimensional inputs to yield compact features for downstream prediction and reduce computational complexity.

1.2. Artificial Intelligence Methods for Net-Load Forecasting

In deep-learning-based time-series forecasting, common architectures include recurrent neural networks (RNNs) [23], long short-term memory (LSTM) networks [24,25], and temporal convolutional networks (TCNs) [26], all of which can effectively extract temporal features. RNNs leverage recurrence to use past information in current computations, establishing temporal relationships among data points [27,28], but they may suffer vanishing or exploding gradients on long sequences. LSTM addresses this by introducing memory cells and gating mechanisms, retaining long-term dependency modeling while mitigating gradient vanishing [29]. The gated recurrent unit (GRU) streamlines the LSTM architecture, reducing parameters and complexity while preserving temporal extraction capability [30]. Convolutional neural networks (CNNs), with local connectivity and weight sharing, are widely used for extracting local correlations in multidimensional data [31]. To bring CNNs’ local feature extraction to long sequences, causal and 1-D convolutions are combined to form TCNs that process sequences efficiently in paral-

lel [32]. As the strengths of these models differ, one can either scale a single architecture or flexibly compose multiple models according to task and data characteristics to harness complementary advantages [33]. For example, Reference [34] combines dual attention with TCN to extract similarity and periodicity in daily load vectors; Reference [35] fuses TCN, LSTM and attention to capture single-sequence features, long-range dependencies and multi-feature correlations; Reference [36] couples CNN's local feature strengths with LSTM's global-information extraction; and Reference [37] separates residential load into long- and short-term components modeled by CNN and LSTM respectively. Beyond deep learning hybrids, machine learning methods can be combined with deep learning depending on data characteristics [38]. Despite extensive work on load forecasting, net load under behind-the-meter PV exhibits greater volatility and randomness, complex temporal patterns and diverse influencing factors, necessitating further research tailored to net-load forecasting.

To address these issues, we propose a net-load forecasting method that incorporates multidimensional temporal information, acknowledging the complexity of net-load temporal patterns and the influence of date attributes on user behavior. A Mamba module is introduced to filter inputs, preserving effective contextual information while improving efficiency. A Bi-Mamba structure is then used to construct a bidirectional temporal-feature extractor that fuses date attributes with forward- and reverse-order net-load features, thereby enhancing prediction stability and accuracy.

The remainder of this paper is organized as follows: Section 2 details the mathematical principles of the Mamba module and the proposed Bi-Mamba forecasting framework, including date encoding and similar-day selection mechanisms. Section 3 describes the dataset, experimental setup, and comparative analysis with baseline models. Section 4 discusses the results, including seasonal performance, ablation studies, and sensitivity analysis. Finally, Section 5 concludes the paper and outlines future work.

2. Materials and Methods

2.1. Mamba Module

2.1.1. Mathematical Principles of the Mamba Module

Mamba is a model built upon structured state–space models (SSMs) [39]. It enhances SSMs so that parameters can be adjusted dynamically based on inputs, and it blends recursive and convolutional computations to handle long-range dependencies. We first introduce the mathematics of the selective SSM at Mamba's core. The classical SSM originates from the Kalman filter, whose structure is shown in Figure 1. The input signal reflects external influences on the system; the SSM maps inputs to latent states of the system and then projects them to outputs. The computation is:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1)$$

$$y(t) = Cx(t) + Du(t) \quad (2)$$

Here, A is the state matrix governing state updates; B is the control matrix injecting inputs into the state; C is the output matrix mapping states to outputs; and D is the feed-through matrix injecting inputs directly to outputs and denotes the next-step system state.

In deep learning SSMs, D is often used as a simple skip connection, allowing the original system equations to be simplified to:

$$\dot{x}(t) = Ax(t-1) + Bu(t) \quad (3)$$

$$y(t) = Cx(t) \quad (4)$$

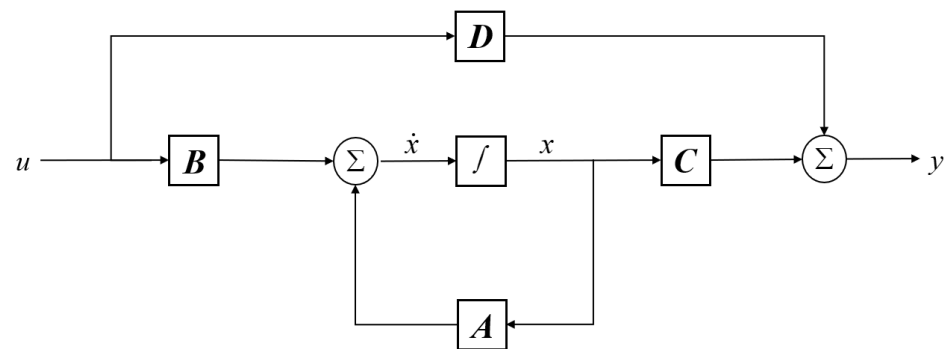


Figure 1. SSM structure.

As the original system is described in continuous time, a discretization is required for computation. Using zero-order hold, the SSM becomes:

$$x_k = \bar{A}x_{k-1} + \bar{B}u_k \quad (5)$$

$$y_k = \bar{C}x_k \quad (6)$$

where $\bar{A} = \exp(\Delta A)$, $\bar{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B$, Δ denotes the time step. The recurrence resembles an RNN and likewise cannot be parallelized. To obtain a convolutional form that alleviates this, we expand (5) as:

$$x_0 = \bar{B}u_0 \quad (7)$$

$$x_1 = \bar{A}x_0 + \bar{B}u_1 = \bar{A}\bar{B}u_0 + \bar{B}u_1 \quad (8)$$

$$\begin{aligned} x_2 &= \bar{A}x_1 + \bar{B}u_2 = \bar{A}(\bar{A}\bar{B}u_0 + \bar{B}u_1) + \bar{B}u_2 \\ &= \bar{A}^2\bar{B}u_0 + \bar{A}\bar{B}u_1 + \bar{B}u_2 \end{aligned} \quad (9)$$

The above expresses the step-by-step iteration. Noting that the last two terms both involve $\bar{A}\bar{B}$ and $\bar{B}$, we may treat the multipliers as convolution kernels and rewrite as:

$$x_k = \sum_{i=0}^k \bar{A}^{k-i} \bar{B}u_i \quad (10)$$

This x_k can be viewed as a $(u_0, u_1, u_2, \dots, u_k)$ parameterized function. Substituting (10) into (6) yields a new function:

$$y_k = \sum_{i=0}^k \bar{C}\bar{A}^{k-i} \bar{B}u_i \quad (11)$$

We thus obtain a convolutional computational structure, where the output of the convolution kernel $\bar{K} = (\bar{C}\bar{B}, \bar{C}\bar{A}\bar{B}, \dots, \bar{C}\bar{A}^k\bar{B}, \dots)$ serves as a convolutional form of $y = u * \bar{K}$. This derivation yields an SSM with a convolutional computation path that supports parallel training, while in inference it still admits a linear-complexity recurrence. However, conventional SSMs still suffer gradient vanishing in the recurrent path. Mamba therefore augments the selective-focus mechanism to improve selective copying and in-context learning. The core idea is to introduce a selection mechanism whereby the parameters governing sequence interactions are no longer fixed but change dynamically with the input. During training, Mamba treats Δ , B and C as input-dependent functions, with A fixed; i.e., selection functions $s_B(u) = \text{LinearN}(u)$ make B input-dependent, and selection functions $s_C(u) = \text{LinearN}(u)$ make C input-dependent, $s_\Delta(u) = \text{Broadcast}_D(\text{Linear}_1(u))$ producing dynamically varying Δ that is broadcast along the feature dimension D .

To further improve training efficiency, the SSM computation in Mamba is reformulated in matrix form. Let the discretized SSM have inputs x and states h , yielding:

$$h_t = Ah_{t-1} + Bx_t \quad (12)$$

$$y_t = C^\top h_t \quad (13)$$

This can be viewed as a sequence mapping akin to $Y = f_\theta(X)$, where $X, Y \in \mathbb{R}^{(T,P)}$, θ is an arbitrary parameter set, so that $P = 1$, when $P > 1$, the equations reduce to:

$$h_t = A_t h_{t-1} + B_t x_t \quad (14)$$

$$y_t = C_t^\top h_t \quad (15)$$

Define an operator $SSM(A, B, C) = SSM(A_{0:T}, B_{0:T}, C_{0:T})$ to denote this sequence operation $X \in \mathbb{R}^{(T,P)} \mapsto Y \in \mathbb{R}^{(T,P)}$, where P is a state-dimension parameter that expands input/output size by a factor of P . By induction, (14) and (15) can be generalized as:

$$h_t = \sum_{s=0}^t A_{t:s} B_s x_s \quad (16)$$

$$y_t = \sum_{s=0}^t C_t^\top A_{t:s}^\times B_s x_s = SSM(A, B, C)(x) \quad (17)$$

Therefore, an SSM transform $y = SSM(A, B, C)(x)$ with N state dimensions are equivalent to a product of sequentially semi-separable ($\mathcal{S}^2$) matrices $y = SSS(A, B, C) \cdot x$. Consequently, SSM algorithms can be implemented as structured matrix multiplications over $\mathcal{S}^2$ matrices, recasting SSM computation in a way that improves training efficiency.

2.1.2. Architecture of the Mamba Module

The Mamba architecture combines a selective SSM with a gated multilayer perceptron (MLP) to process long sequences via gating and selective scanning; the structure is shown in Figure 2. Mamba comprises an H3 backbone and a gated MLP. The H3 backbone consists mainly of an SSM and convolutional layers: the SSM tracks sequence state and performs recursive prediction in state-space form, maintaining current and past information and the convolutional layers transform inputs into state sequences suitable for the SSM, enhancing representation of dependencies. To strengthen nonlinearity, the MLP uses gating to selectively filter inputs and dynamically modulate information flow, mapping inputs into a higher-dimensional-feature space via several fully connected layers. SiLU/Swish activations yield a SwiGLU-like complex activation pattern, further improving nonlinearity. With H3 plus gated-MLP, Mamba keeps inference complexity low via recurrence while using convolutional training, and its selection mechanism improves in-context learning and alleviates gradient vanishing.

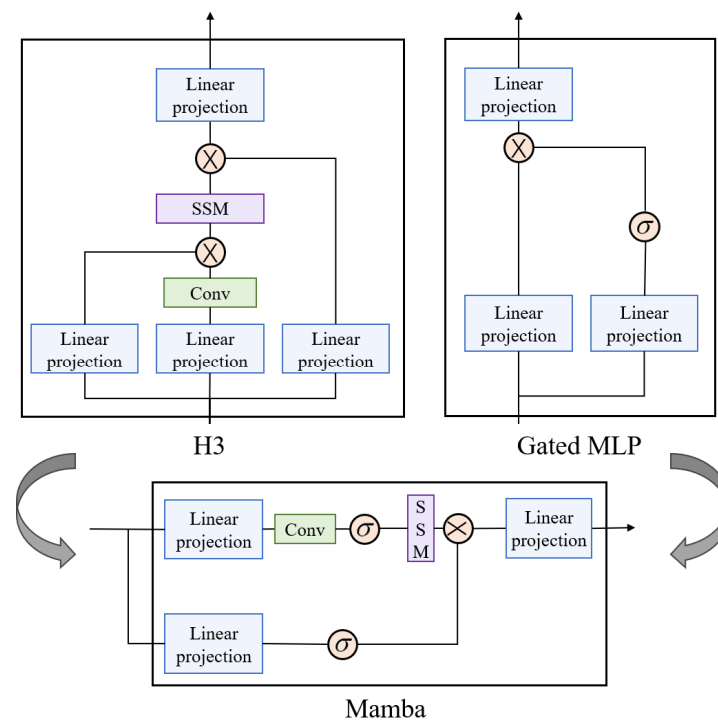


Figure 2. Mamba block.

To strengthen input selection, a multi-head SSM is adopted (Figure 3), akin to attention in that it processes multiple channels in parallel, improving computational efficiency while enhancing feature extraction and contextual retention. During state updates and output generation, the multi-head SSM can focus on different parts of the input to extract effective representations, enabling more flexible handling of diverse data structures—e.g., discrete date attributes and net-load time-series—thereby improving multi-source fusion.

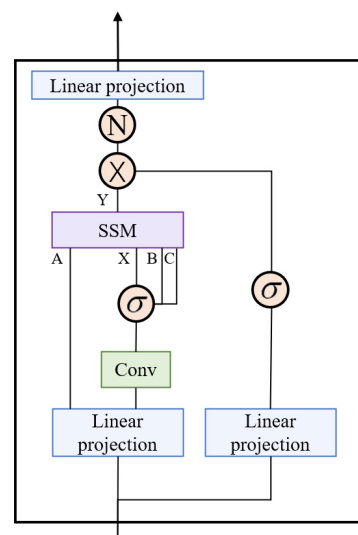


Figure 3. Mamba architecture for multi-head SSMs.

In the context of net-load forecasting, the “selective mechanism” is pivotal. The selection functions make the system matrices dynamic and input-dependent. For instance, when the net load exhibits sudden ramp-ups or drops due to cloud cover affecting PV output, the mechanism allows the model to adjust the discretization step and state transitions locally. This enables the model to selectively “reset” or “update” its hidden state to capture

high-frequency fluctuations, effectively distinguishing between relevant weather-induced anomalies and background load patterns.

2.2. Bi-Mamba-Based Net-Load Forecasting

2.2.1. Encoding of Date Attributes

For discrete variables, one-hot encoding is common, mapping categories to binary vectors to avoid spurious ordinality; it suits labels such as holiday vs. workday. However, month and day have large cardinalities: binary encodings inflate dimensionality, increase computational cost and yield sparsity, and they cannot express periodicity. We therefore adopt cyclic encodings for month and day using sine and cosine:

$$x_{label} \in \{\sin((x_{date}/m_{date}) \cdot 2\pi), \cos((x_{date}/m_{date}) \cdot 2\pi)\} \quad (18)$$

where x_{label} are sine–cosine labels; x_{date} is the specific day or month; and m_{date} is the cycle length (e.g., $m = 31$ for day-of-month and $m = 12$ for month-of-year).

Compared with one-hot encoding, cyclic encoding avoids excessive dimensionality and captures periodicity with few dimensions, aiding the extraction of periodic features. The final day-ahead encoding comprises cyclic encodings of day and month plus a one-hot holiday indicator. The date features are then broadcast to the sequence length and mapped via a linear layer into the feature space to prepare for multi-source fusion.

2.2.2. Temperature-Similar-Day Curve Matching

Temperature influences electricity-use behavior and thus the net-load peaks and troughs; similar temperature profiles may induce similar net-load shapes. To obtain plausible day-ahead temperature curves, we input a one-week look-back of historical temperature curves together with day-ahead date labels into a temperature-forecasting module (with the date labels encoded as in §2.1), producing day-ahead forecasts. Recognizing forecast errors and the strong impact of extrema on net-load peaks/valleys, we extract maximum, minimum and mean values from the forecast curve and use them to match historically similar days. We compute a distance and select the day minimizing the sum across the three attributes, retrieving its net-load curve as the similar-day reference. This supplies historical information on potential fluctuations to facilitate learning. The distance is given by:

$$D(y_{tar}, y_i) = \sum_{j=1}^m \sqrt{(y_i^j - y_{tar}^j)^2} \quad (19)$$

where y_{tar} is the target day's net-load sequence and y_i is a historical day's sequence; $m = 3$ indexes different descriptive attributes of the temperature curve—maximum, minimum and mean.

2.2.3. Bidirectional Temporal-Feature Fusion

Temporal dependencies vary across periods of the net-load curve. Around noon, a trough often occurs due to distributed PV, whereas morning and evening resemble conventional load dominated by residential behavior. We therefore adopt bidirectional feature fusion that uses both forward and reverse sequences. Because dominant drivers differ by period, transitions have distinctive temporal patterns. Analyzing both directions helps identify turning points where fluctuation regimes shift and to recognize latent trends and periodicities. Accordingly, we configure separate forward and backward encoders (Figure 4).

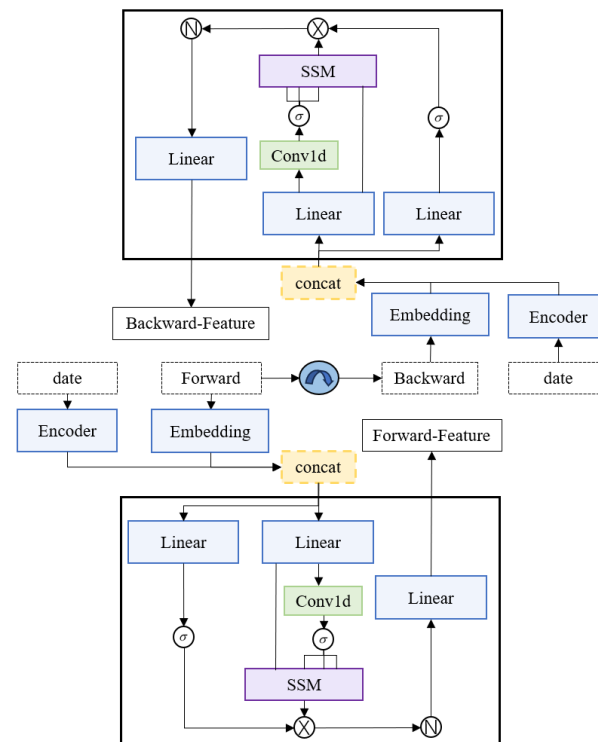


Figure 4. Bi-Mamba architecture.

The historical net-load curve, date labels and similar-day curve are fed to forward and backward encoders built with Mamba-2 modules to extract temporal features. Date labels, being discrete, are invariant to reversal, whereas sequence data yield inverse features, so the encoders learn complementary representations conditioned on the same date attributes. The forward and backward features are then summed and passed through a linear regression head to produce the day-ahead forecast.

2.2.4. Comparison with Existing Bidirectional Architectures

Unlike Bi-LSTM, which relies on recurrent computation that hinders parallelization and struggles with very long sequences due to gradient decay, Bi-Mamba leverages the Structured State Space model to achieve linear computational complexity while maintaining global context awareness. Similarly, compared to Bi-TCN, which depends on stacked dilated convolutions and has a limited receptive field determined by network depth, Bi-Mamba provides a continuous-time formulation that naturally handles irregular sampling and captures infinite-horizon dependencies. This makes the proposed architecture uniquely suited for net-load data characterized by both high-frequency PV fluctuations and long-term seasonal periodicities.

3. Case Study

3.1. Datasets and Experimental Setup

To assess accuracy and generalization, we use provincial net-load data from eastern China. The dataset contains hourly net-load measurements and dates from 1 January 2020 to 31 December 2022. For privacy, values are per-unit. Temperature data are from ECMWF at hourly frequency; as multiple sites cover the region, we average across sites to form the training set. The split is 7:3 for training/testing. Table 1 lists training hyperparameters, the optimal values for all hyperparameters were obtained through discrete Bayesian optimization. The input data concludes hourly time step, similar-day net-load curve, and varying meteorological conditions (wind speed (ws), temperature(t2m), solar radiation (ssrd), pre-

precipitation (tp), humidity (d2m), while the output data is per-unit value of historical net-load. The samples of training data are in Figure 5.

Table 1. Super parameters setting of model.

Symbol	Super Parameters	Value
e	Expansion Factor	2
h	Attention Head Dimension	64
l_r	Learning Rate	10^{-3}
b	Batch Size	50
E	Epochs	200
d_m	Latent Space Size	128
d_s	State Space Dimension Size	128

	A	E	F	G	H	I	J
1	time	net-load	tp	ssrd	ws	t2m	d2m
2	2022-01-01 00:00:00	0.44215563	0	0	1.039480209	270.977905	266.785889
3	2022-01-01 01:00:00	0.360369674	0	0	1.575655937	270.498413	266.350891
4	2022-01-01 02:00:00	0.250497926	0	0	2.058276176	270.471832	266.171631
5	2022-01-01 03:00:00	0.248372643	0	0	2.263070107	270.528931	266.177032
6	2022-01-01 04:00:00	0.273593486	0	0	2.295870781	269.715515	266.09552
7	2022-01-01 05:00:00	0.358642053	0	0	2.365077972	269.413849	266.128662
8	2022-01-01 06:00:00	0.391155613	0	0	2.000854492	269.018005	266.403015
9	2022-01-01 07:00:00	0.400789419	0	0	1.976787567	268.958801	266.417023
10	2022-01-01 08:00:00	0.388637754	0	30688	2.248543739	269.083557	266.401611
11	2022-01-01 09:00:00	0.264031076	0	404096	2.805346489	269.483826	266.422607
12	2022-01-01 10:00:00	0.177278278	0	926272	3.071597099	274.785065	268.018982
13	2022-01-01 11:00:00	0.073964085	0	1354368	3.04668808	275.002899	267.704224

Figure 5. Samples of training data.

Simulations use Python 3.9.13 and PyTorch 2.4.0 on an 11th-Gen Intel® Core™ i7-11800H @ 2.30 GHz CPU, NVIDIA GeForce RTX 3050 GPU, and 16 GB RAM.

3.2. Prediction Performance Between Models

We evaluate using mean absolute error (MAE) and root mean squared error (RMSE) between predictions and measurements. Because net-load values are per-unit, we report nMAE and nRMSE. Smaller values indicate better accuracy.

We compare against fully connected network (FCN), LSTM, TCN, and attention-based model. Inputs to all models include a 7-day look-back of net-load sequences, day-ahead date labels, and the similar-day net-load curve. Table 2 reports overall metrics for all models.

Table 2. Comparison of predictors between models.

Model	nMAE	nRMSE	R ²
FCN	0.0426	0.0491	0.7270
LSTM	0.0387	0.0441	0.7695
TCN	0.0316	0.0372	0.7944
Attention-based	0.0278	0.0325	0.8010
Bi-Mamba	0.0260	0.0292	0.8139

As shown in Table 2, among FCN/LSTM/TCN, the time-series models TCN and LSTM outperform FCN, with TCN better at fusing heterogeneous inputs. Bi-Mamba achieves lower nMAE and nRMSE than the baselines. Compared with the strongest baseline (TCN), Bi-Mamba reduces nMAE by 17.72% and nRMSE by 21.51%. Relative to Mamba, Bi-Mamba further reduces nRMSE by 3.31%, showing stronger stability. Overall, Mamba-family models are more effective for net-load forecasting with multi-source inputs.

To analyze error distributions, Figure 6 presents marginal histograms of actual and predicted net load and scatter plots of predictions vs. observations. Each subplot shows the marginal distribution of predictions (top) and observations (right). Points use predictions as abscissa and observations as ordinate; the red diagonal denotes perfect agreement. Marginal plots show TCN and LSTM predictions with two nearby modes exhibiting clearer separation, whereas the Mamba family displays less separation and better alignment with observations. Scatter plots show TCN/LSTM points more dispersed than Mamba's, whose convex hull encloses a smaller area, indicating closer agreement. Most errors arise when observations are low, but predictions are high—i.e., when distributed PV output is high and net-load magnitudes are low—implying higher meteorological sensitivity and potential bias.

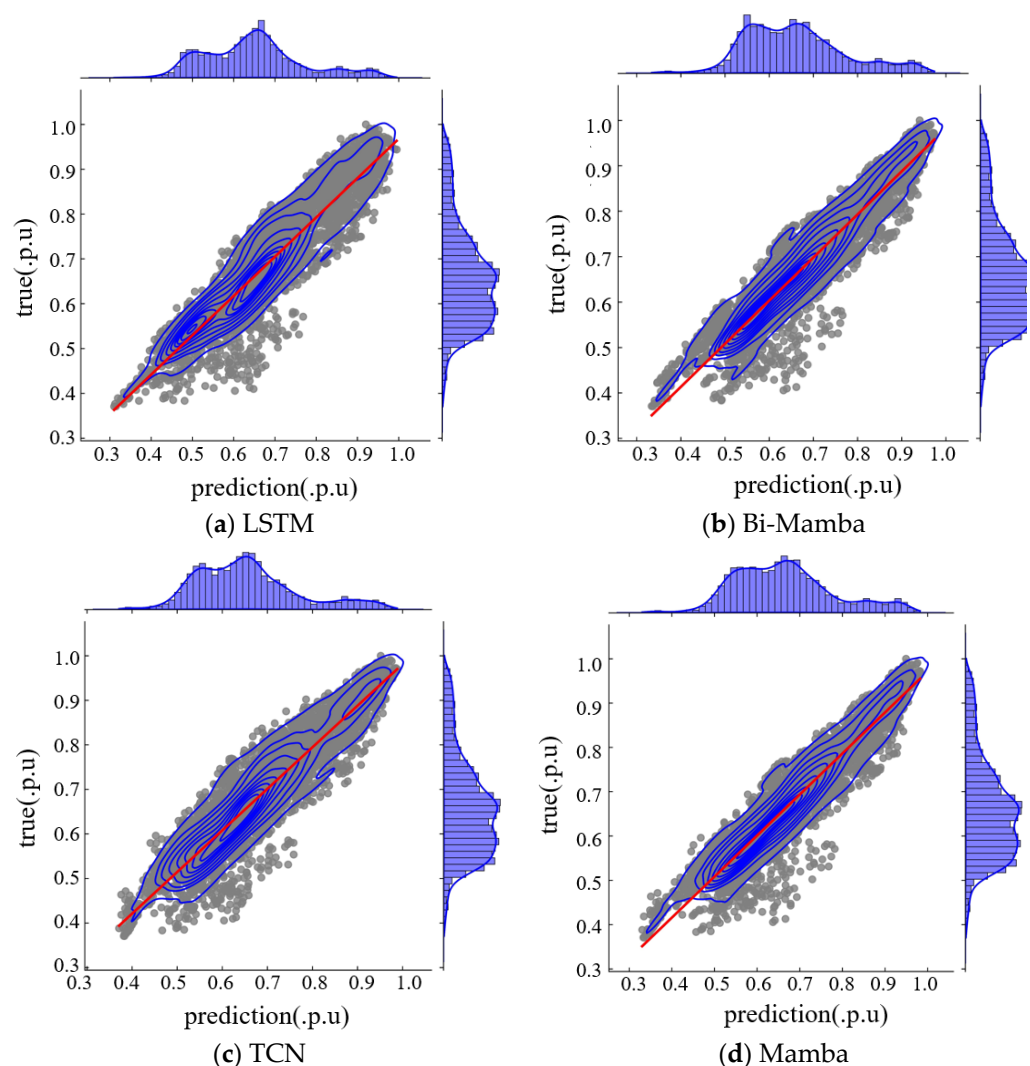


Figure 6. Comparison of net-load forecast distribution.

4. Results

4.1. Prediction Performance Across Seasons

Seasonal meteorological differences influence residential behavior, leading to distinct net-load fluctuation patterns. Figure 7 shows prediction errors by season to assess stability.

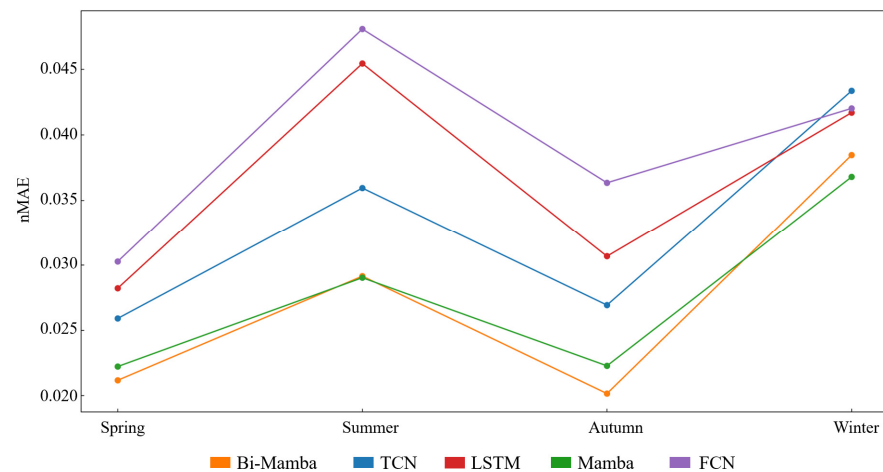


Figure 7. Comparison of prediction errors of the models in four seasons.

As shown in Figure 7, errors are smaller in spring and autumn, and larger in summer and winter. In summer and winter, there are pronounced temperature fluctuations, and users' heating and cooling loads are significantly influenced by temperature variations, which increase peak-valley contrasts and net-load fluctuations, raising errors. Cooling and heating loads in these seasons are highly responsive to temperature fluctuations. A slight deviation in temperature forecast can lead to significant load variance. In addition, in summer, while irradiance is high, frequent cloud movement and thunderstorms cause rapid PV-output ramping. In winter, snow cover or haze can unpredictably reduce generation. These high-frequency disturbances are intrinsically harder to predict than the stable transitional weather of spring and autumn. Mamba models have lower errors than baselines, especially in summer and autumn. Relative to Mamba-2, Bi-Mamba-2 yields smaller errors in all but winter, indicating better stability and that bidirectional fusion helps learn complex dynamics. Figure 8 illustrates representative prediction curves.

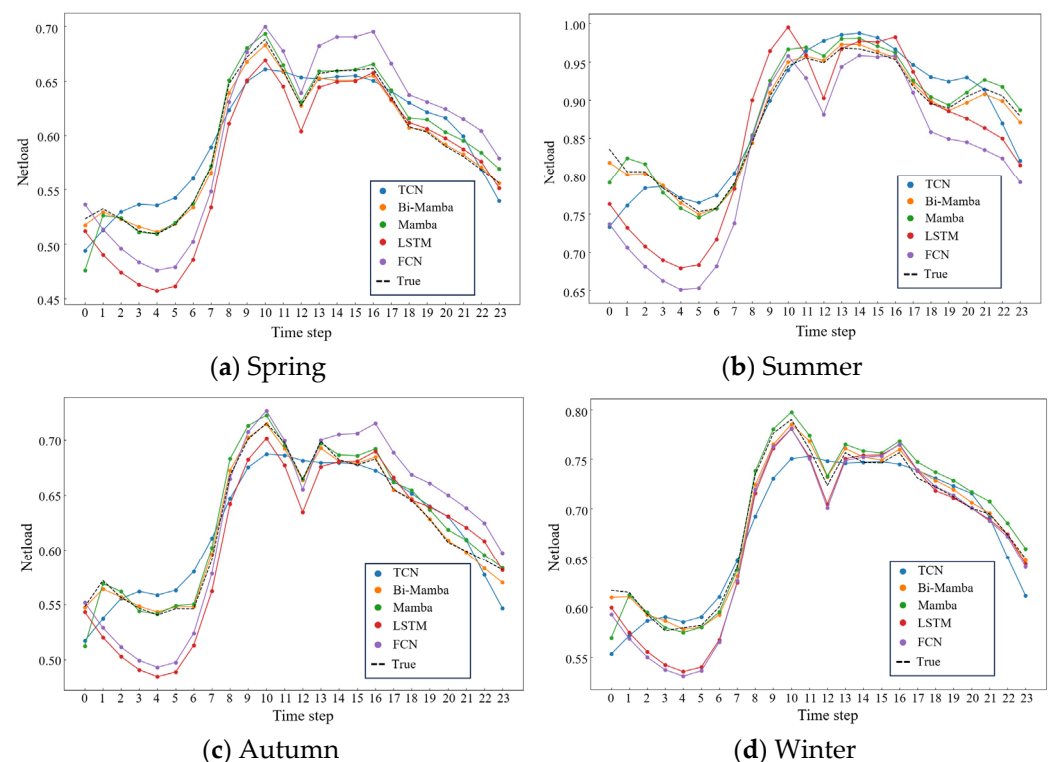


Figure 8. Comparison of net-load prediction results of each model in four seasons.

The figure indicates larger magnitudes in winter/summer than in spring/autumn, with the largest and smaller declines at midday and night in summer. Across seasons, Mamba models track trends well and better capture net-load evolution than baselines. They also incur smaller amplitude errors at key turning points (e.g., 09:00–10:00 and 14:00–15:00), suggesting effective extraction of distinctive temporal features in periods with different dominant drivers.

4.2. Prediction Performance Across Date Types

Beyond seasonal effects, date attributes also affect overall net-load trends. Figure 8 shows results for holidays vs. workdays and across weekdays/weekends. In Figure 9a, because holiday labels are included, the error difference between holidays and workdays is small; Mamba models yield lower errors than baselines, while the difference between the two date types is minor, implying similar bidirectional temporal patterns. In Figure 9b, errors are smaller on weekdays than on weekends, indicating less randomness and stronger regularity in weekday behavior. Overall, Mamba models achieve smaller errors with stable performance, evidencing effective fusion of date attributes with temporal data. Bidirectional models are more stable than unidirectional ones, showing that fusing forward/reverse features improves robustness across date-dependent patterns.



Figure 9. Comparison of the prediction errors of each model on different dates.

4.3. Ablation Studies

To verify the contribution of each component in the Bi-Mamba framework, ablation studies were conducted. We compared the full model against three variants: (1) w/o Direction: replacing Bi-Mamba with a unidirectional Mamba; (2) w/o Date: removing the cyclic date encoding; and (3) w/o SimDay: removing the similar-day net-load curve input.

The results (Table 3) indicate that the bidirectional structure contributes most significantly to error reduction, confirming the importance of capturing both past dependencies and future trends (in offline training context) or reverse-order-feature stability. Removing date encodings and similar-day curves also resulted in performance drops of 5.0% and 7.3% in nMAE respectively, validating the necessity of multidimensional fusion.

Table 3. Ablation Studies.

Model	nMAE	nRMSE	R ²
w/o Direction	0.0262	0.0302	0.8108
w/o Date	0.0273	0.0324	0.8095
w/o SimDay	0.0279	0.0335	0.8044
Bi-Mamba	0.0260	0.0292	0.8139

5. Conclusions

We propose a net-load forecasting method that accounts for multidimensional temporal information. A Mamba module filters temporal features and date attributes are fused with forward/reverse net-load features. A Bi-Mamba architecture builds a bidirectional extractor to ensure stability and accuracy. The main conclusions are:

- (1) **Superior Accuracy:** The proposed Bi-Mamba model outperforms TCN, LSTM, and Transformer baselines, reducing nMAE by 17.72% compared to TCN and achieving an R² of 0.985. The bidirectional feature extraction proves critical for recognizing turning points in the net-load curve.
- (2) **Robustness:** The model demonstrates consistent performance across diverse scenarios, yielding the lowest errors in difficult seasons (summer/winter) and varying day types, effectively mitigating the impact of holiday anomalies.
- (3) **Practical Implications:** For power system operators, the improved accuracy directly translates to reduced-reserve capacity requirements and more efficient economic dispatching. The model's linear complexity allows for rapid retraining and deployment on resource-constrained scheduling platforms.
- (4) **Limitations and Future Work:** A limitation of this study is the reliance on a single region's dataset. Extreme weather events (e.g., typhoons) may still cause large forecast deviations. Future work will focus on cross-regional validation to test generalization and the integration of numerical weather prediction (NWP) grids to further enhance resilience against extreme meteorological variability.

Author Contributions: Conceptualization, G.G., K.Z. and Z.W.; Methodology, G.G., K.Z., Z.W. and F.L.; Software, G.G.; Validation, F.L.; Data curation, F.L. and J.H.; Writing—original draft, J.H.; Writing—review & editing, X.B.; Visualization, J.H. and X.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by State Grid Corporation of China (China) grant number 1400-202456361A-3-1-DG and The APC was funded by State Grid Economic and Technological Research Institute Co., Ltd.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: Author Guodong Guo, Zhidong Wang and Fan Li were employed by the company State Grid Economic and Technological Research Institute Co., Ltd. Author Ke Zhang was employed by the company State Grid Corporation of China. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The authors declare that this study received funding from State Grid Corporation of China. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

References

1. Global Energy Review. 2025. Available online: <https://www.iea.org/reports/global-energy-review-2025> (accessed on 20 May 2025).
2. Hoke, A.; Butler, R.; Hambrick, J.; Kroposki, B. Steady-State Analysis of Maximum Photovoltaic Penetration Levels on Typical Distribution Feeders. *IEEE Trans. Sustain. Energy* **2013**, *4*, 350–357. [\[CrossRef\]](#)
3. Lopez Lorente, J.; Liu, X.; Morrow, D.J. Spatial Aggregation of Small-Scale Photovoltaic Generation Using Voronoi Decomposition. *IEEE Trans. Sustain. Energy* **2020**, *11*, 2677–2686. [\[CrossRef\]](#)
4. Zhang, D.D.; Li, F.N.; Liu, T.H. Key Technologies and Multi-Scenario Applications of Load Forecasting in New Power Systems. *Integr. Intell. Energy* **2023**, *47*, 47–61.
5. Lee, C.M.; Ko, C.N. Short-term load forecasting using lifting scheme and ARIMA models. *Expert Syst. Appl.* **2011**, *38*, 5902–5911. [\[CrossRef\]](#)
6. Fang, N.; Chen, H.; Deng, X.; Xiao, W. Short-Term Power Load Forecasting Based on VMD-ARIMA-DBN. *Proc. CSU-EPSA* **2023**, *35*, 59–65.
7. Lou, Q.; Lyu, Q.; Na, Z.; Ma, D.; Ma, X. Short-term electric power demand forecasting using a hybrid model of SARIMA and SVR. *IOP Conf. Ser. Earth Environ. Sci.* **2020**, *619*, 012035.
8. Yin, C.L.; Liu, K.; Zhang, Q.J.; Hu, K. SARIMA-Based Medium- and Long-Term Load Forecasting. *Strateg. Plan. Energy Environ.* **2023**, *42*, 283–306. [\[CrossRef\]](#)
9. Hastie, T.; Tibshirani, R.; Friedman, J. Overview of Supervised Learning. In *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Hastie, T., Tibshirani, R., Friedman, J., Eds.; Springer: New York, NY, USA, 2009; pp. 9–41. [\[CrossRef\]](#)
10. Abbasi, R.A.; Javaid, N.; Ghuman, M.N.J. Short Term Load Forecasting Using XGBoost. In *Web, Artificial Intelligence and Network Applications*; Barolli, L., Takizawa, M., Xhafa, F., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 1120–1131.
11. Lahouar, A.; Ben Hadj Slama, J. Day-ahead load forecast using random forest and expert input selection. *Energy Convers. Manag.* **2015**, *103*, 1040–1051. [\[CrossRef\]](#)
12. Cao, W.B.; Liu, Y.P.; Mei, H.W.; Shang, H.L.; Yu, Y. Short-term district power load self-prediction based on improved XGBoost model. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106826. [\[CrossRef\]](#)
13. Zhu, K.; Geng, J.; Wang, K. A hybrid prediction model based on pattern sequence-based matching method and extreme gradient boosting for holiday load forecasting. *Electr. Power Syst. Res.* **2021**, *190*, 106841. [\[CrossRef\]](#)
14. Tziolis, G.; Lopez-Lorente, J.; Baka, M.I.; Koumis, A.; Livera, A.; Theocharides, S.; Makrides, G.; Georghiou, G.E. Direct short-term net load forecasting in renewable integrated microgrids using machine learning: A comparative assessment. *Sustain. Energy Grids Netw.* **2024**, *37*, 101256. [\[CrossRef\]](#)
15. Burg, L.; Gürses-Tran, G.; Madlener, R.; Monti, A. Comparative Analysis of Load Forecasting Models for Varying Time Horizons and Load Aggregation Levels. *Energies* **2021**, *14*, 7128. [\[CrossRef\]](#)
16. James, G.; Witten, D.; Hastie, T.; Tibshirani, R.; Taylor, J. Unsupervised Learning. In *An Introduction to Statistical Learning: With Applications in Python*; James, G., Witten, D., Hastie, T., Tibshirani, R., Taylor, J., Eds.; Springer International Publishing: Cham, Switzerland, 2023; pp. 503–556. [\[CrossRef\]](#)
17. Shen, H.T.; Li, F.; Shi, L.; Sun, S.B.; Yang, Z.N.; Yang, T. Short-term Power Load Forecasting Based on Meteorological Data Dimension Reduction and Hybrid Deep Learning. *Electr. Power Constr.* **2024**, *45*, 13–21.
18. Ge, Y.M.; Qiu, C.G.; Xie, L.R.; Li, Y.F.; Li, G.; Zhao, Y.L. Multi-Energy Coupling Power Load Forecasting Based on K-means Clustering and LSTM Model. *Mod. Electr. Power* **2025**, *4*, 369–376.
19. Geng, D.; Zhang, H.; Xu, T. Effective LSTM with K-means Clustering Algorithm for Electricity Load Prediction. In Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence, Shanghai, China, 20–22 December 2019; ACM: New York, NY, USA, 2019; pp. 476–481. Available online: <https://dl.acm.org/doi/10.1145/3366194.3366279> (accessed on 25 February 2025).
20. Dong, L.; Chen, Z.P.; Han, F.J.; Wang, X.H.; Pu, T.J. Short-term Load Forecasting of Residential User Clusters Based on Graph Convolutional Neural Network and K-means Clustering. *Power Syst. Technol.* **2023**, *47*, 4291–4301.
21. López, M.; Valero, S.; Senabre, C. Application of SOM neural networks to short-term load forecasting: The Spanish electricity market case study. *Electr. Power Syst. Res.* **2012**, *91*, 18–27. [\[CrossRef\]](#)

22. Parhizkar, T.; Rafieipour, E.; Parhizkar, A. Evaluation and improvement of energy consumption prediction models using principal component analysis based feature reduction. *J. Clean. Prod.* **2021**, *279*, 123866. [[CrossRef](#)]
23. Bouktif, S.; Fiaz, A.; Ouni, A.; Serhani, M.A. Multi-Sequence LSTM-RNN Deep Learning and Metaheuristics for Electric Load Forecasting. *Energies* **2020**, *13*, 391. [[CrossRef](#)]
24. Mpawenimana, I.; Pegatoquet, A.; Roy, V.; Rodriguez, L.; Belleudy, C. A comparative study of LSTM and ARIMA for energy load prediction with enhanced data preprocessing. In Proceedings of the 2020 IEEE Sensors Applications Symposium (SAS), Kuala Lumpur, Malaysia, 9–11 March 2020; pp. 1–6. Available online: <https://ieeexplore.ieee.org/abstract/document/9220021> (accessed on 20 January 2025).
25. Kong, W.; Dong, Z.Y.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. *IEEE Trans. Smart Grid* **2019**, *10*, 841–851. [[CrossRef](#)]
26. Liu, M.; Sun, X.; Wang, Q.; Deng, S.H. Short-Term Load Forecasting Using EMD with Feature Selection and TCN-Based Deep Learning Model. *Energies* **2022**, *15*, 7170. [[CrossRef](#)]
27. Shi, H.; Xu, M.; Li, R. Deep Learning for Household Load Forecasting—A Novel Pooling Deep RNN. *IEEE Trans. Smart Grid* **2018**, *9*, 5271–5280. [[CrossRef](#)]
28. Xia, M.; Shao, H.D.; Ma, X.D.; Silva, C.W. A Stacked GRU-RNN-Based Approach for Predicting Renewable Energy and Electricity Load for Smart Grid Operation. *IEEE Trans. Ind. Inform.* **2021**, *17*, 7050–7059. [[CrossRef](#)]
29. Qiao, S.; Wang, L.; Zhang, P.C.; Yan, Q.M.; Wang, G.B. Short-term Load Forecasting Based on Modal Decomposition and Long Short-term Memory Network with Attention Mechanism. *Power Syst. Technol.* **2022**, *46*, 3940–3951.
30. Yao, C.W.; Yang, P.; Liu, Z.J. A Load Forecasting Method Based on CNN-GRU Hybrid Neural Network. *Power Syst. Technol.* **2020**, *44*, 3416–3424.
31. Zhao, A.; Mi, L.; Xue, X.; Xi, J.T.; Jiao, Y. Heating load prediction of residential district using hybrid model based on CNN. *Energy Build.* **2022**, *266*, 112122. [[CrossRef](#)]
32. Tang, X.; Chen, H.; Xiang, W.; Yang, J.M.; Zou, M. Short-Term Load Forecasting Using Channel and Temporal Attention Based Temporal Convolutional Network. *Electr. Power Syst. Res.* **2022**, *205*, 107761. [[CrossRef](#)]
33. Zhang, D.H.; Sun, K.; He, J.H. Short-term Load Forecasting Based on Similar Day and Multi-Model Fusion. *Power Syst. Technol.* **2023**, *47*, 1961–1970.
34. Zhu, C.Z.H.; Liu, D.G.; Huang, J.J.; Han, X.N.; Gao, Y.; Sun, Y.Y. Monthly Source-Load Scenario Generation Based on Progressive Growing of Generative Adversarial Networks. *High Volt. Eng.* **2024**, *50*, 3955–3964.
35. Xiao, J.W.; Liu, P.; Fang, H.; Liu, X.K.; Wang, Y.W. Short-Term Residential Load Forecasting with Baseline-Refinement Profiles and Bi-Attention Mechanism. *IEEE Trans. Smart Grid* **2024**, *15*, 1052–1062. [[CrossRef](#)]
36. Feng, Y.; Zhu, J.; Qiu, P.; Zhang, X.Q.; Shuai, K.Y. Short-term Power Load Forecasting Based on TCN-BiLSTM-Attention and Multi-feature Fusion. *Arab. J. Sci. Eng.* **2024**, *50*, 5475–5486. [[CrossRef](#)]
37. Zhu, L.J.; Xun, Z.H.; Wang, Y.X.; Cui, Q.; Chen, W.Y.; Lou, J.C. Short-term Power Load Forecasting Based on CNN-BiLSTM. *Power Syst. Technol.* **2021**, *45*, 4532–4539.
38. Jiang, L.; Wang, X.; Li, W.; Wang, L.; Yin, X.H.; Jia, L. Hybrid Multitask Multi-Information Fusion Deep Learning for Household Short-Term Load Forecasting. *IEEE Trans. Smart Grid* **2021**, *12*, 5362–5372. [[CrossRef](#)]
39. Dao, T.; Gu, A. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. *arXiv* **2024**. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.