

Research Article

Deep-Learning Prediction Model with Serial Two-Level Decomposition Based on Bayesian Optimization

Xue-Bo Jin ^{1,2,3} Hong-Xing Wang^{1,2,3} Xiao-Yi Wang ^{1,2,3} Yu-Ting Bai,^{1,2,3}
Ting-Li Su,^{1,2,3} and Jian-Lei Kong ^{1,2,3}

¹School of Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China

²China Light Industry Key Laboratory of Industrial Internet and Big Data, Beijing Technology and Business University, Beijing 100048, China

³Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing Technology and Business University, Beijing 100048, China

Correspondence should be addressed to Xue-Bo Jin; jinxuebo@btbu.edu.cn and Xiao-Yi Wang; wangxy@btbu.edu.cn

Received 1 July 2020; Revised 19 August 2020; Accepted 31 August 2020; Published 14 September 2020

Academic Editor: Kang Li

Copyright © 2020 Xue-Bo Jin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The power load prediction is significant in a sustainable power system, which is the key to the energy system's economic operation. An accurate prediction of the power load can provide a reliable decision for power system planning. However, it is challenging to predict the power load with a single model, especially for multistep prediction, because the time series load data have multiple periods. This paper presents a deep hybrid model with a serial two-level decomposition structure. First, the power load data are decomposed into components; then, the gated recurrent unit (GRU) network, with the Bayesian optimization parameters, is used as the subpredictor for each component. Last, the predictions of different components are fused to achieve the final predictions. The power load data of American Electric Power (AEP) were used to verify the proposed predictor. The results showed that the proposed prediction method could effectively improve the accuracy of power load prediction.

1. Introduction

With the rapid development of society, electric power is applied in all aspects of production and life. To ensure average production and living needs, power enterprises always produce more energy than needed. However, due to the nonstorage of power, excess energy will cause a waste of resources, and excessive operation also has an impact on the safety of power equipment [1–3]. Therefore, power load prediction is of considerable significance to power enterprises. The benefits of load prediction include effective planning of annual power supply, reduction of power waste and costs, and development of operation plans. An accurate prediction can provide a reliable decision basis for operation and ensure the power system's sustainable development.

However, in reality, many factors cause the results of power load prediction to be challenging. The power load is a very complex nonlinear time series, making it very difficult

to predict the power load accurately. For example, the weather will affect the cost of power [4]; other factors, such as the differences of the region's development levels and the unpredictable natural disasters [5], can also cause various changes in the power load.

The power load data generally contain the following four components:

- (1) Trend component: it reflects the main trend of power load data, which mainly contains upward and downward trends. The trend component is the basic level of power load for a long time. If the power load increases, the component is an upward trend, and if the power load decreases, the component will have a downward trend.
- (2) Daily period component: the power load data have distinct period characteristics in a day, that is, high power load in the day and low power load at night.

- (3) Annual period component: the power load data have another period component in a year, and the power load will change in different months.
- (4) Residual component: this part is the remains after removing the trend and periodic component from the original data, which contain complex nonlinear data and noise.

Figure 1(a) shows the power load data of the United States Electric Power Company (AEP) from January 1, 2017, to January 1, 2020. The horizontal coordinates in the graph indicate that the sampling interval is one hour. Figure 1(b) shows the trend component separated from the original data. Figure 1c(c_1) shows the overall daily period component, while Figure 1c(c_2) shows a portion of our excerpted July 2018 daily period component as a display, and we can see that the daily period component has a distinct period. Figure 1(d) shows the annual period component. There is too much electricity in winter and summer, while the power load in spring and autumn is low.

In recent years, an accurate prediction of time series has become the focus of researchers. Time series usually have nonlinear, nonstationary, and complex period characteristics [6, 7]. Existing methods for predicting time series include statistical prediction, machine learning, and combined prediction [8–10]. The statistical prediction methods are usually based on mathematical models [11, 12]. The establishment of prediction models often includes the regression analysis method [13], the gray model method [14, 15], support vector machine (SVM) [16, 17], autoregressive integrated moving average (ARIMA) [18], and artificial neural network (ANN) [19]. These methods often find it difficult to obtain accurate predictions when dealing with complex nonlinear data.

Unlike the method described earlier, deep learning does not require the preknown information and has stronger learning ability and prediction ability. For example, Tang et al. [20] presented a multilayer two-way recursive neural network based on LSTM and GRU; Gao et al. [21] used GRU to build models for short-term load prediction. Guo et al. [22] presented an integrated in-depth learning method that integrates multiple LSTM networks to develop large-span cycles using LSTM's nonlinear mode and similar day methods. Kollia and Kollias [23] proposed using deep convolution-recursive neural networks to process data in time series or two-dimensional information to improve prediction accuracy. Yin et al. [24] proposed a three-state energy model, which includes three states: generator, power load, and closed status. And a kind of scalable deep learning is proposed for real-time economic power generation scheduling and control based on the three-state energy of the future smart grid. Zhang et al. [25] presented a prediction model based on the restricted Boltzmann machine and Elman. He et al. [26] proposed a deep belief network (DBN) embedded in a parametric copula model. Although these in-depth learning methods' accuracy is improved compared with the traditional methods, it is still difficult to learn enough feature representation in the face of nonstationary data.

Based on the above research, the latest research combines decomposition methods with deep learning to achieve better prediction results. These methods decompose the original data into components, use different prediction methods to predict the decomposed components, and fuse the predicted results to achieve better prediction results. For example, the seasonal-trend decomposition procedure based on loess (STL) can obtain trend, seasonal, and residual components of the complex data [27], which have been used as a hybrid prediction in the authors' former research for weather forecasting [8, 28]. Another decomposition method named as wavelet decomposition, Wang et al. [29], decomposed the original time series to construct the predictor for different sub-signals. Li et al. [30] proposed to use an extreme learning machine (ELM) combined with the variational mode decomposition (VMD). Guo et al. [31] proposed to decompose the original sequence by empirical mode decomposition (EMD) and select different models (including AR, MA, and ARMA) based on different characteristics of the subcomponents. The authors had used the EMD to decompose the PM 2.5 time series data to obtain more accurate forecasting [32, 33]. Compared with VMD and EMD, STL decomposition can guarantee to give a known number of components (including 3 components, such as trend, seasonal, and residual components) and is particularly suitable for sequential data with periodicity.

Comparing statistical and machine learning methods, the results show that the prediction accuracy can be improved by decomposing into multiple components and modeling separately with predictors. Moreover, it is also found that the hyperparameters have a significant impact on the prediction performance. Therefore, optimization methods of hyperparameters have been used. For example, [34–36] decomposed the original data based on a wavelet algorithm and predicted the components by a particle swarm optimization (PSO) neural network. Another optimization method named as the fruit fly optimization algorithm (FOA) is used to select parameters for the generalized regression neural network (GRNN) [37]. In contrast, He et al. [38] used the Bayesian optimization algorithm based on a Parzen estimator to optimize the hyperparameters of the quantile regression forest (QRF) predictor. FOA and PSO are group optimization algorithms, which are not suitable for model hyperparameter tuning because they need to have enough initial sample points, and the optimum efficiency is not high. However, for the training process of deep learning models, we need to desample as little as possible to improve the optimum efficiency. Therefore, the Bayesian optimization algorithm is widely used in the deep learning models as it can obtain the optimal global value with the fewest sampling points.

This paper uses a serial two-level decomposition structure to improve the prediction performance due to the complexity of multiple periods of the power load data. Furthermore, the Bayesian optimization algorithm is applied to optimize the hyperparameters of the model. The innovation contribution of this paper is shown as follows:

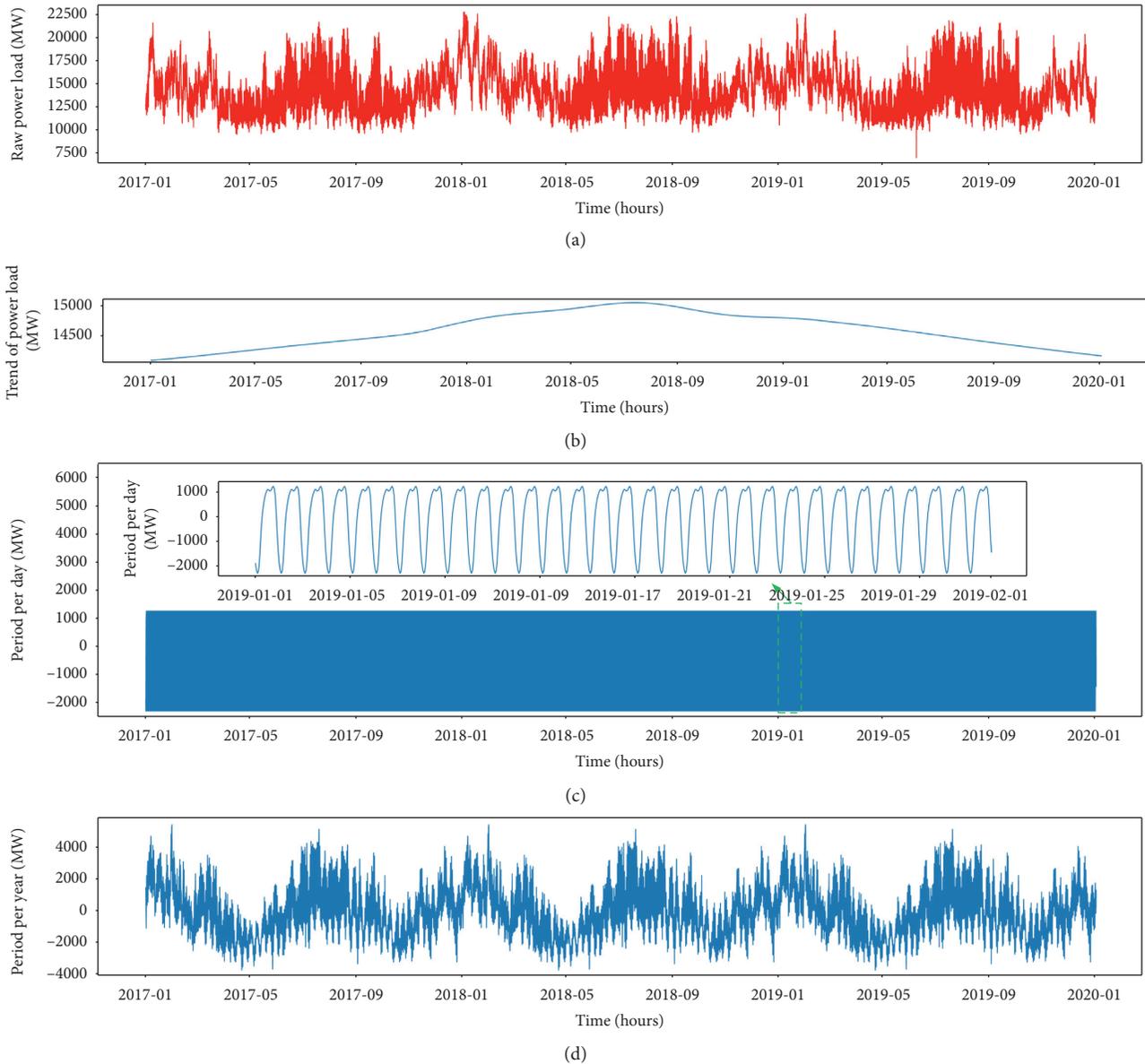


FIGURE 1: Decomposition of the electric load data.

- (1) Bayesian sequential model-based optimization (SMBO) algorithm is used to optimize model parameters to improve the model prediction performance.
- (2) According to the double-period characteristics of power load data, the original data are decomposed with a serial two-level decomposition structure. Four GRUs are used for the trend component, daily period component, annual period component, and residual component.

The rest is arranged as follows. Section 2 introduces the proposed serial two-level decomposition and the model used to realize the prediction in detail. In Section 3, we give the experimental results of the SMBO optimization algorithm

and compare them with other experiments. At last, Section 4 summarizes the conclusion.

2. Serial Two-Level Decomposition Optimization Model

The model consists of decomposition, prediction, and fusion processes. The prediction model's framework is shown in Figure 2, in which the two-level decomposition structure is used at first, and the data are decomposed into four components. In the training phase, GRUs are used to train these four components. In the prediction phase, GRUs are used to predict four components. Finally, the results of each sub-model are fused to get the final prediction results.

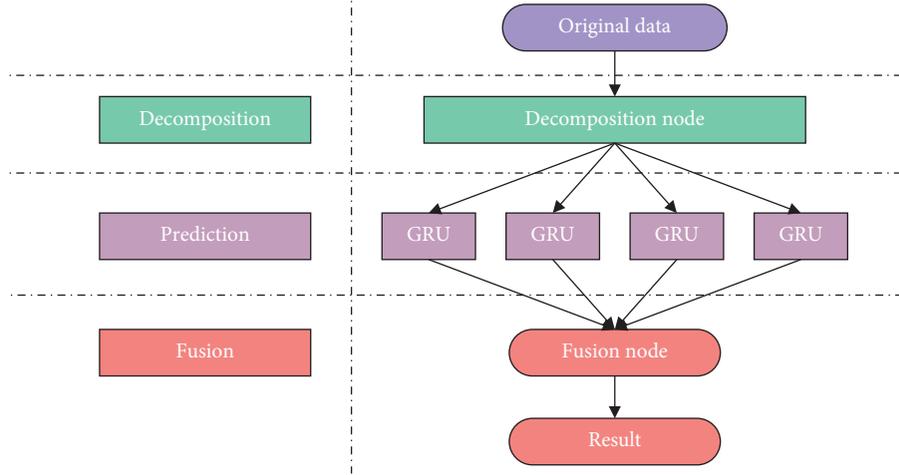


FIGURE 2: Frame diagram of the prediction part.

2.1. Serial Two-Level Decomposition. The time-series data of the original electric load are decomposed with two levels. Figure 3 shows the detailed information of the decomposition node of Figure 2. After the first-level decomposition, the three components of trend, period, and residual are obtained. However, the residual data still contain trend and periodic information. Therefore, the residual obtained by the first decomposition is decomposed again. Similarly, three sets of data, new trends, new periods, and new residuals resulting from the second decomposition, can be obtained.

After the first-level decomposition of the original electric power load data, trend TD_t , period PD_t , and residual RD_t are obtained. The second decomposition of residual RD_t was carried out to obtain trend TY_t , period PY_t , and residual RY_t . Finally, the components with the same characteristics are combined to obtain the final three decomposition results: trend T_t , period P_t , and residual R_t .

2.1.1. First-Level Decomposition. Power load data P_t are a discrete-time series with a length of N , which means $t = 1, 2, 3, \dots, N$, so three sets of data, trend, period, and residual, can be represented as discrete:

$$P_t = PD_t + RD_t + TD_t, \quad t = 1, 2, 3, \dots, N, \quad (1)$$

where PD_t , RD_t , and TD_t are the period component, the residual component, and the trend component. Detailed decomposition steps can be represented as follows:

- (1) In the first-level decomposition, the power load of a day has a potential period, so the decomposition period is set to be 1 day. For the hourly data used here, the number of periods is set to 24. $Num_d = [N/24]$ is used to calculate the number of periods; the meaning of $[a]$ in the formula is to round up the input data a .
- (2) Utilizing the average regression method, the trend component TD_t , which can express the overall trend of time series, is extracted from the original data P_t .

- (3) The following two steps are followed to obtain the period component of the original data P_t :

- (a) Calculate the initial value of the periodic component through $XD_t = P_t - TD_t$.
- (b) Because $[a]$ is described to round up a , $Num_d \times 24$ and N may not be the same. Therefore, instead of making selection on all the data, select data 1 st to $Num_d \times 24$ from the data XD_t , superpose the data with the point of the same time, and divide by Num_d to get a periodic curve, which is duplicated Num_d times so that the periodic component PD_t with the same N points can be obtained.

- (4) Raw data minus period and trend data gives residual RD_t , such as $RD_t = P_t - PD_t - TD_t$.

2.1.2. Second-Level Decomposition. Considering that the first-level decomposition does not completely decompose the original data P_t and the residual data RD_t still contain rich periodic information and trend information, the second-level decomposition of the residual RD_t obtained from the first-level decomposition is carried out in the similar method. Furthermore, we can get a second set of data that represents the annual trend, the annual period, and the residual:

$$RD_t = PY_t + RY_t + TY_t, \quad t = 1, 2, 3, \dots, N, \quad (2)$$

where PY_t , RY_t , and TY_t are the period component per year, residual component, and trend component, respectively. Detailed decomposition steps can be represented as follows:

- (1) The period of the second decomposition is set to 1 year, 8760 hours. The number of periods is then calculated using $Num_y = [N/8760]$, where $[a]$ is described as rounding a .
- (2) Utilizing the average regression method, the trend component TY_t , which can express the overall trend

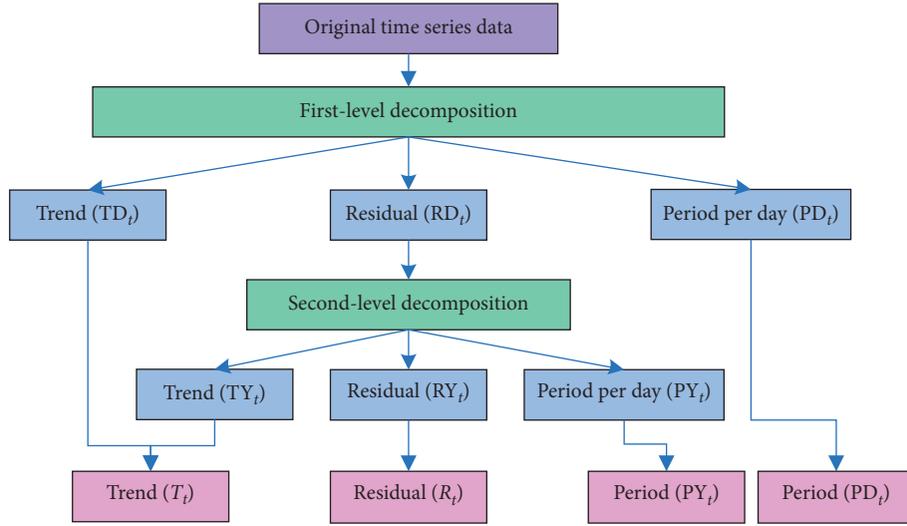


FIGURE 3: Structure of the decomposition node.

of time series, is extracted from the residual data RD_t .

- (3) The following three steps are followed to obtain the period component of the residual data RD_t :
 - (a) Use $XY_t = RD_t - TY_t$ to calculate the initialization period component of the input data.
 - (b) Select data 1st to $\text{Num}_y \times 8760$ from the data XY_t , superpose the data with the point of the same time, and divide by Num_y to get a periodic curve, which is duplicated Num_y times so that the periodic component XPY_t with the same N points can be obtained.
 - (c) The window size of XPY_t is 24 hours, and the mean value is used instead of the original data to obtain a new periodic component PY_t of length N .
- (4) Daily residual data minus period and trend data gives residual RY_t , such as $RY_t = RD_t - PY_t - TY_t$.

2.2. Subpredictor. After two-level decomposition, the five components of residual RY_t , trend TD_t , trend TY_t , period PD_t , and period PY_t were obtained. Trend TD_t and trend TY_t represent the linear trend of the data, and then TD_t and TY_t are combined to get the trend of the data. Therefore, four groups of subdata are finally obtained. Four GRU networks are trained as predictors for each subdata.

GRU network is the development of LSTM models. It simplifies the structure of the model, reduces the network parameters that need to be trained, and inherits the ability of LSTM to solve long-dependency problems. Hence, GRU is a good model structure for prediction. The GRU model consists of two parts, the update gate and the reset gate, and each GRU cell is structured, as shown in Figure 4.

The update gate's function is to adjust the information transmission from the previous moment to the current moment. The smaller the value, the less information from the last moment to the present moment. The goal of the reset

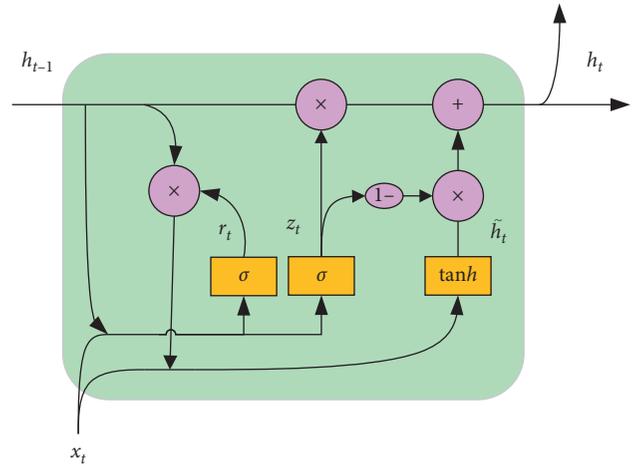


FIGURE 4: The detailed structure within a GRU cell.

gate is to adjust the degree of ignored information from the previous moment. The larger the reset gate value is, the less information is ignored so that the new input can be fused with more stored information.

The formula for the forward propagation of input data in each GRU is shown in the following:

$$\begin{aligned}
 z_t &= \sigma(x_t U^z + h_{t-1} W^z + b^z), \\
 r_t &= \sigma(x_t U^r + h_{t-1} W^r + b^r), \\
 \tilde{h}_t &= \tanh(x_t U^h + (h_{t-1} \odot r_t) W^h + b^h), \\
 h_t &= (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1},
 \end{aligned} \tag{3}$$

where x_t is the input; z_t , r_t , \tilde{h}_t , and h_t represent the candidate state of the update gate, the reset gate, the hidden node at the t -th time point, and output state of the hidden node at the t -th time point; U and W represent the weight in the model; b represents the bias; \odot represents the multiplication of elements; and σ and \tanh are the activation functions used

in the cell, and the mathematical expression of the activation functions is as follows:

$$\begin{aligned}\sigma(x) &= \frac{1}{1 + e^{-x}}, \\ \tan h(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}}.\end{aligned}\quad (4)$$

According to the structure of the above GRU cell and the relationship between the input and output data, a GRU network is built and shown in Figure 5. The network includes multiple GRU units, and the number of network layers is 2. As shown in Figure 5, X_t ($t = 1, 2, \dots, N$) is the input of the GRU network, Y_{m+t} ($t = 1, 2, 3, \dots, N$) is the output, and m is the number of GRU cells in each layer.

2.3. Sequence Model-Based Optimization (SMBO). Before deep-learning model training, we need to initialize the hyperparameters of the model, which can improve the model's prediction performance. As to the two decomposition prediction models for the trend component and daily period component, the traditional parameter selection method can achieve prediction by the network's initialization parameters. However, the annual period component and residual data are complicated, so here, we use one of the Bayesian optimizations, named the SMBO algorithm [39].

SMBO needs an objective function and then updates the posterior distribution of the parameter space objective function. Here, the objective function $L(x)$ is selected as the root mean square error (RMSE), that is,

$$L(k) = \sqrt{\frac{\sum_{i=1}^{\text{num}} (y(k_i) - \hat{y}_i)^2}{\text{num}}}, \quad (5)$$

where num is the number of input hyperparameter group k , $y(k_i)$ is the prediction result obtained by the model using hyperparameter combination k_i , and \hat{y}_i is the real value. Then, for the SMBO algorithm, we have

$$k^* = \arg \min_{k \in K} L(k), \quad (6)$$

where k^* is the best parameter combination determined by the SMBO algorithm, k is a set of input hyperparameters, and K is the parameter space of multidimensional hyperparameters.

The update of the parameter space includes two steps: Gaussian process (GP) and hyperparameter selection. In the Gaussian process, the algorithm realizes the modeling and fitting optimization of the objective function and obtains the posterior distribution corresponding to the input k ; in the process of hyperparameter selection, "development" and "exploration" are used to realize the process of finding the optimal parameter with the minimum cost. "Exploration" refers to finding appropriate parameters in the unsampled hyperparameter space, which often leads to the global optimal parameter combination. "Development" will search the last set of the hyperparameter space according to the posterior probability distribution. When the set objective function $L(k)$ follows the Gaussian distribution,

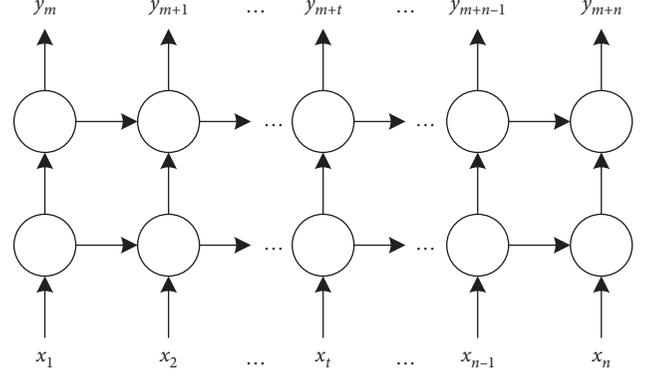


FIGURE 5: Network structure of the gated recurrent unit (GRU).

$$L(k) \sim \text{GP}(\mu(k), O(k, k')), \quad (7)$$

where $\mu(k)$ is the average value of $O(k, k')$ and $L(k)$, $O(k, k')$ is the covariance matrix of $L(k)$, GP is expressed as a Gaussian process, $L(k)$ is the objective function, $A \sim B$ is the distribution of A obeying B , and initialization can be expressed as

$$O = \begin{bmatrix} o(k_1, k_1) & \cdots & o(k_1, k_i) \\ \vdots & \ddots & \vdots \\ o(k_i, k_1) & \cdots & o(k_i, k_i) \end{bmatrix}. \quad (8)$$

During parameter searching of the SMBO algorithm, the covariance matrix of the Gaussian process will change with the number of iterations. If the hyperparameter group entered in step $i + 1$ is k_i , the covariance matrix can be expressed as

$$O' = \begin{bmatrix} O & o^T \\ o & o(k_{i+1}, k_{i+1}) \end{bmatrix}, \quad (9)$$

where $o = [o(k_{i+1}, k_1), o(k_{i+1}, k_2), \dots, o(k_{i+1}, k_i)]$; then, the posterior probability of the objective function $L(k)$ can be obtained successively:

$$P(L_{i+1} | D_{i+1}, k_{i+1}) \sim N(\mu_{i+1}(k), \sigma_{i+1}^2(k)), \quad (10)$$

where D is the observation data, $\mu_{i+1}(k)$ is the average value of $L(k)$ in the $i + 1$ step, $\sigma_{i+1}^2(k)$ is the variance of $L(k)$ in the $i + 1$ step, $P(L_{i+1} | D_{i+1}, k_{i+1})$ is the probability of obtaining the objective function L in the case of step $i + 1$ data D and parameter group k , N is the normal distribution, and $A \sim B$ is the distribution of A obeying B . The next step is to find the best parameter through hyperparameter selection after the posterior probability is obtained. The upper confidence bound (UCB) acquisition function is used in the "development" in this paper:

$$k_{i+1} = \arg \max H(k | D_i) = \arg \max \mu(k) + \zeta_{i+1}^{1/2} \sigma_i(k), \quad (11)$$

where ζ_{i+1} is a constant, $H(k | D_i)$ is the UCB acquisition function, and k_{i+1} is the hyperparameter group selected in step $i + 1$.

The SMBO algorithm is shown in Algorithm 1.

3. Experiment Results and Discussion

In this study, the electric power load data are from American Electric Power Company (AEP), which includes 26,280 data from January 1, 2017, to January 1, 2020. In the experiment, the first 70% data were selected as the training set data and the remaining 30% as the test set data. The data are first decomposed and then normalized for each subcomponent separately. In the training process, trend T_t and period PD_t are put directly into different subpredictors for training, while period PY_t and residual RY_t are first subjected to hyperparameter seeking by the SMBO algorithm and then put into different subpredictors for training. In the testing process, different components are put directly into the trained subpredictors to get the final prediction results, and the results from different subpredictors are fused to get the final prediction results. Then, the power load prediction is used to plan the precision supply of the power and development of operation plans. The overall system flow diagram is shown in Figure 6.

3.1. Experimental Setup. The predictor uses Keras to build a learning model. All models were trained and tested on a PC server with Intel Core-CPU i7-2.21 GHz processor with 32 G RAM. In deep learning, many hyperparameters need to be set (for example, the number of network layers, weight

initialization, and learning rate). The GRU network structure is set to two layers.

For the complex components such as the annual periodic component and residual component, this paper uses the SMBO algorithm to find some optimal hyperparameters of the network. The rest of the parameters use the Keras default initialization parameters to obtain the model parameters by optimizing the predetermined optimization function.

For components such as the daily period component and trend component, the GRU model uses the Nadam optimization algorithm, and all parameters use Keras default values. Activation functions in the network use \tanh and ReLU. The learning and prediction steps are set to 24; that is, the model uses the previous day's power load to predict the next day's power load. Predicting one day in advance can help related departments get a general idea of the next day's power load and make appropriate plans based on the load.

In this study, five indexes are used to evaluate the performance of the model, including root mean square error (RMSE), normalized mean square error (NRMSE), mean absolute error (MAE), symmetric mean absolute percentage error (SMAPE), and Pearson correlation coefficient (R). The smaller the first four of the five indicators, the more accurate the model prediction. The larger the value of the fifth indicator (R), the better the fitting effect between the observed value and the predicted value. The calculation formula of these five indicators is as follows:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}, \\ \text{NRMSE} &= \frac{1}{\max(y) - \min(y)} \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}, \\ \text{MAE} &= \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \text{SMAPE} = \frac{1}{n} \sum_{i=1}^n \frac{|\hat{y}_i - y_i|}{(|\hat{y}_i| + |y_i|)/2}, R = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \end{aligned} \quad (12)$$

where n is the number of samples, \hat{y} is the ground-truth value of the power load, $\bar{\hat{y}}$ is the average of the ground-truth value, y is the predicted value, and \bar{y} is the average of the prediction.

3.2. Hyperparameter Selection Based on Bayesian Optimization. Table 1 shows the superparameter space of the SMBO algorithm. The selected hyperparameters include the number of neurons in the first layer, batch size, and optimizer. All the superparameter groups are tested in the model with an epoch of 100, and the optimized group of hyperparameters is finally obtained for subsequent network training.

Table 2 shows the comparison of the prediction performance of the Bayesian parameter optimization and no-optimization method for the annual periodic component PY_t and the residual component RY_t . According to Table 2, it can be observed that the Bayesian optimization can significantly improve the effectiveness of the model. For example, the RMSE of the annual period component PY_t declined by 25.9% from 471.25 to 349.0855. Similarly, the residual component RY_t fell 5.5% from 606.5918 to 572.7338.

The comparison between the predicted power load for December 12 to 23, 2019, and the real power load is shown in Figure 7. We can see that the weekend power load is lower, while the weekday power load is higher. As to one day, the

Input: $L(k)$ is the root mean square error of the proposed model, T is the number of selected hyperparameter groups, H is the UCB acquisition function, X is the input data, S is the proposed model, x is the input hyperparameter group.

Output: returns the optimal hyperparameter group x^* .

$D \leftarrow \text{InitSamples}(L(k), X)$

for $i \leftarrow |D|$ to T do

Model the objective function and calculate the posterior probability. $P(L|D, k) \leftarrow \text{Fitmodel}(S, D)$

Parameter group selection using the UCB acquisition function. $k_i \leftarrow \text{argmax}_{k \in K} H(k | P(L|D, k))$

Using superparameter group k^* to train network to get the prediction $y_i \leftarrow L(k_i)$

Update data set $D \leftarrow D \cup (k_i, y_i)$

end for

$k^* = \text{argmin}_{k \in K} L(k)$

return k^*

ALGORITHM 1: Sequential model-based optimization.

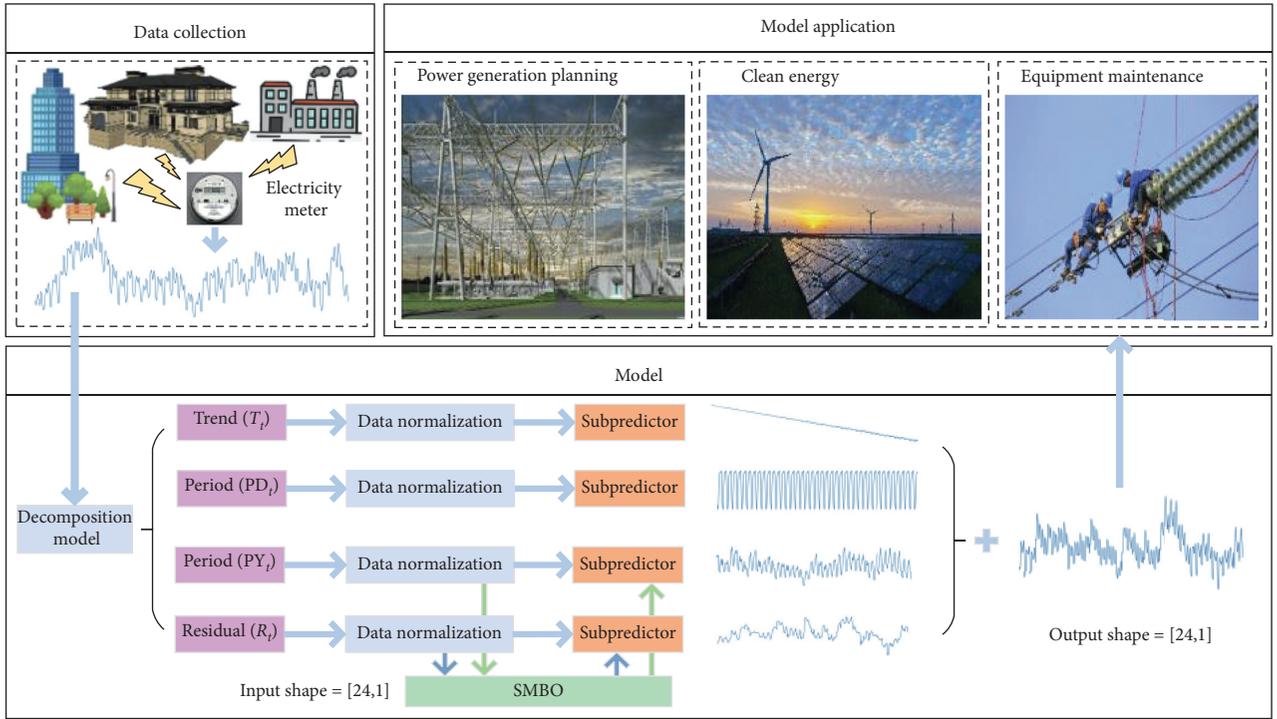


FIGURE 6: Load prediction system.

TABLE 1: Bayesian optimization hyperparameter space and search results.

Hyperparameters	Type	Range of values	The optimized hyperparameter for PY_t	The optimized hyperparameter for RY_t
No. 1 hidden units	Integer	{48, 42, 36, 30, 24}	36	48
Batch size	Integer	{1, 2, 4, 8, 16, 32}	32	2
Optimizer	Categorical	{Adam, Nadam, Adadelta}	Nadam	Nadam

TABLE 2: Comparison of the RMSE between Bayesian optimization and normal methods.

	Bayesian optimization	No-optimization
PY_t	349.0855	471.2532
RY_t	572.7338	606.5918

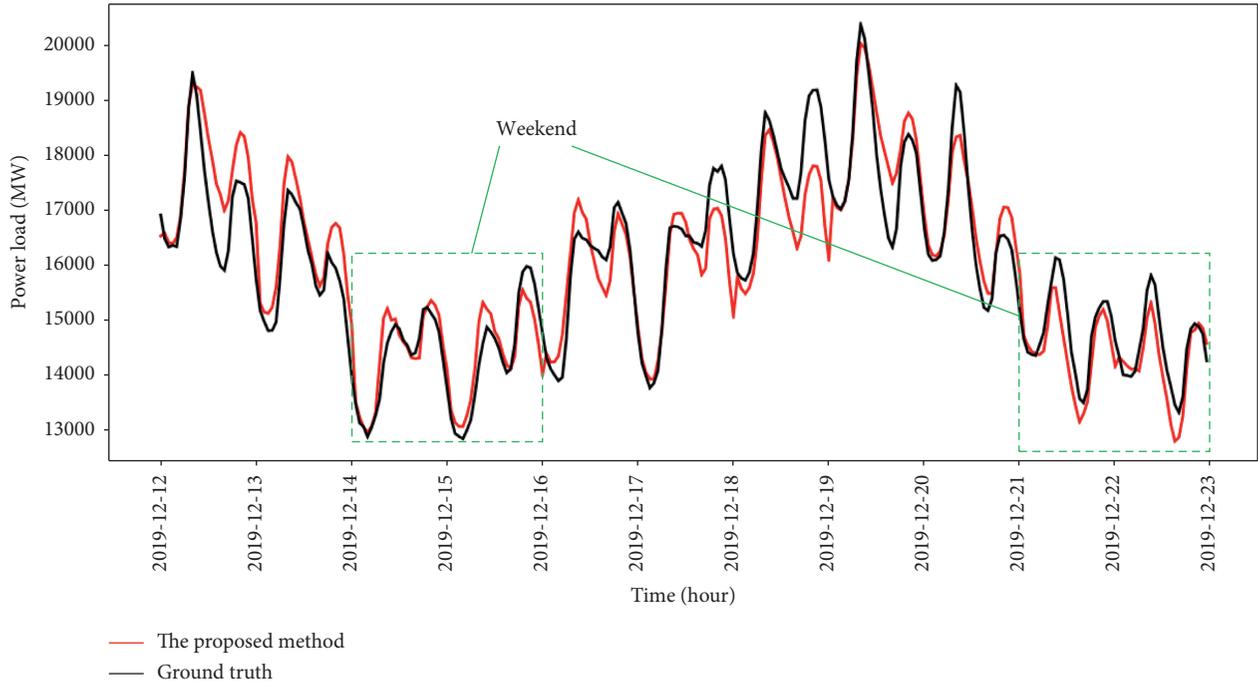


FIGURE 7: Comparison results between the predicted value and the ground truth of the electric load in December 2019.

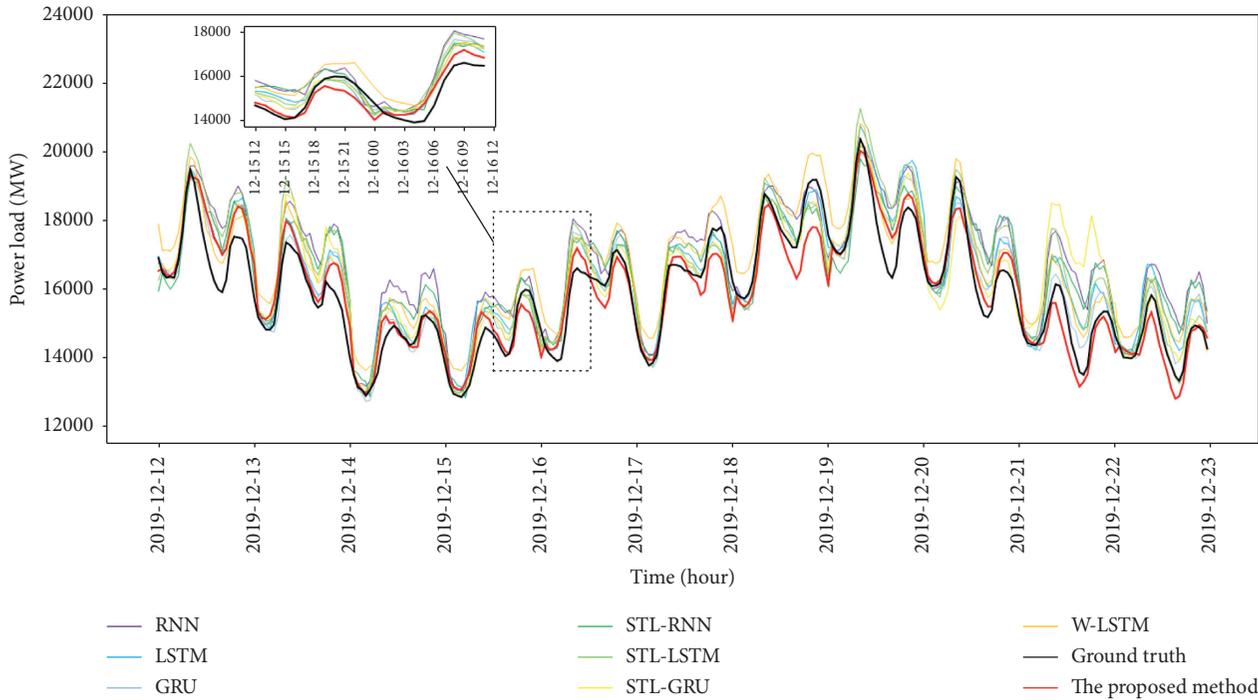


FIGURE 8: Prediction results of the power load by different models.

morning and afternoon are higher, and the noon is lower. Therefore, it is reasonable that the proposed method decomposes the original power load data twice, taking into account the periodicity per day and year.

3.3. Comparison of Prediction Results with Different Models. In the setup experiment, we compare the performance of the proposed method with seven models, i.e., recurrent neural network (RNN) [40], long short-term memory (LSTM) [41],

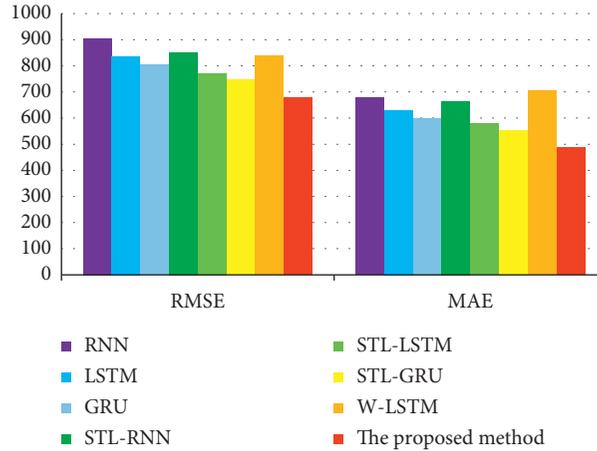


FIGURE 9: RMSE and MAE of different models.

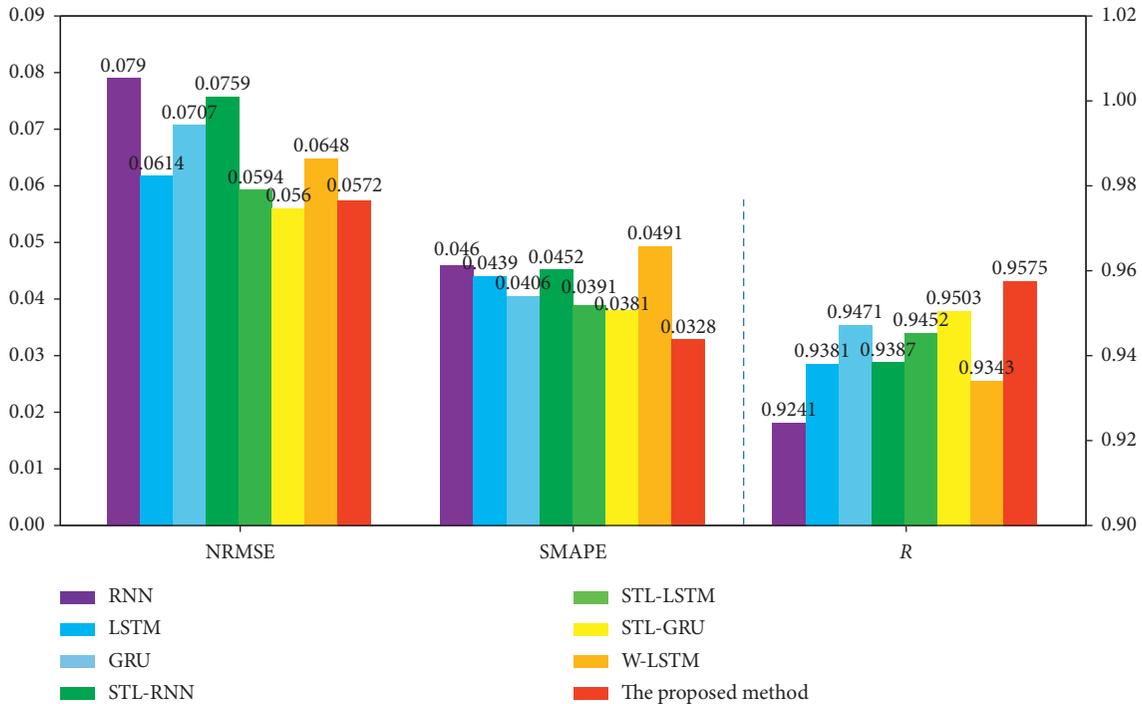


FIGURE 10: NRMSE, SMAPE, and R of different methods.

GRU [42], STL-RNN (RNN based on STL), STL-LSTM (RNN based on STL) [43], STL-GRU (GRU based on STL) [8], and wavelet-LSTM (W-LSTM) [44]. The hourly power load data used are from February 6, 2019, to December 31, 2019. The partial prediction results of each model are shown in Figure 8. It can be seen from the figure that the proposed model has the best performance.

Figures 9 and 10 show the comparison results of the five indicators, respectively. From Table 3, we can see that the RMSE of RNN, LSTM, and GRU is 905.4590, 835.8678, and 805.0688, respectively, and the MAEs are 677.6044, 630.6519, and 599.9143, respectively. Through STL

TABLE 3: The indicators of different models in the same test set.

Model	RMSE	NRMSE	MAE	SMAPE	R
RNN [40]	905.4590	0.0790	677.6044	0.0460	0.9241
LSTM [41]	835.8678	0.0614	630.6519	0.0439	0.9381
GRU [42]	805.0688	0.0707	599.9143	0.0406	0.9471
STL-RNN	851.9837	0.0759	664.7142	0.0452	0.9387
STL-LSTM [43]	771.5973	0.0594	579.3349	0.0391	0.9452
STL-GRU [8]	747.3044	0.0560	554.4263	0.0381	0.9503
W-LSTM [44]	837.0609	0.0648	705.9575	0.0491	0.9343
The proposed method	676.6433	0.0572	486.0197	0.0328	0.9575

decomposition, the RMSE of STL-RNN, STL-LSTM, and STL-GRU RMSE is 851.9837, 771.5973, and 747.3044, respectively, decreasing by 5.9%, 7.6%, and 7.2%. Moreover, MAEs are 664.7142, 579.3349, 554.4263, respectively, decreasing by 1.9%, 8.1%, and 7.6%, respectively. Therefore, it can be seen that the decomposition method is effective to improve the prediction performance.

On the contrary, the results show that the GRU network has the best prediction performance. For example, compared with the RMSE of RNN and LSTM, the RMSE of GRU is reduced by 11.1% and 3.7%, respectively. As to the decomposition method, compared with STL-RNN and STL-LSTM, the RMSE of STL-GRU is decreased by 12.3% and 3.1%, respectively. It validates the correctness of selecting GRU as the subpredictor in this paper.

Furthermore, we find that the serial two-level decomposition is rational, and the proposed model works best, which obtains the least of RMSE 676.6433, MAE 486.0197, and SMAPE 0.0328, the highest R 0.9575, and the second-least NRMSE 0.0572. It is believed that the original data contain nonlinear information; after two serial decompositions, the complex periodic information and trend information are predicted separately, which can better fit the data, and the combination can obtain a predicted result in better performance. The proposed deep-learning prediction models in this paper can combine some parameter estimation algorithms [45–51] such as the iterative algorithms [52–57] and the recursive algorithms [58–64] to study new modeling and prediction approaches for different engineering application problems [65–69] such as system modeling, information processing, and transportation communication systems.

4. Conclusions

More accurate results of power load prediction can make the power generation companies and power operation companies better control the operation status, facilitate the regulation of the market, save costs, and prevent pollution. This study uses a serial two-stage decomposition structure to decompose the electric power load time series according to different periods, which reduces the complex nonlinear relationship of the original data of the electric power load. The overall trend component indicates that the electric power load data change slowly, which can be understood as the electric power load remains at a certain level for a long time. The daily period component indicates daily variation, higher in the day and slightly lower in the night; furthermore, as to one year, it is lower in winter and higher in summer, all of which correspond to the actual use of electricity.

After decomposing the raw power load sequence, the GRU is used to build the model prediction component. The predictions from the subpredictors are then fused to obtain a more accurate prediction. After a two-stage decomposition of the data, the trend information in the original complex time series, as well as the multiple period information, is separated into subsequences. Different prediction models for subsequences were built based on their different

characteristics to obtain the final prediction results. The proposed prediction methods in this paper can be applied to other literature studies [70–76] for different purposes. In the future research, to further improve the model's performance, new network structures would be adopted, and other decompositions or combined methods would be tried. The model proposed in this study can be applied not only to power prediction but also to other data that contain multiple period information.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (nos. 61673002, 61903009, and 61903008), the Beijing Municipal Education Commission (nos. KM201910011010 and KM201810011005), the Young Teacher Research Foundation Project of BTBU (no. QNJJ2020-26), and Beijing Excellent Talent Training Support Project for Young Top-Notch Team (no. 2018000026833TD01).

References

- [1] K. Liu, Y. Shang, Q. Ouyang, and W. D. Widanage, "A data-driven approach with uncertainty quantification for predicting future capacities and remaining useful life of lithium-ion battery," *IEEE Transactions on Industrial Electronics*, p. 1, 2020.
- [2] K. Liu, X. Hu, Z. Wei, Y. Li, and Y. Jiang, "Modified Gaussian process regression models for cyclic capacity prediction of lithium-ion batteries," *IEEE Transactions on Transportation Electrification*, vol. 5, no. 4, pp. 1225–1236, 2019.
- [3] K. Liu, Y. Li, X. Hu, M. Lucu, and W. D. Widanage, "Gaussian process regression with automatic relevance determination Kernel for calendar aging prediction of lithium-ion batteries," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 3767–3777, 2020.
- [4] J. Hong and W. S. Kim, "Weather impacts on electric power load: partial phase synchronization analysis," *Meteorological Applications*, vol. 22, no. 4, pp. 811–816, 2015.
- [5] A. Arab, A. Khodaei, S. K. Khator, and Z. Han, "Transmission network restoration considering AC power flow constraints," in *Proceedings of the 2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 816–821, Miami, FL, USA, 2015.
- [6] Y. Bai, X. Jin, X. Wang, T. Su, J. Kong, and Y. Lu, "Compound autoregressive network for prediction of multivariate time series," *Complexity*, vol. 2019, Article ID 9107167, 11 pages, 2019.
- [7] Y. Bai, X. Jin, X. Wang, X. Wang, and J. Xu, "Dynamic correlation analysis method of air pollutants in spatio-temporal analysis," *International Journal of Environmental Research and Public Health*, vol. 17, no. 1, p. 360, 2020.

- [8] X. Jin, X. Yu, X. Wang, Y. Bai, T. Su, and J. Kong, "Deep learning predictor for sustainable precision agriculture based on internet of things system," *Sustainability*, vol. 12, no. 4, p. 1433, 2020.
- [9] L. Wang, T. Zhang, X. Jin et al., "An approach of recursive timing deep belief network for algal bloom forecasting," *Neural Computing and Applications*, vol. 32, no. 1, pp. 163–171, 2020.
- [10] L. Wang, T. Zhang, X. Wang et al., "An approach of improved multivariate timing-random deep belief net modelling for algal bloom prediction," *Biosystems Engineering*, vol. 177, pp. 130–138, 2019.
- [11] B. Ryabko, J. Astola, and A. Gammerman, "Application of Kolmogorov complexity and universal codes to identity testing and nonparametric testing of serial independence for time series," *Theoretical Computer Science*, vol. 359, no. 1–3, pp. 440–448, 2006.
- [12] B. Huchuk, S. Sanner, and W. O'Brien, "Comparison of machine learning models for occupancy prediction in residential buildings using connected thermostat data," *Building and Environment*, vol. 160, Article ID 106177, 2019.
- [13] L. Duan, D. Niu, and Z. Gu, "Long and medium term power load forecasting with multi-level recursive regression analysis," in *Proceedings of the 2008 2nd IEEE International Symposium on Intelligent Information Technology Application*, pp. 514–518, Shanghai, China, 2008.
- [14] L. X. Dai and F. F. Hu, "Application optimization of grey model in power load forecasting," *Advanced Materials Research*, vol. 347–353, pp. 301–305, 2011.
- [15] W. Li, Z. G. Zhang, and N. Yan, "The mid-long term power load forecasting based on gray and SVM algorithm," *Advanced Materials Research*, vol. 143–144, pp. 1164–1169, 2010.
- [16] Y. H. Zhao, X. C. Zhao, and W. Cheng, "The application of chaotic particle swarm optimization algorithm in power system load forecasting," *Advanced Materials Research*, vol. 614–615, pp. 866–869, 2012.
- [17] N. Ye, Y. Liu, and Y. Wang, "Short-term power load forecasting based on SVM," in *Proceedings of the World Automation Congress 2012*, pp. 47–51, Puerto Vallarta, Mexico, 2012.
- [18] K. Im and J. Lim, "A design of short-term load forecasting structure based on ARIMA using load pattern classification," *Communications in Computer and Information Science*, Springer, vol. 185, pp. 296–303, Berlin, Germany, 2011.
- [19] B. Zhao, Y. Liang, X. Gao, and X. Liu, "Short-term load forecasting based on RBF neural network," *Journal of Physics: Conference Series*, vol. 1069, Article ID 012091, 2018.
- [20] X. Tang, Y. Dai, T. Wang, and Y. Chen, "Short-term power load forecasting based on multi-layer bidirectional recurrent neural network," *IET Generation, Transmission & Distribution*, vol. 13, no. 17, pp. 3847–3854, 2019.
- [21] X. Gao, Y. Wang, Y. Gao et al., "Short-term load forecasting model of GRU network based on deep learning framework," in *Proceedings of the 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pp. 1–4, Beijing, China, 2018.
- [22] H. Guo, L. Tang, and Y. Peng, "Ensemble deep learning method for short-term load forecasting," in *Proceedings of the 2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, pp. 86–90, Shenyang, China, 2018.
- [23] I. Kollia and S. Kollias, "A deep learning approach for load demand forecasting of power systems," in *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 912–919, Bangalore, India, 2018.
- [24] L. Yin, Q. Gao, L. Zhao, and T. Wang, "Expandable deep learning for real-time economic generation dispatch and control of three-state energies based future smart grids," *Energy*, vol. 191, Article ID 116561, 2020.
- [25] X. Zhang, R. Wang, T. Zhang, Y. Liu, and Y. Zha, "Short-term load forecasting using a novel deep learning framework," *Energies*, vol. 11, no. 6, p. 1554, 2018.
- [26] Y. He, J. Deng, and H. Li, "Short-term power load forecasting with deep belief network and Copula models," in *Proceedings of the 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, pp. 191–194, Hangzhou, China, 2017.
- [27] R. B. Cleveland, W. S. Cleveland, J. E. Mcrae, and I. Terpenning, "STL: a seasonal-trend decomposition procedure based on loess," *Journal of Official Stats*, vol. 6, no. 1, pp. 3–33, 1990.
- [28] X. Jin, N. Yang, X. Wang, Y. Bai, T. Su, and J. Kong, "Integrated predictor based on decomposition mechanism for PM_{2.5} long-term prediction," *Applied Sciences*, vol. 9, no. 21, p. 4533, 2019.
- [29] H. Wang, M. Ouyang, Z. Wang, R. Liang, and X. Zhou, "The power load's signal analysis and short-term prediction based on wavelet decomposition," *Cluster Computing*, vol. 22, no. S5, pp. 11129–11141, 2017.
- [30] W. Li, C. Quan, X. Wang, and S. Zhang, "Short-term power load forecasting based on a combination of VMD and ELM," *Polish Journal of Environmental Studies*, vol. 27, no. 5, pp. 2143–2154, 2018.
- [31] S. Guo, L. Ruan, H. Dong et al., "EMD-based short-term load forecasting," in *Proceedings of the 2014 International Conference on Automatic Control Theory & Application*, pp. 141–145, Bangkok, Thailand, 2014.
- [32] X. Jin, N. Yang, X. Wang, Y. Bai, T. Su, and J. Kong, "Hybrid deep learning predictor for smart agriculture sensing based on empirical mode decomposition and gated recurrent unit group model," *Sensors*, vol. 20, no. 5, p. 1334, 2020.
- [33] X. Jin, N. Yang, X. Wang, Y. Bai, T. Su, and J. Kong, "Deep hybrid model based on EMD with classification by frequency characteristics for long-term air quality prediction," *Mathematics*, vol. 8, no. 2, p. 214, 2020.
- [34] Z. Yan, D. Li, L. Yao, and H. Xue, "Short-term power load forecasting based on improved T-S fuzzy-neural network," in *Proceedings of the 2016 World Congress on Intelligent Control and Automation (WCICA)*, pp. 109–113, Guilin, China, 2016.
- [35] C. Li, S. Li, and Y. Liu, "A least squares support vector machine model optimized by moth-flame optimization algorithm for annual power load forecasting," *Applied Intelligence*, vol. 45, no. 4, pp. 1166–1178, 2016.
- [36] Z. Liu, X. Sun, S. Wang, M. Pan, Y. Zhang, and Z. Ji, "Midterm power load forecasting model based on kernel principal component analysis and back propagation neural network with particle swarm optimization," *Big Data*, vol. 7, no. 2, pp. 130–138, 2019.
- [37] H. Li, S. Guo, C. Li, and J. Sun, "A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm," *Knowledge-Based Systems*, vol. 37, pp. 378–387, 2013.
- [38] F. He, J. Zhou, L. Mo et al., "Day-ahead short-term load probability density forecasting method with a decomposition-based quantile regression forest," *Applied Energy*, vol. 262, Article ID 114396, 2020.
- [39] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm

- configuration,” *Lecture Notes in Computer Science*, vol. 6683, pp. 507–523, Springer, Berlin, Germany, 2011.
- [40] J. Zhang and X. Xiao, “Predicting chaotic time series using recurrent neural network,” *Chinese Physics Letters*, vol. 17, no. 2, pp. 88–90, 2000.
- [41] F. Gers, N. N. Schraudolph, and J. Schmidhuber, “Learning precise timing with LSTM recurrent networks,” *Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2002.
- [42] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu, “Deep learning with gated recurrent unit networks for financial sequence predictions,” *Procedia Computer Science*, vol. 131, pp. 895–903, 2018.
- [43] Y. Huo, Y. Yan, D. Du et al., “Long-term span traffic prediction model based on STL decomposition and LSTM,” in *Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–4, Matsue, Japan, 2019.
- [44] S. Mouatadid, J. F. Adamowski, M. K. Tiwari, and J. M. Quilty, “Coupling the maximum overlap discrete wavelet transform and long short-term memory networks for irrigation flow forecasting,” *Agricultural Water Management*, vol. 219, pp. 72–85, 2019.
- [45] F. Ding, L. Lv, J. Pan, X. Wan, and X.-B. Jin, “Two-stage gradient-based iterative estimation methods for controlled autoregressive systems using the measurement data,” *International Journal of Control, Automation and Systems*, vol. 18, no. 4, pp. 886–896, 2020.
- [46] X. Zhang and F. Ding, “Recursive parameter estimation and its convergence for bilinear systems,” *IET Control Theory & Applications*, vol. 14, no. 5, pp. 677–688, 2020.
- [47] F. Ding, L. Xu, D. Meng, X.-B. Jin, A. Alsaedi, and T. Hayat, “Gradient estimation algorithms for the parameter identification of bilinear systems using the auxiliary model,” *Journal of Computational and Applied Mathematics*, vol. 369, Article ID 112575, 2020.
- [48] H. Ma, J. Pan, F. Ding, L. Xu, and W. Ding, “Partially-coupled least squares based iterative parameter estimation for multi-variable output-error-like autoregressive moving average systems,” *IET Control Theory & Applications*, vol. 13, no. 18, pp. 3040–3051, 2019.
- [49] F. Ding, X. Zhang, and L. Xu, “The innovation algorithms for multivariable state-space models,” *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 11, pp. 1601–1608, 2019.
- [50] X. Zhang and F. Ding, “Adaptive parameter estimation for a general dynamical system with unknown states,” *International Journal of Robust and Nonlinear Control*, vol. 30, no. 4, pp. 1351–1372, 2020.
- [51] X. Zhang, F. Ding, and L. Xu, “Recursive parameter estimation methods and convergence analysis for a special class of nonlinear systems,” *International Journal of Robust and Nonlinear Control*, vol. 30, no. 4, pp. 1373–1393, 2020.
- [52] L. Xu, L. Chen, and W. Xiong, “Parameter estimation and controller design for dynamic systems from the step responses based on the Newton iteration,” *Nonlinear Dynamics*, vol. 79, no. 3, pp. 2155–2163, 2015.
- [53] L. Xu, “The damping iterative parameter identification method for dynamical systems based on the sine signal measurement,” *Signal Processing*, vol. 120, pp. 660–667, 2016.
- [54] L. Xu, “The parameter estimation algorithms based on the dynamical response measurement data,” *Advances in Mechanical Engineering*, vol. 9, no. 11, pp. 1–12, 2017.
- [55] F. Ding, Y. Liu, and B. Bao, “Gradient-based and least-squares-based iterative estimation algorithms for multi-input multi-output systems,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 226, no. 1, pp. 43–55, 2012.
- [56] L. Xu and F. Ding, “Iterative parameter estimation for signal models based on measured data,” *Circuits, Systems, and Signal Processing*, vol. 37, no. 7, pp. 3046–3069, 2018.
- [57] L. Xu and F. Ding, “Parameter estimation for control systems based on impulse responses,” *International Journal of Control, Automation and Systems*, vol. 15, no. 6, pp. 2471–2479, 2017.
- [58] X. Zhang and F. Ding, “Hierarchical parameter and state estimation for bilinear systems,” *International Journal of Systems Science*, vol. 51, no. 2, pp. 275–290, 2020.
- [59] F. Ding, G. Liu, and X. P. Liu, “Parameter estimation with scarce measurements,” *Automatica*, vol. 47, no. 8, pp. 1646–1655, 2011.
- [60] X. Zhang, Q. Liu, F. Ding, A. Alsaedi, and T. Hayat, “Recursive identification of bilinear time-delay systems through the redundant rule,” *Journal of the Franklin Institute*, vol. 357, no. 1, pp. 726–747, 2020.
- [61] X. Zhang, F. Ding, L. Xu, and E. Yang, “Highly computationally efficient state filter based on the delta operator,” *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 6, pp. 875–889, 2019.
- [62] Y. Wang and F. Ding, “Novel data filtering based parameter identification for multiple-input multiple-output systems using the auxiliary model,” *Automatica*, vol. 71, pp. 308–313, 2016.
- [63] X. Zhang, F. Ding, and E. Yang, “State estimation for bilinear systems through minimizing the covariance matrix of the state estimation errors,” *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 7, pp. 1157–1173, 2019.
- [64] F. Ding, L. Qiu, and T. Chen, “Reconstruction of continuous-time systems from their non-uniformly sampled discrete-time systems,” *Automatica*, vol. 45, no. 2, pp. 324–332, 2009.
- [65] Y. Bai, X. Wang, X. Jin, Z. Zhao, and B. Zhang, “A neuron-based Kalman filter with nonlinear autoregressive model,” *Sensors*, vol. 20, no. 1, p. 299, 2020.
- [66] Y. Bai, X. Wang, X. Jin, T. Su, J. Kong, and B. Zhang, “Adaptive filtering for MEMS gyroscope with dynamic noise model,” *ISA Transactions*, vol. 101, pp. 430–441, 2020.
- [67] X. Wang, Y. Bai, Y. Yang, J.-b. Yu, Z.-y. Zhao, and X.-b. Jin, “Fuzzy boost classifier of decision experts for multicriteria group decision-making,” *Complexity*, vol. 2020, Article ID 9107167, 10 pages, 2020.
- [68] R. Wang, Y. Li, H. Sun, and Z. Chen, “Integrated cabin contaminant monitoring network based on Kalman consensus filter,” *ISA Transactions*, vol. 71, pp. 112–120, 2017.
- [69] Y. Liu, F. Ding, and Y. Shi, “An efficient hierarchical identification method for general dual-rate sampled-data systems,” *Automatica*, vol. 50, no. 3, pp. 962–970, 2014.
- [70] J. Ding, F. Ding, X. P. Liu, and G. Liu, “Hierarchical least squares identification for linear SISO systems with dual-rate sampled-data,” *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2677–2683, 2011.
- [71] F. Ding, G. Liu, and X. Liu, “Partially coupled stochastic gradient identification methods for non-uniformly sampled systems,” *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1976–1981, 2010.
- [72] L. Xu and F. Ding, “Parameter estimation algorithms for dynamical response signals based on the multi-innovation theory and the hierarchical principle,” *IET Signal Processing*, vol. 11, no. 2, pp. 228–237, 2017.
- [73] L. Xu, “Recursive least squares and multi-innovation stochastic gradient parameter estimation methods for signal

- modeling,” *Circuits Systems and Signal Processing*, vol. 36, no. 4, pp. 735–1753, 2017.
- [74] L. Xu, F. Ding, Y. Gu, A. Alsaedi, and T. Hayat, “A multi-innovation state and parameter estimation algorithm for a state space system with d-step state-delay,” *Signal Processing*, vol. 140, pp. 97–103, 2017.
- [75] L. Xu, W. Xiong, A. Alsaedi, and T. Hayat, “Hierarchical parameter estimation for the frequency response based on the dynamical window data,” *International Journal of Control, Automation and Systems*, vol. 16, no. 4, pp. 1756–1764, 2018.
- [76] L. Xu, F. Ding, and Q. Zhu, “Hierarchical Newton and least squares iterative estimation algorithm for dynamic systems by transfer functions based on the impulse responses,” *International Journal of Systems Science*, vol. 50, no. 1, pp. 141–151, 2019.