A power load forecasting model based on a combined neural network

Jie Li; Chenguang Qiu; Yulin Zhao; Yuyang Wang

AIP Advances 14, 045231 (2024)
https://doi.org/10.1063/5.0185448
A power load forecasting model based on a combined neural network

Cite as: AIP Advances 14, 045231 (2024); doi: 10.1063/5.0185448
Submitted: 31 October 2023 • Accepted: 30 March 2024 •
Published Online: 17 April 2024

Jie Li, Chenguang Qiu, Yulin Zhao, and Yuyang Wang

AFFILIATIONS
Power Dispatching Center, State Grid Jiangsu Electric Power Company, Nanjing 210000, China

*Author to whom correspondence should be addressed: zhaoyund@yeah.net

ABSTRACT
The supply of electric power is vital for the daily lives of people, industrial production, and business services. At present, although enough electric power can be supplied to meet the power demand, there are still some challenges, especially in terms of long-distance power transmissions and long-term power storage. Consequently, if the power production capacity exceeds the immediate consumption requirements, i.e., the produced electric power cannot be consumed in a short period, and much electric power could be wasted. Evidently, to minimize the wastage of electric power, it is necessary to properly plan power production by accurately forecasting the future power load. Therefore, a preferable power load forecasting algorithm is crucial for the planning of power production. This paper proposes a novel deep learning model for the purpose of power load forecasting, termed the SSA-CNN-LSTM-ATT model, which combines the CNN-LSTM model with SSA optimization and attention mechanisms. In this model, the CNN module extracts the features from the sequential data, and then the features are passed to the LSTM module for modeling and capturing the long-term dependencies hidden in the sequences. Subsequently, an attention layer is employed to measure the importance of different features. Finally, the output is obtained through a fully connected layer, yielding the forecasting results of the power load. Extensive experiments have been conducted on a real-world dataset, and the metric $R^2$ can reach 0.998, indicating that our proposed model can accurately forecast the power load.

© 2024 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/). https://doi.org/10.1063/5.0185448

I. INTRODUCTION
Power load forecasting is vital to the stability of the power system, which helps to improve the efficiency of power production and utilization. Specifically, a preferable power load forecasting method can help reduce electricity waste and power costs and ensure power grid security. However, there are various factors affecting the power load, such as time, temperature, and holidays; thus, it is quite difficult to forecast the power load accurately. Therefore, the accuracy enhancement of power load forecasting has become an important issue in recent years. Essentially, power load forecasting is a problem of time-sequence forecasting, i.e., the task of power load forecasting is to forecast the electric power levels in the future based on the previous ones. For the task of time-sequence forecasting, the current methods include traditional statistical methods, machine learning methods, and deep learning methods.

In the early years, power load forecasting techniques mainly rely on traditional statistical methods, which can be realized without complex data analysis and high computational complexity. These methods analyze the historical records of power load and decompose the seasonal patterns to build some time sequence models for forecasting the future power load, such as the Linear Regression (LR) method and the Autoregressive Integrated Moving Average Model (ARIMA) method. The forecasting accuracy of these methods is high when the historical records of power loads are linear and regular. However, the variation of power load is typically not stable and could be influenced by many factors, and the features hidden in the historical records of power load cannot be captured by traditional statistical methods. In addition, these methods perform badly in long-term power load forecasting.

With the advent of machine learning concepts, some machine learning methods have been applied to power load forecasting. Machine learning methods are suitable for solving forecasting problems, and they can capture the nonlinear relationships and patterns hidden in complex data. The typical works include Random Forests (RF), Support Vector Regression (SVR), and XGBoost (XGB). Machine learning methods can also take multiple factors into account that affect the power load, rather than only considering

© Author(s) 2024
the single feature of historical electricity consumption. In addition, machine learning methods are able to process large-scale data, so they have an outstanding advantage in the long-term forecasting of power loads.

In recent years, deep learning methods have made significant advancements in power load forecasting. The first deep learning method used for power load forecasting is the backpropagation neural network (BPNN), which is a feed-forward neural network. However, BPNN does not perform well in learning the time sequence data. Another deep learning model, the Recurrent Neural Network (RNN), focuses on learning the time sequence data. RNN can effectively deal with the time, temperature, holidays, and other characteristics of the power load data. Nevertheless, when RNN deals with long sequences, the gradient disappearance or gradient explosion could be caused, and it could struggle to capture the long-term dependencies. To solve the above-mentioned problems, Hochreiter and Schmidhuber propose a Long Short Term Memory (LSTM) recurrent neural network. Furthermore, some works combine Convolutional Neural Networks (CNNs) and LSTM into a hybrid model, which makes use of the advantages of CNN and LSTM jointly to carry out the task of power load forecasting and can achieve good forecasting accuracy. However, the combined hybrid model treats the input features equally, while the effects of the influencing factors on the power load are quite different. Moreover, the number of units in the hidden layer of the LSTM model has a great impact on the forecasting results, and the parameter tuning is important as well.

With the appearance of transformer, there have been new innovations in some power load forecasting models. Ran et al. propose a power load forecasting method based on a transformer framework that achieves good results in terms of forecasting accuracy while consuming a high amount of time. Xu et al. take the Informer structure into power load forecasting; although the forecasting accuracy can be improved, the time cost still remains very high. Rajbhandari et al. conduct a study of electricity demand and its relation to the previous day’s lags and temperature by examining the case of a consumer distribution center in urban Nepal. The effect of temperature on load is specially investigated.

For the forecasting of power loads, the critical factors that affect the power load mainly include time, temperature, humidity, light intensity, wind power, rainfall, etc. Among these factors, the power load typically shows an obvious regularity over time due to the fact that each year has the same periodic seasons, holidays, and other temporal factors that significantly affect the power load. In addition, note that the temperature is also strongly related to the power load required by industrial production and residential lives. To this end, we select time and temperature as the main factors related to the power load. Note that the time cost also needs to be controlled. Therefore, we choose CNN-LSTM as the basic framework and integrate an attention layer after the CNN-LSTM layer to handle the multi-feature inputs. In addition, we use the SSA optimization algorithm [Sparrow Search Algorithm (SSA)] to optimize the number of units in the LSTM.

The challenge of this research is mainly that the accuracy requirements for power load forecasting are very high to meet the planning requirements of electricity production for power companies (such as our company, State Grid Jiangsu Electric Power Company). Specifically, the power load of tomorrow must be accurately predicted to determine the electricity production of tomorrow. If the electricity production exceeds the actual power load, then some electricity resources could be wasted; otherwise, if the electricity production is much smaller than the actual power load, then much electricity resources should be purchased from other provinces, reducing the production profit (the cost of purchased electricity is evidently larger than that of produced electricity).

This paper is organized as follows: In Sec. II, we introduce our proposed model. Then, in Sec. III, the SSA-CNN-LSTM-ATT model is compared with some other models in terms of forecasting the accuracy of power load. Finally, the conclusion is drawn in Sec. IV.

II. SSA-CNN-LSTM-ATT MODEL

Figure 1 shows the flow chart for power load forecasting. First, the dataset is divided into two parts: training data and testing data. Then, our proposed SSA-CNN-LSTM-ATT is trained, and the model parameters are optimized for power load forecasting. Figure 2 illustrates the structure of SSA-CNN-LSTM-ATT. The main network structure of SSA-CNN-LSTM-ATT is the CNN-LSTM network, which adopts the SSA algorithm for the parameter optimization of LSTM. Moreover, an attention layer is added after the LSTM layer to further improve forecasting accuracy.

A. CNN-LSTM

The CNN-LSTM network is comprised of a convolutional neural network module and a long short-term memory module. The functions of the two modules are described as follows:

![Flow chart of power load forecasting](image-url)

**FIG. 1.** Flow chart of power load forecasting.
CNN is taken as a feature extractor. By employing some convolutional layers, CNN effectively learns local features from the time sequence data. After the CNN layer, the sample is flattened into a one-dimensional vector as the input of the LSTM layer.

The LSTM module is used for sequence modeling. The extracted features are input into the LSTM module to capture temporal dependencies. By utilizing the memory units and gating mechanisms, the LSTM module effectively handles the long-term dependencies and learns the temporal patterns hidden in the sequence data. The LSTM module continuously adapts to the internal memory states and generates the outputs based on the previous states and inputs.

As illustrated in Fig. 2, the combination of the feature extraction of CNN for sequence data and the temporal dependency capturing of LSTM can yield a more comprehensive and accurate modeling capability. This combination significantly enhances the performance and effectiveness of our proposed SSA-CNN-LSTM-ATT model.

B. Attention mechanism

The time-series data outputted by CNN-LSTM have a temporal dependence, and the importance of temporal features varies over time. Some subsequences may have higher feature importance than others. If they are left untreated, their importance cannot be
reflected. In our work, an attention layer is added after the LSTM layer to address this issue, and the attention mechanisms proposed by Bahdanau et al.\(^2^4\) and Luong et al.\(^2^5\) are applied, respectively. Finally, the one with the best performance is selected.

The Bahdanau attention mechanism and the Luong attention mechanism have the same basic principles. Figure 3 shows the execution process of the Luong attention mechanism. First, the encoder generates the hidden states for each element in the input sequences. The previous hidden states and the output of the decoder are used by the decoder LSTM to generate a new hidden state. According to the new hidden state of the decoder and the hidden state of the encoder, the alignment scores are calculated. The alignment scores of each hidden state of the encoder are combined and represented into a single vector, which is then softmaxed. After that, the hidden states of the encoder are multiplied by their respective alignment scores to form a context vector. Finally, the context vector is concatenated with the hidden state of the decoder, and a new output is generated through a fully connected layer. The difference between the Bahdanau attention mechanism and the Luong attention mechanism lies in the calculation of alignment scores.

**TABLE I.** Explanations of symbols.

<table>
<thead>
<tr>
<th>Sign</th>
<th>Meaning</th>
<th>Sign</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X_i^t)</td>
<td>The (i)th dimensional position information of the (k)th sparrow in the (t)th iteration</td>
<td>(\alpha)</td>
<td>Random number in the interval [0, 1]</td>
</tr>
<tr>
<td>(SV)</td>
<td>Safety value ranging from 0.5 to 1</td>
<td>(L)</td>
<td>Matrix with elements equal to 1</td>
</tr>
<tr>
<td>(X^t_{P})</td>
<td>Current location of the best discoverer</td>
<td>(\beta)</td>
<td>A random number that follows a normal distribution with a mean of 0 and a variance of 1</td>
</tr>
<tr>
<td>(A^+)</td>
<td>(A^+ = A^T (AA^T)^{-1})</td>
<td>(f_k)</td>
<td>Fitness value of the current individual</td>
</tr>
<tr>
<td>(f^w)</td>
<td>Current maximum fitness value</td>
<td>(f^w)</td>
<td>Current minimum fitness value</td>
</tr>
</tbody>
</table>

\(^2^4\) Bahdanau et al., 2014
\(^2^5\) Luong et al., 2015
C. Parameter optimization

The unit parameter setting of LSTM is an important issue for model training. In SSA-CNN-LSTM-ATT, the number of LSTM units is optimized by the SSA algorithm. The SSA algorithm updates the positions and velocities of sparrows in a random manner and selects the best solutions (with the minimum loss) based on the fitness evaluation of each sparrow. The schematic diagram of the SSA optimization algorithm is shown in Fig. 4.

The explanations of main symbols are provided in Table I.

The optimization process for the number of LSTM units by SSA is described as follows:

(i) Initialization of the population. A set of initial sparrow individuals is randomly generated, where each individual represents a configuration of parameters to be optimized.

(ii) Fitness evaluation. The fitness of each individual is calculated based on some metrics, such as Mean Squared Error (MSE).

(iii) Position updates. The positions of sparrows are updated based on their fitness and positions. This updating process simulates the movements and adjustments of sparrows during the search. The initial number of sparrows is set to \( N \).

The formula for updating the discoverer’s location is expressed as

\[
X_{L_j}^{t+1} = \begin{cases} 
X_{L_j}^t \cdot \exp \left( -\frac{k}{a \cdot \text{iter}_{\text{max}}} \right) & \text{if } R_2 < SV, \\
X_{L_j}^t + Q \cdot L & \text{otherwise.} 
\end{cases}
\] (1)

The formula for updating the tracker’s location is expressed as

\[
X_{L_j}^{t+1} = \begin{cases} 
X_{L_j}^t + \beta \cdot \frac{X_{L_j}^t - X_{L_j}^{t+1}}{\left| X_{L_j}^t - X_{L_j}^{t+1} \right|} & \text{if } f_k \neq f_g, \\
X_{L_j}^t + K \cdot \frac{f_k - f_w + \varepsilon}{\left| X_{L_j}^t - X_{L_j}^{t+1} \right|} & \text{otherwise.} 
\end{cases}
\] (2)

(iv) Termination condition. The termination of the optimization process is determined based on a stopping criterion. If the termination condition is not satisfied, return to (ii).

(v) Output of the optimal solution. The optimal number of LSTM units is outputted.

D. Loss function

SSA-CNN-LSTM-ATT adopts Mean Squared Error (MSE) as the loss function, which is a commonly used loss function typically used for regression problems. MSE measures the average squared difference between the forecasting values and the real values. A smaller MSE indicates a smaller gap between the forecasting values and the real values. MSE is written as

\[
\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,
\] (4)

The update formula for hazard warning is expressed as

\[
X_{L_j}^{t+1} = \begin{cases} 
X_{L_j}^t + \beta \cdot \frac{X_{L_j}^t - X_{L_j}^{t+1}}{\left| X_{L_j}^t - X_{L_j}^{t+1} \right|} & \text{if } f_k \neq f_g, \\
X_{L_j}^t + K \cdot \frac{f_k - f_w + \varepsilon}{\left| X_{L_j}^t - X_{L_j}^{t+1} \right|} & \text{otherwise.} 
\end{cases}
\] (3)

The format of the dataset is shown in Table II.

### Table II. Format of the dataset.

<table>
<thead>
<tr>
<th>Datetime</th>
<th>point_time</th>
<th>high_tmp</th>
<th>low_tmp</th>
<th>xq</th>
<th>Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022/1/1</td>
<td>0:05</td>
<td>8</td>
<td>4</td>
<td>6</td>
<td>77 656.540</td>
</tr>
<tr>
<td>2022/1/1</td>
<td>0:10</td>
<td>8</td>
<td>4</td>
<td>6</td>
<td>77 379.620</td>
</tr>
<tr>
<td>2022/1/1</td>
<td>0:15</td>
<td>8</td>
<td>4</td>
<td>6</td>
<td>77 267.120</td>
</tr>
<tr>
<td>2022/1/1</td>
<td>0:20</td>
<td>8</td>
<td>4</td>
<td>6</td>
<td>76 962.820</td>
</tr>
<tr>
<td>2022/12/31</td>
<td>23:50</td>
<td>10</td>
<td>-2</td>
<td>6</td>
<td>71 471.340</td>
</tr>
<tr>
<td>2022/12/31</td>
<td>23:55</td>
<td>10</td>
<td>-2</td>
<td>6</td>
<td>71 223.640</td>
</tr>
</tbody>
</table>

![Histogram of power load data](image-url)
where $n$ denotes the number of samples, $y_i$ denotes the real value, and $\hat{y}_i$ denotes the forecasting value. MSE calculates the loss by computing the squared differences between each forecasting value and the corresponding real value.

### E. Dataset

The dataset used for experiments is a real-world dataset (the power load dataset of Jiangsu Province in 2022). The sampling points are collected every 5 min for a total of 105 120 sample points. The dataset is in CSV format. The format of the dataset is shown in Table II, where “high_tmp” and “low_tmp” represent the highest temperature and lowest temperature (in degree Celsius) of the day, respectively. “xq” denotes the day of the week. The last column of “load” denotes the power load. The first five columns are input features, and the last column is the value to be forecasted. After EDA analysis, it can be found that the dataset has no missing or abnormal values, and the data are true and credible (because the data are obtained from the real power grid).

As shown in Fig. 5, the horizontal axis represents the value of the power load, and the vertical axis represents the frequency. When the value of the power load falls into the interval [67 600, 84 100], the highest frequency and proportion are yielded.

### F. Parameter configuration

Our proposed SSA-CNN-LSTM-ATT model is realized based on the tensor flow framework of Python. The model is trained with the following hyperparameters:

1. Type of convolution: One dimensional.
2. No. of filter: 64 with kernel size 3.
3. Activation: ReLU for CNN, LSTM, and dense layer; Softmax for attention.
4. Optimizer: Adam.
5. Number of units: best_units optimized by SSA.

### TABLE IV. Comparisons between two attention mechanisms.

<table>
<thead>
<tr>
<th>Attention mechanism</th>
<th>$R^2$↑</th>
<th>MAE↓</th>
<th>RMSE↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0.991</td>
<td>601</td>
<td>789</td>
</tr>
<tr>
<td>Bahdanau</td>
<td>0.997</td>
<td>349</td>
<td>446</td>
</tr>
<tr>
<td>Luong</td>
<td>0.998</td>
<td>266</td>
<td>355</td>
</tr>
</tbody>
</table>

As depicted in Table III, the deep learning model innovatively combines four modules: CNN, LSTM, attention, and SSA optimization algorithms to achieve high accuracy in power load forecasting. Our proposed model first extracts the features of the input time series through a CNN layer, and then the LSTM layer captures the long-term dependencies for the sequence modeling. The attention layer processes the importance of different time steps, allowing this model to adaptively focus on the most important historical information at the current prediction time step rather than simply considering all historical data equally. In addition, the parameters of the LSTM units are optimized by the SSA optimization algorithm to achieve the optimal configuration. Therefore, high accuracy in power load forecasting can be achieved by combining the above four modules.

### C. Evaluation metrics

We employ the following metrics for measuring forecasting accuracy:

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

where $n$ is the number of samples, $y_i$ is the real value, and $\hat{y}_i$ is the forecasting value.

### TABLE V. Comparisons among related models. Boldface denotes the best results in Tables V and Tables VI.

<table>
<thead>
<tr>
<th>Model</th>
<th>$R^2$↑</th>
<th>MAE↓</th>
<th>RMSE↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>0.984</td>
<td>839</td>
<td>1079</td>
</tr>
<tr>
<td>CNN-LSTM</td>
<td>0.991</td>
<td>601</td>
<td>789</td>
</tr>
<tr>
<td>CNN-LSTM-ATT</td>
<td>0.993</td>
<td>591</td>
<td>670</td>
</tr>
<tr>
<td>SSA-CNN-LSTM-ATT</td>
<td>0.998</td>
<td>266</td>
<td>355</td>
</tr>
</tbody>
</table>

### TABLE VI. Comparisons among related methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>$R^2$↑</th>
<th>MAE↓</th>
<th>RMSE↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVR</td>
<td>0.733</td>
<td>3822</td>
<td>4359</td>
</tr>
<tr>
<td>XGB</td>
<td>0.994</td>
<td>514</td>
<td>641</td>
</tr>
<tr>
<td>RF</td>
<td>0.996</td>
<td>419</td>
<td>532</td>
</tr>
<tr>
<td>SSA-CNN-LSTM-ATT</td>
<td>0.998</td>
<td>266</td>
<td>355</td>
</tr>
</tbody>
</table>

As depicted in Table III, the deep learning model innovatively combines four modules: CNN, LSTM, attention, and SSA optimization algorithms to achieve high accuracy in power load forecasting.
$R^2$ (R-squared) is a statistical metric used to evaluate the goodness-of-fit of a regression model, and it is expressed as

$$R^2 = 1 - \frac{\sum (\hat{y}_i - y_i)^2}{\sum (\bar{y} - \bar{y})^2},$$  

(5)

where $y_i$ represents the real values, $\hat{y}_i$ represents the forecasting values, $\sum (\hat{y}_i - y_i)^2$ represents the sum of squares of the difference between forecasting values and real values in the regression model, and $\sum (\bar{y} - \bar{y})^2$ represents the sum of squares of the differences between the real value and the mean. The $R^2$ value ranges from 0 to 1. The value closer to 1 indicates that the model can better explain the variability of the dependent variable, also indicating a better fit. The value closer to 0 indicates that the model has a weaker ability to explain the dependent variable, indicating a poorer fit.

MAE (Mean Absolute Error) is the average of the absolute difference between the forecasting values and the real values, and it is expressed as

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

(6)

where $n$ denotes the number of sample points, and a smaller MAE indicates a higher forecasting accuracy.
RMSE (Root Mean Squared Error) measures the root mean squared difference between the forecasting values and the real values, which is expressed as

\[
RMSE = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2},
\]  

(7)
where MSE denotes the mean squared error as described in (4). A smaller RMSE indicates higher forecasting accuracy. Compared with MAE, RMSE is more sensitive to the larger forecasting error.

III. EXPERIMENT RESULTS AND ANALYSIS

A. Forecasting accuracy

First, we compare the performance of different attention mechanisms by training the models using the Luong attention mechanism and the Bahdanau attention mechanism. We selected 10,512 sampling points from the winter as the testing dataset. The experiment results are shown in Table IV.

From Table 4, it can be observed that the Luong attention mechanism outperforms the Bahdanau attention mechanism. Additionally, the models with attention mechanisms yield better forecasting results compared to those without attention mechanisms. Therefore, we adopted the Luong attention mechanism as the attention layer in the following experiments.

Then, we compare the SSA-CNN-LSTM-ATT model with some related deep learning models, including LSTM, CNN-LSTM,
CNN-LSTM-ATT, and some machine learning methods such as Support Vector Regression (SVR), Random Forests (RF), and XGBoost (XGB).

From Tables V and VI, we find that, compared with both deep learning models and machine learning methods, our proposed SSA-CNN-LSTM-ATT achieves a significant improvement in terms of $R^2$, which can reach a value of 0.998. Additionally, SSA-CNN-LSTM-ATT yields a smaller error (with a MAE of 266 and a RMSE of 355), indicating that more accurate and stable forecasting results can be obtained by SSA-CNN-LSTM-ATT. The above-mentioned experiments were conducted on the data in three hours. In Sec. III C, we will conduct daily experiments, weekly experiments, and monthly experiments in summer and winter, respectively.

### B. Error metric

As shown in Fig. 6, the average error ratio of our proposed SSA-CNN-LSTM-ATT model is about 0.002, which is obviously lower than that of other models, also indicating that our proposed model has a higher prediction accuracy.

### C. Loss and forecasting results in winter and summer

As shown in Fig. 7, the loss value rapidly decreases and converges, ultimately tending toward 0.

This experiment is conducted at high temperatures in the summer and low temperatures in the winter. The test is divided into three time dimensions: month, week, and day. Our method is compared with the SVR method.

As shown in Figs. 8–10, we first test the winter data. The blue solid line represents the true values, the green dashed line represents the forecasted values of the SVR method, and the orange dashed line represents the forecasting values of our method. From these figures, it is obvious that our method fits better with the true values, whether forecasting for a month, a week, or a day.

Then, we test the summer data, as shown in Figs. 11–13. As shown in these figures, the orange dashed line almost fits perfectly with the blue solid line, while the green dashed line still has a significant gap. This phenomenon shows that our method also has high accuracy in the case of high temperature data in the summer.

### D. Time cost

Experiment results indicate that the average forecasting time (one week) of our proposed model is about 2.33 s, while the average forecasting time (30 days) is about 8.67 s, which are acceptable.

### IV. CONCLUSIONS

This paper proposes a CNN-LSTM model combining the SSA algorithm and the Luong attention mechanism for power load forecasting. Our proposed model takes advantage of the feature extraction ability of CNN and the time sequence modeling advantage of LSTM, adds a Luong attention layer to weight the outputs of LSTM, and then optimizes the number of units of LSTM through the SSA algorithm to yield preferable forecasting results. This paper evaluates the performance of the proposed model by training and testing it on a real-world dataset from Jiangsu Province in 2022. The experiment results show that this model has the lowest MAE and RMSE.
compared with other models or methods, while the $R^2$ value is the highest.

ACKNOWLEDGMENTS

This research is supported by Dispatch Balance and Multiple Data Sources Intelligent Analysis for the Power Grid of State Grid Jiangsu under Grant No. B11002238PD4.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions


DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES