

Mamba-Driven Transformer-LSTM Collaborative Architecture for Charging station Load Forecasting[#]

Xiao Hu¹, Zezhen Zhang¹, Hongjie Zhu², Yang Gao^{3*}, Chuang Wang¹, Shiping Jin¹, Jinduo Yang^{1,5}, Jiaquan Yang⁴

1 Northeast Electric Power University

2 Southwest Minzu University

3 Shanghai Jiao Tong University

4 Electric Power Research Institute of Yunnan Power Grid Co., Ltd

5 State Grid Xinjiang Electric Power Co., Ltd. Electric Power Research Institute

(Corresponding Author: jjgyxky@126.com)

ABSTRACT

Charging stations, as essential infrastructure for smart scheduling and energy management in smart cities, significantly improve the operational efficiency and intelligence of urban energy systems. This study proposes a load forecasting model that integrates LSTM and Transformer networks, introducing the Mamba module as a critical bridge to fuse temporal features extracted by both models, thereby constructing a superior knowledge representation to enhance prediction accuracy. First, LSTM and Transformer networks are used separately to extract latent temporal features from the load data. Subsequently, the Mamba module fuses and enhances these features to capture finer-grained information. Comprehensive experimental results demonstrate that the proposed method achieves an average prediction accuracy of 93.4% on data from multiple electric vehicle charging stations, without substantially increasing computational overhead.

Keywords: charging station load forecasting, intelligent scheduling, energy management, LSTM, Transformer, Mamba

NONMENCLATURE

Abbreviations

SSM	State-Space Model
LSTM	Long Short-Term Memory
MSE	Mean Square Error
MAE	Mean Absolute Error

Symbols

X	Load
-----	------

1. INTRODUCTION

Load prediction at charging stations is crucial for maintaining efficient operation of the charging

infrastructure and ensuring the secure and stable functioning of the electrical grid [1-3]. Accurate load forecasting not only optimizes the scheduling and allocation of charging resources but also mitigates operational risks caused by load fluctuations. Load forecasting is influenced by multiple complex factors, and conventional methods struggle to capture key features from these temporal relationships effectively. Their limited modeling capability often results in predictions that poorly approximate actual values.

To address this issue, this study proposes **MLTFuser**, a novel method that leverages the Mamba [4] module to fuse features extracted from LSTM [5] and Transformer [6] networks for enhancement and optimization, thereby achieving more accurate load forecasting. Specifically, by considering the LSTM as a "detail expert" skilled at capturing local context and neighboring information, and the Transformer as a "global observer" capable of rapidly identifying dependencies across the entire sequence, MTLFuser complementarily fuses features extracted by both models to leverage their combined strengths in modeling temporal features. Moreover, simply fusing features from the LSTM and Transformer networks fails to fully exploit their complementarity due to feature redundancy introduced during the information extraction process. To overcome this limitation, this study incorporates the Mamba module as an intermediary to refine and filter features from both sources, thereby providing purer and more distinctive representations for subsequent network learning.

Experimental results on real-world datasets demonstrate that MTLFuser effectively uncovers hidden state information and interrelationships among multiple factors in complex temporal data. In multiple charging station load forecasting tasks, the method achieves an average prediction accuracy of 93.4%, confirming its

[#] This is a paper for the 11th Applied Energy Symposium: Low Carbon Cities & Urban Energy Systems (CUE2025), July 18-22, 2024, Kitakyushu, Japan.

effectiveness and generalization capability in practical applications. The following sections sequentially present the proposed method (Section 2) and experimental results (Section 3). Finally, Section 5 concludes the paper with a summary and outlook (Section 4)

2. METHOD

Fig.1. illustrates the overall architecture of the proposed model. First, the input time-series data is processed by two preprocessing modules, which perform missing value imputation and data normalization. The preprocessed data is then fed into multi-layer LSTM and Transformer models separately for feature extraction. During this process, the features extracted at each layer are further refined using the State Space Model (SSM) module of Mamba. Finally, the optimized features from both networks are concatenated and passed through a linear mapping layer to generate the final output. The remainder of this section details the functional components of each module.

2.1 Long Short-Term Memory

LSTM mitigates the gradient vanishing problem commonly encountered by traditional RNNs in modeling

term patterns from historical records. This enhances the model's responsiveness to load variation trends and improves prediction accuracy.

Specifically, the load input data for each charging station is represented as $X \in \mathbb{R}^{T \times D}$, where T denotes the number of time steps and D represents the feature dimension. The data are first standardized by computing the mean $\mu \in \mathbb{R}^D$ and standard deviation $\sigma \in \mathbb{R}^D$ for each feature dimension, and the raw data are normalized as follows:

$$X_{\text{norm}} = \frac{X - \mu}{\sigma} \quad (1)$$

Subsequently, a sliding window mechanism is applied to construct the input sequence x_{in} and prediction target y_{out} , where:

$$x_{\text{in}} = [x_t, x_{t+1}, \dots, x_{t+L-1}], y_{\text{out}} = x_{t+L} \quad (2)$$

Here, L denotes the predefined time window length. The constructed input sequence x_{in} is fed into the LSTM layer to extract temporal feature representations within this interval. The computations within the LSTM unit are as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (3)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (4)$$

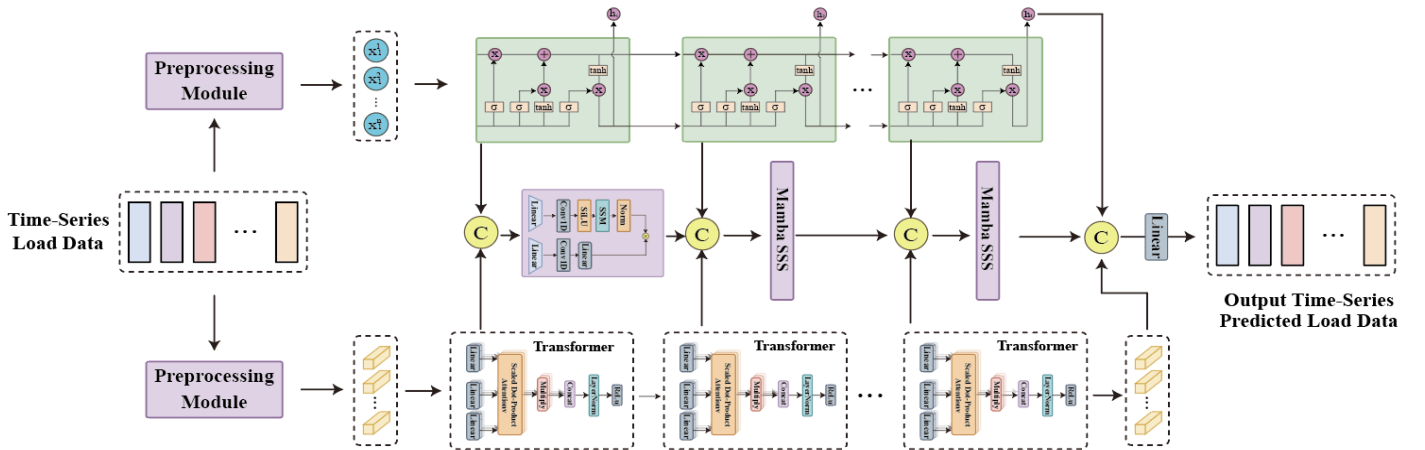


Fig. 2 Large diagram

long-term dependencies through the introduction of gating mechanisms, including the input gate, forget gate, and output gate. These mechanisms make LSTM particularly well-suited for capturing long-range temporal dependencies. Compared to standard RNNs, LSTM's memory cells maintain gradients more effectively during backpropagation, improving model trainability and enabling superior performance on long-sequence tasks. In the context of load forecasting, LSTM serves as a key component of the network architecture, modeling temporal dependencies in load data and extracting long-

$$c_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (5)$$

$$o_t = f_t \odot c_{t-1} + i_t \odot c_t, h_t = o_t \odot \tanh(c_t) \quad (6)$$

where $h_t \in \mathbb{R}^H$ denotes the hidden state at time step t , encapsulating the temporal features. In a multi-layer LSTM architecture, each layer $n \in \{1, 2, \dots, N\}$, produces a hidden state $h_t^{(n)}$. The collection of states $F_{\text{LSTM}} = \{h_t^{(1)}, h_t^{(2)}, \dots, h_t^{(N)}\}$ encodes hierarchical temporal features from the input sequence. The final hidden state,

F_{LSTM} is then passed to the Mamba module for further processing.

2.2 Transformer

Unlike traditional LSTM networks that extract features sequentially based on temporal order, the Transformer employs a self-attention mechanism to model global dependencies within the sequence and dynamically assign weights to each time step. This mechanism not only captures long-range dependencies effectively but also enhances robustness in handling complex temporal relationships. Furthermore, by relaxing the strict dependence on temporal order, the Transformer enables parallel computation, thereby significantly improving the efficiency of long-sequence modeling.

In this study, to more effectively extract complex temporal patterns from load data, a Transformer branch is incorporated into the feature extraction module. This branch employs a self-attention mechanism to capture global temporal features from the input sequence, thereby enhancing the model's ability to learn long-term dependencies and improving the accuracy of load forecasting.

Specifically, the preprocessed and normalized time-series data x_{in} is first projected into the Transformer's embedding space:

$$E_t = x_{\text{in}} W_{\text{embed}} + b_{\text{embed}}, E_t \in \mathbb{R}^{x \times d_{\text{model}}} \quad (7)$$

Here, $W_{\text{embed}} \in \mathbb{R}^{D \times d_{\text{model}}}$ denotes the learnable embedding parameter matrix, d_{model} refers to the hidden dimension of the Transformer layer in the model, and $b_{\text{embed}} \in \mathbb{R}^{d_{\text{model}}}$ is the corresponding bias vector. Since the Transformer architecture lacks inherent awareness of sequence order, positional encodings are added to differentiate between time steps. This study employs the standard sine/cosine-based positional encoding scheme:

$$PE(\text{pos}, 2i) = \sin\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right) \quad (8)$$

$$PE(\text{pos}, 2i+1) = \cos\left(\frac{\text{pos}}{10000^{2i/d_{\text{model}}}}\right) \quad (9)$$

where pos denotes the positional index of each time step in the input sequence, ranging from 0 to $T-1$, with values between 0 and 1. This index helps the model capture temporal information. Subsequently, the positional encodings are added element-wise to the input embeddings to integrate sequential position information into the model:

$$Z_0 = E_t + PE, Z_0 \in \mathbb{R}^{x \times d_{\text{model}}} \quad (10)$$

where Z_0 represents the input to the first layer of the Transformer model, consisting of the initial token embeddings summed with positional encodings.

The Transformer encoder consists of N stacked layers, each comprising two sub-layers: multi-head self-attention and a feed-forward neural network (FFN). Given the output of the $(l-1)$ -th layer, denoted as $Z_{(l-1)}$ the computation of the l -th layer is defined as follows:

$$Q = Z_{l-1} W_l^Q, K = Z_{l-1} W_l^K, V = Z_{l-1} W_l^V \quad (11)$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (12)$$

Here, Q , K , and V denote the Query, Key, and Value, respectively. And $W_l^Q, W_l^K, W_l^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ denotes the weight matrix for the l -th layer, and $d_k = d_v = d_{\text{model}} / h$, and h represents the number of attention heads in that layer. Next, the multi-head attention of the input is computed using the aforementioned Q , K , and V :

$$\text{MultiHead}(Z_{t-1}) = \text{Concat}(H_1, H_2, \dots, H_h) W_l^O \quad (13)$$

$$H_i = \text{Attention}(Z_{l-1} W_{l,i}^Q, Z_{l-1} W_{l,i}^K, Z_{l-1} W_{l,i}^V) \quad (14)$$

where $W_l^O \in \mathbb{R}^{h \cdot d_k \times d_{\text{model}}}$ is the output projection matrix. Finally, the extracted features from this layer are obtained using residual connections and layer normalization, which are then passed to subsequent modules:

$$Z_l[i] = \text{LayerNorm}(Z_l^i + \text{MultiHead}(Z_{l-1})) \quad (15)$$

The global features $F_{\text{Trans}} = \{Z_t^{(1)}, Z_t^{(2)}, \dots, Z_t^{(N)}\}$ extracted by the Transformer and the local temporal features obtained from the LSTM are jointly input into the State Space Model (SSM) layer of the Mamba module. In this layer, Mamba performs unified modeling and fine-grained refinement of the combined features, thereby enabling complementary enhancement.

2.3 Mamba

In time series analysis, different features often capture distinct characteristics of the data. For example, some features may be more sensitive to short-term fluctuations, while others are better suited for modeling long-term trends. To fully leverage this diverse information, we incorporate the State Space Model (SSM) module within the Mamba framework. This module not only processes each type of feature independently but also integrates them effectively, thereby constructing a more comprehensive and accurate time series representation. This integrated approach improves not only prediction accuracy but also the model's stability and adaptability.

First, two complementary features, F_1 and F_2 , are extracted from the original time series data. These features are then input into the State Space Model (SSM) module within the Mamba architecture. Within the SSM module, state transition and observation equations are defined for each feature, enabling the model to learn both their temporal dynamics and mutual interactions. Upon training completion, the SSM module intelligently fuses information from both features to generate optimized feature representations or directly produce predictions. Specifically, The features extracted from each LSTM layer $F_{\text{LSTM}} = \{h_t^{(1)}, h_t^{(2)}, \dots, h_t^{(N)}\}$ and those from the Transformer $F_{\text{Trans}} = \{Z_t^{(1)}, Z_t^{(2)}, \dots, Z_t^{(N)}\}$ are fed into the State Space Model (SSM) layer of the Mamba module, where the fusion and refinement of features from different models are performed. The specific formula is given as follows:

$$F_{\text{input}} = \text{Concat}(F_{\text{LSTM}}, F_{\text{Trans}}) \quad (16)$$

$$H_2 = \text{Conv1D}(\text{Linear}(F_{\text{input}})) \quad (17)$$

$$F_{\text{fused}} = \text{LayerNorm}(\text{SSM}(\text{SiLU}(H_2))) + \text{Linear}(F_{\text{input}}) \quad (18)$$

Finally, the fused features serve as residual connections for the two branch networks and are added respectively to the original features to produce optimized feature representations, thereby enhancing the network's expressive capacity and prediction performance.

3. EXPERIMENTS

3.1 Dataset

The dataset used in this study comprises operational load records from 45 battery swap stations located in a city in Jilin Province, spanning the period from March 2022 to February 2023. It contains a total of 394,200 hourly time-series entries, with each station contributing 8,760 observations. Each data vector includes multidimensional features such as current load values, meteorological conditions, and temporal information (e.g., day of the week, hour of the day). All data were collected from real-world operations, ensuring high practical relevance. To address missing values in some records, spline interpolation was applied for imputation to maintain data integrity and support stable model training.

3.2 Implementation Details

Five-layer LSTM and Transformer networks serve as backbone feature extractors, with hyperparameters configured according to their original sources. Experiments utilize a batch size of 64 and sequential input lengths of 36, 72, and 96 historical time steps to predict load values over the subsequent 24, 36, 64, and

96 time steps. The models are trained using the AdamW optimizer with a linear warm-up phase of 1,500 iterations ($T_{\text{warm}} = 1.5K$) followed by a linear decay schedule. All experiments are performed on a single NVIDIA GeForce RTX A100 GPU equipped with 40 GB of memory.

3.3 Main Results

Tab.1. presents a comparison of the proposed method with several advanced time-series models in terms of MSE and MAE. Compared to models that utilize a single feature extraction mechanism, the proposed multi-temporal feature fusion strategy enables more comprehensive exploitation of latent data representations. It is particularly effective in producing predictions that closely match the ground truth during sharp fluctuations in load patterns, demonstrating enhanced generalization and temporal modeling capabilities. Notably, across prediction tasks with varying historical time step lengths, the proposed method consistently achieves superior performance in most cases, further validating its effectiveness and robustness in multi-scale load modeling scenarios.

Tab 1. Comparison with predictions that are out of sync with other models.

Model	target time span/h	evaluation metrics	
		MAE	MSE
MLTFuser	24	0.071	0.032
	36	0.032	0.021
	64	0.119	0.118
	96	0.332	0.362
Mamba	24	0.428	0.323
	36	0.373	0.247
	64	0.512	0.462
	96	0.517	0.558
LSTM	24	0.522	0.812
	36	0.212	0.515
	64	0.586	0.381
	96	1.443	0.461
Transformer	24	0.261	0.390
	36	0.281	0.343
	64	0.482	0.561
	96	1.251	0.761

This study further evaluates prediction accuracy across varying historical time step lengths for different forecasting intervals. As illustrated in Fig. 2, samples with an absolute prediction error below 0.05 are classified as positive, while those exceeding this threshold are considered negative. Experimental results indicate that the proposed method consistently outperforms

mainstream models under diverse historical input configurations, achieving higher load forecasting accuracy. It demonstrates improved capability in capturing temporal dependencies and maintaining low prediction errors. These enhancements are primarily attributed to the Mamba module's pivotal role in feature fusion, which effectively integrates complementary features extracted by the LSTM and Transformer networks.

Tab.2. compares the computational speeds of the proposed method with those of other models. Experimental results indicate that, despite utilizing fewer Transformer and LSTM layers, the proposed method achieves prediction accuracy comparable to multi-layer Transformer models while substantially reducing computational time.

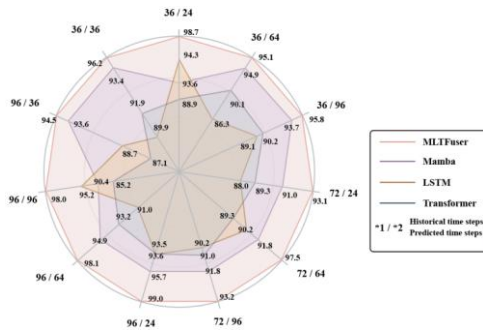


Fig 2. Comparison of MLTFuser's accuracy with other models.

This improvement is mainly due to Mamba's efficiency in the feature fusion process: by effectively integrating complementary temporal information extracted from both LSTM and Transformer, Mamba facilitates efficient and targeted optimization of knowledge representation, thereby enhancing runtime performance without compromising model expressiveness.

Tab 2. Comparison of model complexity.

Model	Transformer/LSTM Layers	Prediction Accuracy (%)	Computational Time (s)
MLTFuser	5	93.6	1.8
Mamba	5	93.1	1.9
Transformer	5	91.8	3.7
LSTM	5	92.5	2.4

4. CONCLUSIONS

This study proposes a load forecasting framework that integrates LSTM and Transformer architectures using Mamba as a bridging module. The framework leverages Mamba to fuse features extracted from both temporal models, enabling more effective knowledge representation and improving the accuracy of charging

station load predictions. Experiments conducted on a real-world dataset demonstrate that the proposed method effectively captures latent temporal patterns influenced by multiple complex factors, achieving a prediction accuracy of 93.7%. These findings validate the potential of combining LSTM and Transformer for enhanced temporal modeling and underscore the critical role of Mamba in enriching complementary features and improving forecasting performance.

ACKNOWLEDGEMENT

This work was supported in part by the National Key R&D Program of China (2023YFB2407300). National Natural Science Foundation of China (52407127), Shanghai Pujiang Program (24PJA045)

REFERENCE

[1] Zhu J, Yang Z, Mourshed M, Guo Y, Zhou Y, Chang Y, Wei Y, Feng S. Electric vehicle charging load forecasting: A comparative study of deep learning approaches. *Energies*. 2019 Jul 13;12(14):2692.

[2] Zhao P, Hu W, Cao D, Zhang Z, Liao W, Chen Z, Huang Q. Enhancing multivariate, multi-step residential load forecasting with spatiotemporal graph attention-enabled transformer. *International Journal of Electrical Power & Energy Systems*. 2024 Sep 1;160:110074.

[3] Alansari M, Al-Sumaiti AS, Abughali A. Optimal placement of electric vehicle charging infrastructures utilizing deep learning. *IET Intelligent Transport Systems*. 2024 Aug;18(8):1529-44.

[4] Gu, A., & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

[5] Graves A, Graves A. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*. 2012:37-45.

[6] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. *Advances in neural information processing systems*. 2017;30.