IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

IEEE POWER & ENERGY SOCIETY SECTION

Received 4 May 2024, accepted 5 July 2024, date of publication 16 July 2024, date of current version 24 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3428540

RESEARCH ARTICLE

GRO-Bagging Day-Ahead Power Curve Forecasting Model Based on Multi-Cycle Feature Extraction

YAOXIAN LIU[1], KAIXIN ZHANG[1], SONGSONG CHEN[2], YING ZHOU[2], AND JINGWEN CHEN[1]

[1]School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China

[2]Beijing Key Laboratory of Demand-Side Multi-Energy Complementary Optimization and Supply-Demand Interaction Technology, China Electric Power Research Institute Company Ltd., Haidian, Beijing 100192, China

Corresponding author: Kaixin Zhang (220612082@sust.edu.cn)

This work was supported in part by the State Grid Corporation Headquarters Science and Technology Project under Grant 5400-202321572A-3-2-ZN.

ABSTRACT Power loads usually have multiple cycles, and the traditional forecasting methods only use the historical loads under various types of cycles together as input features to construct the forecasting model, ignoring the deep features under multiple cycles. The simultaneous inputs will lead to the cycles overlapping and influencing each other, which will be difficult to deal with when building the model. Therefore, this paper proposes a GRO-Bagging day-ahead power curve forecasting model based on multi-cycle feature extraction. First, the multi-periodicity of the power load is analyzed. The one-dimensional time series is converted to two-dimensional according to the multiple cycles of power load. Then, the multi-periodic feature extraction is performed by a multi-size convolutional feature extraction layer with parallel selected based on the data characteristics and modeling mechanism, and Bootstrap Aggregating (Bagging) method is used to construct different prediction models for the datasets containing different periodic features; finally, Gold Rush Optimization (GRO) algorithm is introduced and improved by using the Tent chaotic mapping and elite strategy, and the improved optimization algorithm is used for the weight optimization of the model, the error feedback mechanism is introduced to achieve the weight dynamic Updating. To prove the superiority of the proposed model, a series of comparison experiments and ablation experiments are carried out on real datasets, and the results show that the proposed method has higher prediction accuracy, and prove that the multi-period feature extraction and dynamic weighting methods have a positive and active effect on load prediction.

INDEX TERMS Bagging, integrated learning, LSTM, multi-periodic features, short-term load forecasting, gold rush optimization algorithm, XGBoost.

I. INTRODUCTION

Short-Term Load Forecasting (STLF) refers to the prediction of electricity loads for a short time in the future, usually hours to days [1]. This work is important for electricity market operations, generation planning, reliability analysis, and security assessment [2]. Accurate short-term load forecasts can help power plants, transmission systems, and distribution networks make timely adjustments to meet future electricity demand and avoid problems associated with supply-demand imbalances, such as overloads, wasted energy, or blackouts [3].

However, the combined effects of economic factors, meteorological factors, social activities, and other factors lead

The associate editor coordinating the review of this manuscript and approving it for publication was Maria Carmen Falvo.

to the power load showing obvious multi-periodic, non-linear, and time-varying characteristics [4]. Therefore, it is of great significance to realize accurate short-term load forecasting by exploring power load characteristics in depth. In short-term load forecasting, the common forecasting methods are mainly classical statistical, machine learning, and deep learning. Traditional mathematical statistical methods include the Autoregressive Model (AR) [5], Autoregressive Moving Average Model (RAMA) [6], and Autoregressive Integrated Moving Average Model (ARIMA) [7], etc. Many of these methods are based on the assumption of linearity, where the relationship between load and time or other factors is assumed to be linear, however, in real situations, loads usually exhibit non-linear characteristics.

Machine learning methods are mainly Decision Trees, Random Forest (RF) [8], and Gradient Boosting Trees (GBDT) [9]. It is mainly used to construct regression models through various influencing factors to realize regression prediction, however, machine learning models can easily overfit the training data, especially when the amount of data is insufficient or the model complexity is too high.

With the continuous development of artificial intelligence, some deep learning methods are widely used, such as Recurrent neural network (RNN) [10], Long Short-Term Memory (LSTM) [11], and Gated Recurrent Unit (GRU) [12]. Compared to machine learning, deep learning models are better able to extract complex nonlinear features from data and can better utilize historical load data, weather data, holidays, and other factors to improve forecast accuracy.

Existing studies have shown that the actual electricity load usually exhibits multi-period characteristics, such as daily and annual variations in weather, and weekly and quarterly variations in electricity consumption [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. Conventional forecasting models are unable to reflect the different cyclical characteristics of loads, which may also affect the accuracy of forecasting results.

Therefore, [13] proposed a short-term load forecasting model with a multi-temporal scale spatiotemporal convolutional network by taking the past load data as the characteristics and considering the time series characteristics of the load data. The forecasting model was trained with historical characteristic loads of 7, 21, 99, and 199 days, which proved that the introduction of the historical cycle loads could improve the accuracy of the short-term forecasts. Reference [14] generates multiple time series by combining macro and micro information from continuous and discrete time series, including four different types of periodic series, which are inputted into the RNN model for prediction, and the results can effectively improve the accuracy. Reference [15] proposes a multi-period prediction model for household load forecasting, which better learns the electricity consumption patterns of users by learning the characteristics and contextual information of the relevant load curves in multiple historical cycles, and also further proves that the load has obvious multi-periodicity and time series. All of the above studies have considered the cycle features, but all of them only take the historical load under the cycle as an input feature, have not carried out in-depth mining, and have not considered the problem that different cycles will affect each other superimposed when multiple cycle features are input at the same time.

Due to the insufficient generalization of a single model, most of the current models are enhanced by integrating multiple models to enhance the model prediction capability [16]. However, most of the existing integration strategies adopt the mean value calculation method and weight fixing mode, which do not realize the complementarity between algorithms, and the theoretical basis of this combination is weak [17], [18]. Reference [19] by using the dynamic error function and optimal weight optimization algorithm, in order to balance the contradiction between the speed and accuracy of dynamic adjustment, for different periods, choose different weight adjustment algorithms for enhancement and compare with the traditional fixed-weight combination prediction model. Reference [20] proposes an adaptive online learning technique based on the Hidden Markov Model line learning forecasting method, which recursively updates the model parameters, and can effectively learn the dynamic changes of the load, and then improve the forecasting accuracy, proving that the consideration of the recent data in the model updating can effectively improve the accuracy.

Aiming at the shortcomings of the above research, this paper further extracts different cyclic features of power loads, extracts different cyclic information through integration learning and adopts an intelligent optimization algorithm for the weight calculation of the integrated model. A GRO-Bagging day-ahead power curve forecasting model based on multi-cycle feature extraction is proposed to realize load periodicity feature extraction and learning, which effectively improves the accuracy and generalization of the forecasting model. Firstly, a one-dimensional time series is converted into two dimensions according to the multiple periodicity of power loads, an Inception-CNN layer is introduced according to the characteristics of convolution operation, the historical data between adjacent time points and adjacent cycles are combined and extracted to form new features, and multiple subdatasets containing features of different cycles are established to realize the decoupling of the temporal pattern of the loads; then, two basic models, LSTM and XGBoost two basic models, using the Bagging method to construct different prediction models for datasets containing different cycle features; finally, the Gold Rush Optimization Algorithm is improved for the integrated model weight optimization problem, and the Tent chaotic mapping and elite strategy are introduced into the Gold Rush Optimization Algorithm and used to solve the integrated model weights, to construct the GRO based on the multi-cycle feature extraction- Bagging day-ahead load forecasting model. A series of comparative experiments are conducted on real data, and the experiments show that the method of this paper can effectively improve the prediction accuracy. The main contributions of this study are as follows:

1. A multi-cycle feature extraction method is proposed to further extract the different cycle features of power loads and realize the decoupling of load features under the influence of multi-cycle.

2. Bagging integrated learning strategy is used to integrate the prediction model considering different cycle features, chaotic mapping, and elite strategy are introduced to improve the Gold Rush Optimization Algorithm, and the improved optimization algorithm is used for weight solving, which enhances the model weight solving effect and improves the prediction accuracy.

3. A weight updating strategy for the integrated load forecasting model is proposed, which realizes dynamic updating of the weights by setting a history review window for prediction error feedback, making the model better adaptable to future data.

The remainder of the paper is organized as follows: section II describes the multicycle feature extraction method, the improved optimization algorithm, and the integrated model architecture. Data presentation and performance evaluation of the associated prediction results are carried out in Section III. Finally, it is summarized in Section IV.

II. THE PROPOSED FRAMEWORK

In this section, the proposed predictive model is described in detail, including its main structure and properties. The model is realized by extracting cycle features, building prediction models considering different cycles, and finally building a Bagging integrated model. The framework of the proposed approach is shown in Fig.1, and its methodology consists of the following steps.

1) Step 1: to analyze the correlation of historical loads under multiple cycles, convert a one-dimensional time series to a two-dimensional according to the multiple periodicities of power loads, and introduce an Inception-CNN [21] layer to combine and extract new features from historical data between adjacent time points and adjacent cycles according to the characteristics of the convolutional operation, and combine the new cycle features with other exogenous variables to form multiple sub-datasets containing different cycle features, and realize the decoupling of loads in time patterns. See II-A and II-B for specific steps.
2) Step 2: Adopting the idea of Bagging integrated learning, two base models, LSTM and XGBoost, are selected to build prediction models for datasets containing different cycles. See II-D for specific steps.
3) Step 3: For the integrated algorithm weight optimization problem, the Gold Rush Optimization Algorithm GRO is introduced, and the Gold Rush Optimization Algorithm GRO algorithm is improved by introducing the Tent Chaos Mapping Algorithm and the Elite Strategy, to make it faster convergence speed and better solutions in the weight solving task. See II-C for specific steps.
4) Step 4: The Bagging integration algorithm uses the improved Gold Rush optimization algorithm to optimize the model weights, and through the setting of the history review window, the prediction error feedback is carried out, thus realizing the updating of the weights of the integrated model, which ensures the generalization of the model, and at the same time, makes the model more in line with the characteristics of the recent trend of the data. See II-E and II-F for specific steps.
5) Step 5: Validating various aspects of the proposed model, this paper sets up a series of ablation experiments and compares them with current mainstream prediction models. Evaluate the validity of the proposed prediction model through several commonly used performance metrics including Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE). See III for specific steps.

A. MULTI-CYCLE ANALYSIS

Maximum Information Coefficient [22] (MIC) is a statistical method for calculating the strength of the relationship between two variables proposed by Reshef et al. in 2011 and is considered a more flexible and efficient method than the traditional correlation coefficient. MIC measures the relationship between two variables by calculating the maximum possible joint distribution of their mutual information to measure the relationship between them. It is capable of capturing all types of relationships, not just linear ones. The calculated MIC has a value between 0 and 1, which indicates the degree of correlation between the two variables, with 1 indicating perfect correlation and 0 indicating no correlation, and the closer to 1 the higher the correlation.

Existing studies have shown that power load has obvious daily, weekly, monthly, seasonal, and annual periodicity. The US load data set used in this paper has 10 consecutive years of power load data with a sampling interval of 1 hour. The MIC correlation analysis was carried out between the time load value of each day for 30 consecutive days and the historical load at the same time under the daily, weekly, monthly, seasonal, and annual cycles.

As shown in Fig. 2 for the MIC calculation results of the historical cycle length from 1 to 7, the overall MIC is between 0.5 and 0.7, the MIC under the seasonal cycle shows cycle fluctuation in line with the phenomenon of alternating seasons of spring, summer, fall and winter, the MIC under the weekly cycle increases with the length of the cycle, and the MIC under the daily, monthly and yearly cycles shows decreasing with the increase of the cycle length, and the MIC results show strong correlation between the historical load data of the same time under the different cycles and the current load. From the MIC results, there is a strong correlation between the historical load data and the current load at the same time in different cycles.

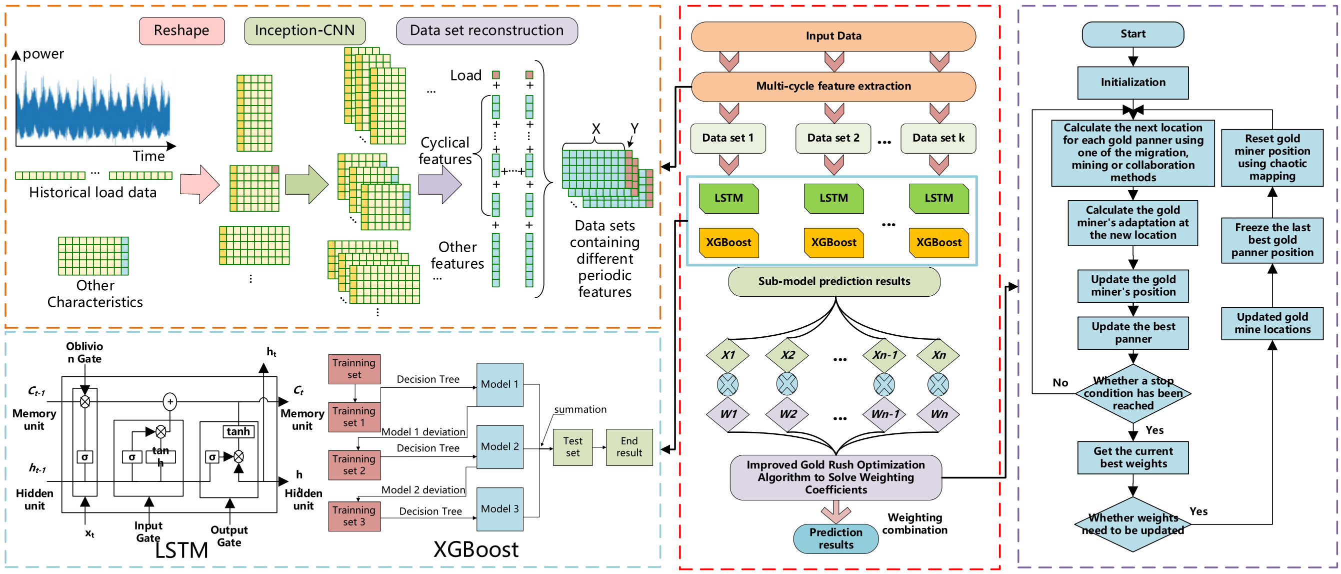

FIGURE 1. Schematic diagram of GRO-Bagging day-ahead power curve forecasting model based on multi-cycle feature extraction.

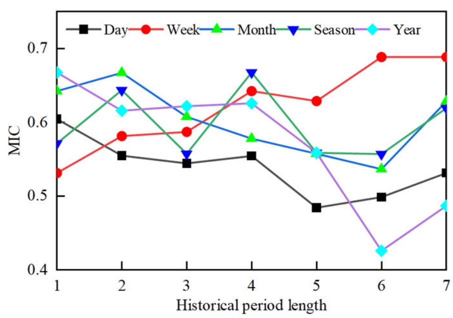

FIGURE 2. Correlation analysis of MIC of multi-cycle historical power load.

B. MULTI-CYCLE FEATURE EXTRACTION ALGORITHM

In this paper, based on the idea of Inception structure in GoogleNet [21], a mechanism multi-size feature convolutional extraction layer with parallel processing (referred to as Inception-CNN layer) is introduced for multi-cycle feature extraction of power loads.

The original one-dimensional load time series X^{1D} is first converted into a two-dimensional load time series X_i^{2D} according to the cycle length $\{t_1, \cdots, t_n\}$ of n types of cycles such as daily, weekly, monthly, seasonal, annual, etc., in order to prepare for the two-dimensional processing in the next step. The specific formula is as follows:

$$X_i^{2D} = \text{Reshape}_{|len(X^{1D})/t_i|, t_i}(X^{1D}), \; i \in \{1, \cdots, n\} \tag{1}$$

where reshape denotes a dimensional change, a set of two-dimensional tensor X_i^{2D} obtained by deforming the original data, which represents i derived from different cycles of two-dimensional time series, for each two-dimensional time series through the Padding layer will be extended by the data, and then through a convolutional layer with a multi-head structure, the multi-head structure through the introduction of a parallel processing mechanism in the same layer to extract different features, and then a layer of Inception-CNN layer contains different sizes and dimensions. That is, one Inception-CNN layer contains different sizes and dimensions of convolutional layers and pooling layers, etc. Each 2D tensor is derived into multiple 2D tensors, and new features are formed by combining and extracting historical data between neighboring time points and neighboring cycles through different sizes of convolution kernels and pooling kernels according to the characteristics of the convolution operation. The specific formula is as follows:

$$\hat{X}_{i,j}^{2D} = \text{Inception}\left(\text{Padding}\left(X_i^{2D}\right)\right), j \in \{1, \cdots, r\} \tag{2}$$

where $\hat{X}_{i,j}^{2D}$ denotes the j-th 2D feature matrix extracted under the t_i-cycle feature, Inception$(\cdot)$ denotes feature extraction through one Inception-CNN layer, Padding$(\cdot)$ denotes the data padding operation, j denotes the number of 2D feature matrices generated, and r denotes the number of convolutional and pooling layers included in the Inception-CNN layer Sum.

In order to maintain the original 2D tensor shape and position after the convolution and pooling operations, while not discarding the original information, this paper chooses to use the historical data to perform adaptive Padding according to the size of the convolution kernel and the pooling kernel, i.e., complete the filling of the columns in accordance with the sequential nature of the time series, and then complete the filling of the rows according to the periodicity of the time series, and the number of padding entries required to maintain the shape of inputs The number of fill entries needed to maintain the shape of the input is always equal to kernel_size minus 1, where kernel_size is the size of the convolution or pooling kernel, and the two-dimensional time series data is adaptively expanded according to the time dimension. Taking the 3×3 convolutional kernel as an example, it is expanded as shown in Fig. 3, filling the data according to temporality and period on the left side and the bottom side, respectively, and the numbers in the figure represent the time points.

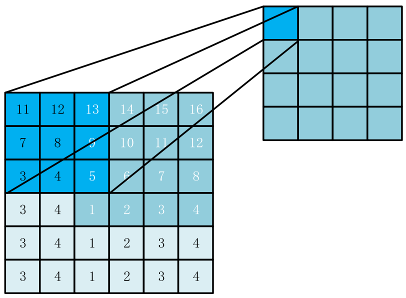

FIGURE 3. Adaptive Fill Schematic.

In this paper, different convolution kernel parameters are used to achieve the calculation of first-order difference during the cycle and first-order difference during the cycle, as shown in equation (3) respectively. Average pooling, maximum pooling, and minimum pooling operations are performed between weeks and cycles through 2×1 and 1×2 pooling Windows, respectively. Together with extracting the history load using a 1×1 convolution kernel, a total of 9 features are extracted per cycle.

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 1 \end{bmatrix} \tag{3}$$

Existing studies have shown that time-derived features as well as weather features have a significant impact on electricity load, therefore, we also introduce peripheral features, including temperature, year, month, day, week, weekday, weekend, and moment of the moment to be predicted in a total of 8 dimensions, which are encoded as continuous features according to the temporal order, and then combined together with the first m features extracted by convolution and pooling under different 2D vectors, which are jointly used as the input feature quantities of the moment to be predicted, and the load of the moment to be predicted is used as the output to generate the n-class dataset with different input features.

C. GOLD RUSH OPTIMIZATION ALGORITHM IMPROVEMENT

The Gold Rush Optimizer algorithm [23] (GRO) is a meta-heuristic algorithm proposed in 2023 that is inspired by the process of panning for gold during the gold rush. The algorithm simulates the way gold miners search for gold on the riverbed, and find the optimal solution by continuously trying different locations and methods. The algorithm has high convergence speed and search efficiency.

In this paper, we make relevant improvements to the GRO algorithm for the weight optimization problem of integrated models.

1) TENT CHAOS MAPPING

Tent chaotic mapping is commonly used to study nonlinear behavior in dynamical systems, its name comes from the fact that its folded shape is similar to the top of a tent (Tent), Tent chaotic mapping can exhibit a uniform distribution under appropriate parameter settings. Since the initialization of the gold miner's position using a random method will lead to an uneven distribution of the initial position. Therefore, in this paper, Tent chaotic mapping is used to initialize the population to ensure the diversity and balance of the initialization of the gold miners' positions.

The Tent chaotic mapping formula is as follows:

$$x_{n+1} = f(x_n) = \begin{cases} x_n/\alpha, x_n \in [0, \alpha) \\ (1 - x_n)/(1 - \alpha), x_n \in [\alpha, 1] \end{cases} \tag{4}$$

where x_n denotes the n iteration chaotic value, x_{n+1} denotes the $n+1$ iteration chaotic value, and α is a control parameter.

The behavior of the Tent mapping depends on the choice of the parameter A. The system's behavior becomes more linear when A increases. When A is smaller, the behavior of the system tends to be more linear, while when A increases, the nonlinear characteristics of the system become more obvious and may exhibit chaotic behavior. Since the conventional Tent chaotic mapping produces chaotic values between 0 and 1, in this paper, a normalization step is added after the Tent chaotic mapping to scale it between -1 and 1. The normalization formula is shown in equation (5):

$$X_{Sta} = 2\,(X - 0.5) \tag{5}$$

where X denotes the generated chaotic value and X_{Sta} denotes the value after normalization is performed.

Through many experiments, in this paper, α is set to 0.499, and the distribution of its Tent mapping is shown in Fig. 4, from which it can be seen that the generated chaotic values are distributed between -1 and 1, and are uniformly distributed.

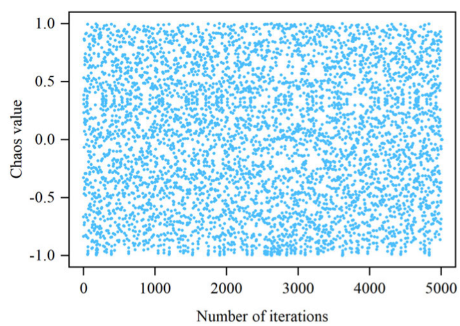

FIGURE 4. Tent chaotic map distribution.

2) ELITE STRATEGY

Due to the regularity of the electricity load, keeping the last model weights can help to speed up the updating speed of the model weights. Therefore, when initializing the gold miner's position each time, freezing the position of the last best gold miner, so that it has the historical experience when calculating the weights next time can accelerate the speed of optimization.

3) WEIGHT OPTIMIZATION PROCESS

In this paper, the improved gold panning optimization algorithm is used for integrated model weight optimization, and Fig. 5 shows the specific flow of the improved gold panning optimization algorithm.

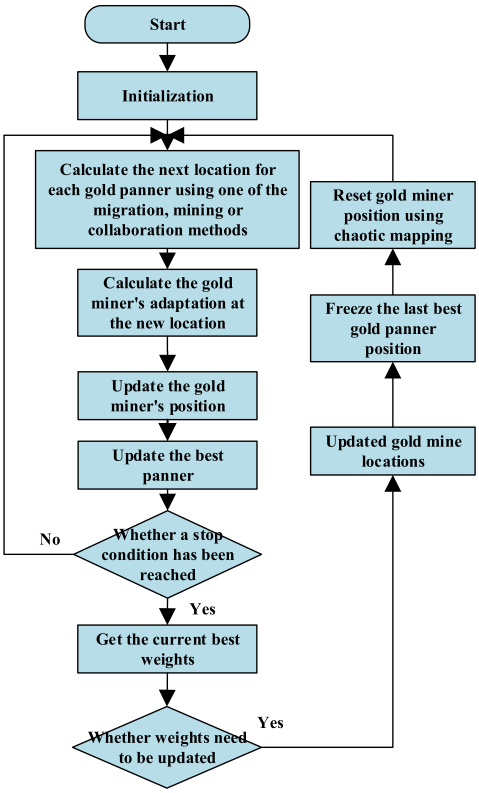

FIGURE 5. Improved gold optimization algorithm flow chart.

First, the initial parameters of the optimization algorithm are set, including the number of gold panners, the maximum number of iterations, the search space, etc., and then the gold panner positions and evaluation matrix are initialized, and the Tent chaotic mapping is used to initialize the gold panner positions as the integrated model weights.

The next position is calculated using migration, gold panning, or collaboration for each gold panner:

(1) Migration process

When a gold mine is found, the gold miner migrates there to pan for gold, and the location of the richest gold mine is the optimal solution in the search space. Since its exact location is unknown, the location of the optimal gold panner each time is considered as the location of the richest gold mine

(2) Gold panning process

The goal of each gold panner is to find more gold in the gold panning area. The location of each panner is considered as the approximate location of the gold mine.

(3) Collaborative Process

As gold panning is sometimes carried out through teamwork, two gold panners are randomly selected. Three-player collaboration is realized between the current computational gold panner and the two randomly selected gold panners.

The gold panners keep moving in order to get more gold, and in order to decide whether the panners should stay where they are or move to a new location, an evaluation function is used to compare the two. If the value of the objective function improves, the gold miner's position is updated; otherwise, the gold miner stays at the previous position.

When the stopping condition is reached, the position of the best gold miner is obtained as the best weight, and when the weight update is performed again, an elite strategy is used to retain the best weight, i.e., one best gold miner's position in the last gold mine is retained, and the rest of the gold miners are used to generate a set of mixed-pure sequences using Tent chaotic mapping method as the new position and a new iteration is performed again.

D. BAGGING

The basic idea of Bootstrap Aggregating (Bagging) [24] is to abstract n times from the training dataset with returns to form n sub-datasets, train a base classifier for each sub-dataset, and finally synthesize their results to obtain the final output. The advantage of Bagging is that by integrating multiple models, it can reduce the risk of overfitting and improve the robustness and generalization performance of the model, and it generally uses a simple voting or averaging method.

Considering the multi-period characteristics of power loads, multiple forecasting models are utilized to learn for different period characteristics, and each model is only responsible for learning one type of period characteristics so that the models have stronger learning and characterization capabilities. At the same time, due to the modeling mechanism and starting point of different forecasting models, the effective information that can be extracted by using different forecasting methods is also different.

In this paper, after comprehensively considering the characteristics of the data and the modeling mechanism of the model, LSTM and XGBoost are selected as the base models.

1) LSTM

Among them, LSTM, as a deep learning model, has a powerful nonlinear modeling capability to learn complex patterns in data, including nonlinear relationships. Through gating mechanisms and memory units, LSTM can automatically choose what information to keep and what to forget during training, thus helping to reduce the risk of overfitting [27].

As shown in Fig. 6, the basic unit of the LSTM network contains forgetting gate, input gate, and output gate. Input x_t in the forgetting gate together with state memory cell S_{t-1} and

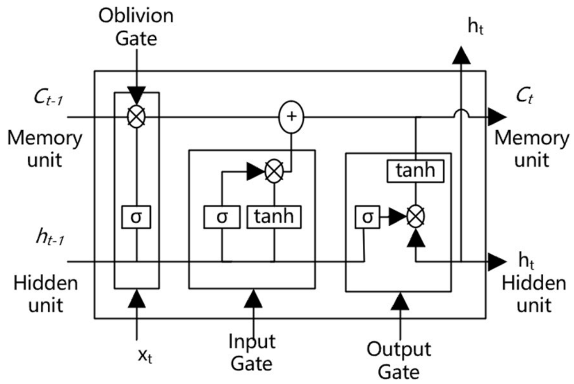

FIGURE 6. **LSTM network structure.**

intermediate output h_{t-1} determine the forgotten part of the state memory cell. The input x_t in the input gate determines the retention vector in the state memory cell after the change of sigmoid and tanh functions, respectively. The intermediate output h_t is jointly determined by the updated S_t and output o_t. The formula is shown in the following equation.

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \quad (6)$$

$$i_t = \sigma(W_{\mathrm{ix}}x_t + W_{\mathrm{ih}}h_{t-1} + b_{\mathrm{i}}) \quad (7)$$

$$g_t = \phi(W_{\mathrm{gx}}x_t + W_{\mathrm{gh}}h_{t-1} + b_{\mathrm{g}}) \quad (8)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \quad (9)$$

$$S_t = g_t \odot i_t + S_{t-1} \odot f_t \quad (10)$$

$$h_t = \phi(S_t)o_t \quad (11)$$

where: f_t, i_t, g_t, o_t, h_t and S_t are the states of oblivion gates, input gates, input nodes, output gates, intermediate outputs and state units, respectively; W_{fx}, W_{rh}, W_{ix}, W_{ih}, W_{gx}, W_{gh}, W_{ox} and W_{oh} are the matrix weights of the corresponding gates multiplied by the inputs x_t and the intermediate outputs h_{t-1}, respectively; b_f, b_i, b_{g}, b_o are the bias entries of the corresponding gates, respectively; $\odot$ denotes the multiplication of the elements of the vectors by the bits; σ denotes the variation of the sigmoid function; and ϕ denotes the variation of the tanh function.

2) XGBOOST

XGBoost, on the other hand, uses a combination of multiple trees to make predictions and has strong generalization capabilities [25]. It is suitable for dealing with structured data and can handle various types of features. With adequate feature engineering, XGBoost usually gives better prediction results. It integrates multiple decision trees through the gradient boosting framework, which is able to capture the complex relationships in the data, thus improving the generalization ability of the model. Its objective function consists of a loss function and a regular term, which limits the complexity of the model through the regular term and can effectively avoid the problem of low model generalization ability. The core idea of the algorithm is as follows:

As shown in Fig. 7, the tree is continuously added and grown by feature segmentation, and each added tree is equivalent to learning a new function $f(x)$ to fit the residuals of the last prediction.

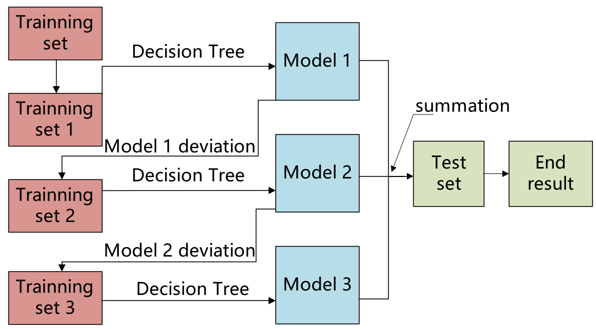

FIGURE 7. **XGBoost principle schematic.**

When k trees are obtained at the end of training, depending on the features of the samples, in each tree the samples fall to the corresponding leaf nodes, each of which corresponds to a score.

Finally, the scores corresponding to each tree are summed to obtain the final prediction for that sample.

E. GRO-BAGGING PREDICTION ALGORITHM BASED ON MULTI-CYCLE FEATURE EXTRACTION

As shown in Fig. 8 for the prediction algorithm flow proposed in this paper, firstly, the original data is subjected to multi-cycle feature extraction to form datasets with different cycle features, and then all the datasets are trained and predicted by using LSTM and XGBoost models to generate the results of multiple prediction sub-models, and then finally the final prediction results are obtained by using the improved GRO algorithm to weight the prediction results and combine them to obtain final prediction results.

For the constructed combined prediction model, the mathematical expression is shown below:

$$\tilde{y}_t = \sum_{i=1}^{N} w_{it}\tilde{y}_{it} \quad (12)$$

where $\tilde{y}_t$ denotes the integrated prediction result at moment t, N denotes the number of predictive sub-models, w_{it} is the weight of the ith model at moment t, and $\tilde{y}_{it}$ is the prediction value of the ith model at moment t.

The problem of optimizing the weights with the objective of minimizing the root-mean-square error, whose model weight coefficients are solved, can be transformed into solving the optimal solution of the following objective function:

$$obf\ (x) = \min\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \tilde{x}_i)^2}\right) \quad (13)$$

where $\tilde{x}_i$ is the i th model prediction value, x_i is the true value, and n is the number of sub-models. The improved gold rush

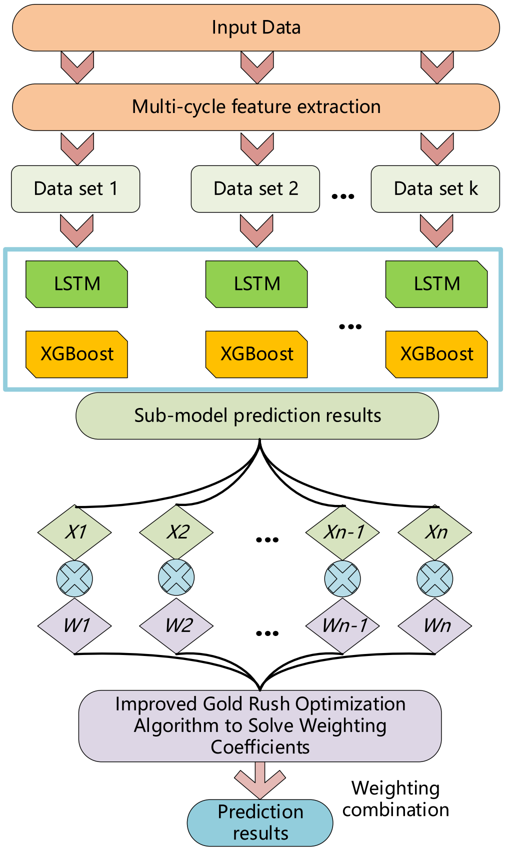

FIGURE 8. GRO-Bagging based on multi-period feature extraction.

optimization algorithm is used to solve for the integration weights, and the smaller the objective function value indicates the better performance of the combined prediction model.

Traditional combined prediction models usually limit the weight coefficients of individual models to the interval [0,1], i.e., it is assumed that the contribution of each individual model to the combined model is positive. However, there are some problems in practical applications. For example, in the short run, if the forecasts of multiple individual models that are combined are high, then the combined forecast must be higher than the true value. The forecasting direction of some individual models is opposite to the direction of the observed series, and if these models are given positive weights, it will lead to a large error in the combined result. Therefore, using the method of assigning negative weights to this model can effectively adjust the predictive trend of the combined model, thus improving the predictive effect of the model. In order to effectively improve the overall performance of the integrated model, we control the weight range between -1 and 1, i.e:

$$-1 \leq w_{it} \leq 1 \tag{14}$$

where w_{it} represents the weight of the i th model at time t.

F. GRO-BAGGING PREDICTION ALGORITHM BASED ON MULTI-CYCLE FEATURE EXTRACTION

Considering the temporal and cyclical nature of power loads, this paper further introduces a weight updating mechanism for the integrated model as shown in Fig. 9, i.e., the integrated model is always updated based on the average of the prediction errors of the historical K days for the integrated model weight w. Considering its cyclical nature, K is set as a multiple of the cycle. Taking the K+1th to 2Kth day weights as an example, the integrated model weights are updated by utilizing the error between the predicted and true values from day 1 to day K. Considering the prediction error of the historical K days can smooth the rate of weights updating and reduce the influence of abnormal samples, and at the same time, it prompts the integrated model to update the weights by considering the most recent data more often.

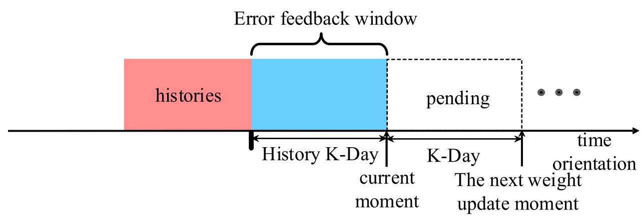

FIGURE 9. The weight update diagram based on error feedback.

III. CASE STUDY

A. SIMULATION SOFTWARE AND HARDWARE

The experiments were performed with a computer equipped with an AMD Ryzen5 5600 6-Core processor, an NVIDIA GeForce GTX 4070 graphics card, and 16.0 GB of memory (RAM). The software simulation was implemented via Python, and the main libraries used included Pytorch and Scikit-learn.

B. INTRODUCTION TO THE DATASET

The data used for the experiments are GEFCom2014-E, an extended version of the GEFCom2014-L dataset provided by the Global Electricity Energy Forecasting Competition [26], which contains hour-by-hour load data and temperature data for a region in the U.S. for the period of January 1, 2006, through December 31, 2014, with 24 load values per day. The data length allowed for simultaneous consideration of daily, weekly, monthly, seasonal, and annual cycles. The data of the first 3 years are not involved in the model construction because it is necessary to use 3 years ago loads as features. In this paper, some data are selected for experiments, and the selection interval is January 2010 to March 2011 for a total of 15 months of data.

C. PARAMETER SETTING

For all sub-models in the integrated model, all features with a historical cycle step of 3 after multi-period feature extraction, temperature, and date features at the moment to be predicted are selected as inputs to the prediction model. Considering the model variability, in which the LSTM model inputs features in the order of period, the historical time step is 3, then the input format is 3 × 17, and the load at the point of time to be predicted is used as the output of the prediction model. XGBoost model adopts a one-time input, and the input features are 33, and the load at the point of time to be predicted is used as the output of the prediction model. Since the minimum period considered in this paper is 1 day, day-ahead load forecasting can be realized. The individual hyperparameters of the model are determined by grid search, and the specific parameter Settings of the XGBoost, LSTM, and GRO algorithms are shown in Table 1. Some parameters not listed use default values.

TABLE 1. Specific parameters of the proposed model.

Methods	Hyper Parameter Setting
XGBoost	Tree depth 6, learning rate 0.2, number of trees 300 Subsample 0.8, min_child_weight 5
LSTM	Number of hidden layers 2, Number of neurons 64, 32, Learning rate 0.01, Optimizer Adam, Loss function MSE, Dropout 0.01, Activation function Relu
GRO	Number of gold diggers 50, Search space -1 to 1, Maximum number of iterations 200, Gold digger position dimension 10

For the sake of fairness in algorithm comparison, all regression models follow the conventional feature construction method, using the date feature of the moment to be predicted, the temperature feature, and the load at the same moment of the 7 days of history as input features to predict 24 points in the future. The time series forecasting model input multivariate historical series (including date features and temperature) were 7 × 24 in length, and direct multistep forecasting (24 steps) was performed. The data divisions are all divided into training, test, and validation sets in the ratio of 8:1:1. All hyperparameters of the model are determined by grid search, and the parameter combination with the smallest error is selected as the optimal parameter of the model by setting the search range and step size and traversing the experiments on the possible combinations.

For the prediction model of the neural network, MSE is used for the loss function, and in order to prevent overfitting, early stopping criterion and Dropout are used, in which the early stopping criterion is used by checking the validation loss at the end of each training cycle, and the count will be increased by 1 if does not reduce the error in the validation set, and the training will be stopped when the threshold is reached, and the last best model will be retained as the final model, and in this paper, the threshold for early stopping is set to 10. Dropout is a regularization technique that prevents overfitting by randomly dropping a certain percentage of neurons in the training process. In this paper, a dropout probability of 0.01 is finally chosen through multiple experiments to prevent overfitting and maintain the model learning ability. To prevent the gradient explosion problem, the Adam adaptive learning rate optimization algorithm is used. To prevent gradient vanishing, the Relu activation function and normalization are used before each layer.

As the order of magnitude of various types of input features may differ, it will affect the model training. In this paper, the input features are normalized and the original data are scaled to the range of [0,1] by linear transformation with the following formula:

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{15}$$

where X_{norm} indicates the scaled data, X indicates the original data to be scaled, X_{min} indicates the minimum value, and X_{max} indicates the maximum value.

D. EVALUATION INDICATORS

Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) were chosen as the evaluation indexes of the model [27].

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{16}$$

$$MAE = \frac{1}{n}\sum\nolimits_{i=1}^{n} \mid \hat{y}_i - y_i \mid \tag{17}$$

$$MAPE = \frac{100\%}{n}\sum\nolimits_{i=1}^{n} \mid \frac{\hat{y}_i - y_i}{y_i} \mid \tag{18}$$

In the above formula, y_i is the actual observed value, $\hat{y}_i$ is the predicted value of the model, and n is the number of samples.

E. VALIDATING MODEL FRAGMENTS

1) MULTI-CYCLE FEATURE EXTRACTION

In order to verify the positive effect of multi-period feature extraction on load forecasting, a single forecasting model is used to model datasets containing different feature periods. As shown in Fig. 10, all five datasets with different cycle features are modeled by the LSTM model, comparing the prediction errors of three consecutive days, it can be seen that there is a lower error in the weekly feature dataset on the first day, the second day has a lower error in the monthly feature dataset, and the third number of days has a lower error in the daily feature dataset, which shows that the datasets with different cycle features under the same model can lead to different forecasting results, proving that more effective information can be provided in load forecasting after multi-cycle feature extraction. Therefore coupling the results of multiple forecasting models, each focusing on learning one periodic feature, can enhance the learning and characterization ability of the integrated model.

2) MODEL INTEGRATION

In order to demonstrate that the effective information that can be extracted by using different prediction methods is also different, we chose the dataset after performing weekly cycle

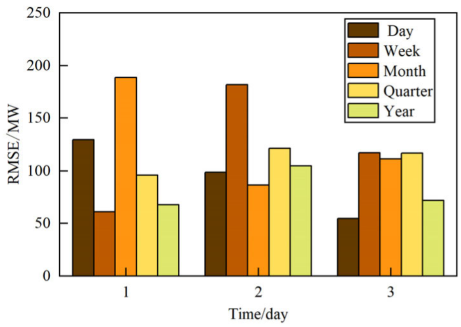

FIGURE 10. Prediction errors for different periodic characterization datasets.

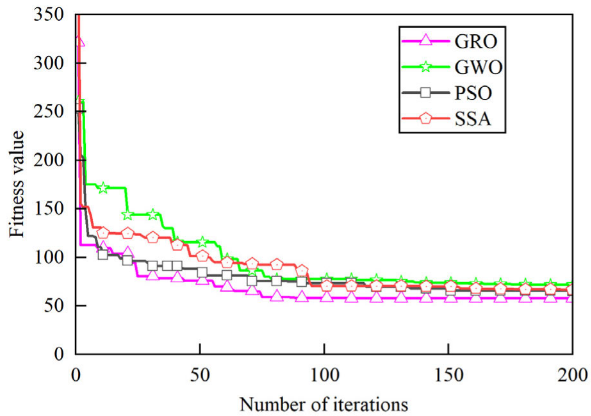

FIGURE 12. The optimization of fitness curves with different algorithmic weights.

feature extraction for the comparison of the prediction errors of two different prediction models for 14 consecutive days.

As shown in Fig. 11, it can be seen that the performance of different models changes over time with the same input data. Therefore, making full use of the results of each load forecasting model and combining different types of forecasting methods into an integrated learning model can improve the accuracy of load forecasting.

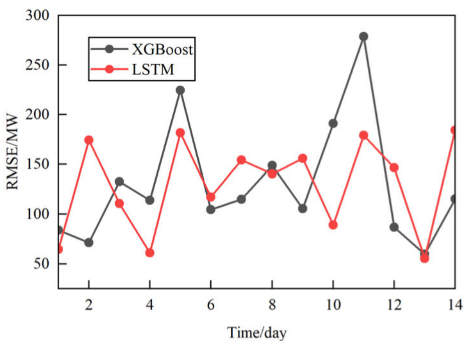

FIGURE 11. Comparison of errors of different forecasting models.

3) IMPROVED GOLD PANNING OPTIMIZATION ALGORITHM

In order to verify the effectiveness of the GRO algorithm, this paper compares the mainstream optimisation algorithms, including the Grey Wolf Algorithm (GWO), Sparrow Search Algorithm (SSA) and Particle Swarm Optimisation Algorithm (PSO), in the actual weights solving. The basic parameters of all optimization algorithms are set the same, the number of populations is 50, and the maximum iteration is 200. From the convergence curves in Fig. 12, it's clear that the GRO algorithm has faster convergence and better solutions for weight solving.

In order to verify the effectiveness of the proposed improvement strategy, this paper firstly predicts the results on the sub-model for 14 days, and takes the prediction error of the first 7 days as the basis for calculating the next weights to simulate the error feedback process, and uses the improved GRO algorithm to solve the weights, and compares it with the original unimproved GRO algorithm, as shown in Fig. 13, which shows the convergence curve of the weight update, the improved GRO algorithm has lower adaptation value in the beginning and has better convergence speed and better solution, while the original GRO algorithm is slower to converge. The results also show that the characteristics of power loads in a short period of time are highly similar.

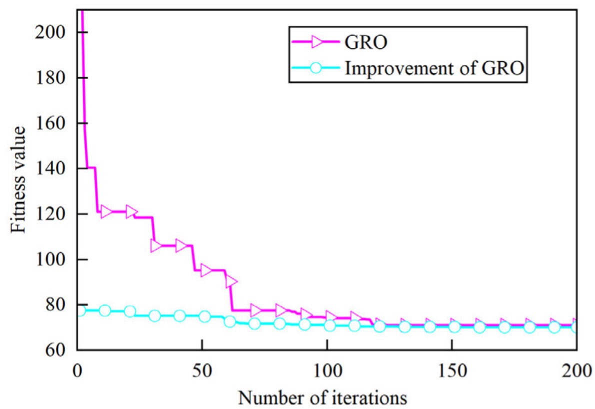

FIGURE 13. Improve GRO and GRO fitness curve.

F. ANALYSIS OF PROJECTED RESULTS

Fig. 14 gives a comparison of the prediction results in 7 days using this paper's method and the comparison algorithms, including the Bagging algorithm using averaging (this paper's algorithm removes the GRO optimization), LSTM, XGBoost, LightGBM [28], BPNN [29], TCN [30], Informer, and N-Beats [31]. As can be seen in the figure, the prediction curves of this paper's algorithm (red) are overall closer to the true value curves (black), and the prediction effect of the algorithm proposed in this paper is better than the other comparison algorithms.

The prediction errors for each day are shown in Table 2. The MAE, RMSE, and MAPE of the model proposed in this paper are smaller than those of other models for most of the time in a consecutive week, which proves that

TABLE 2. One-week forecast error.

Methods	Error	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
N-Beats	RMSE（MW）	170.40	206.10	149.68	150.98	139.71	162.83	179.83
	MAE（MW）	151.40	161.48	125.17	123.92	111.41	131.37	141.62
	MAPE（%）	4.62	4.81	3.75	3.57	3.33	4.36	4.74
Informer	RMSE（MW）	104.59	72.64	98.56	94.28	153.83	166.41	83.92
	MAE（MW）	85.38	53.32	71.08	78.88	119.15	132.78	65.95
	MAPE（%）	2.49	1.62	2.33	2.65	3.70	4.22	1.99
TCN	RMSE（MW）	113.97	132.18	234.48	237.71	98.21	148.63	111.82
	MAE（MW）	88.07	116.12	196.34	222.60	82.33	119.28	98.49
	MAPE（%）	2.76	3.74	5.78	6.64	2.66	3.99	3.37
LSTM	RMSE（MW）	178.93	102.57	145.65	60.09	170.80	138.71	116.27
	MAE（MW）	122.34	94.95	123.33	48.79	162.37	95.23	97.51
	MAPE（%）	3.74	3.00	3.61	1.47	4.89	3.28	3.30
XGBoost	RMSE（MW）	106.05	58.83	188.95	148.11	109.23	113.77	124.36
	MAE（MW）	84.59	48.52	172.54	120.97	91.29	88.21	100.97
	MAPE（%）	2.66	1.47	5.11	3.41	2.72	3.00	3.43
LightGBM	RMSE（MW）	111.42	120.05	62.17	162.83	156.47	104.36	120.19
	MAE（MW）	80.56	84.27	50.36	157.73	139.65	88.15	109.54
	MAPE（%）	2.39	2.48	1.53	4.75	4.42	2.94	3.76
BPNN	RMSE（MW）	149.49	92.30	127.54	113.38	43.48	96.39	114.15
	MAE（MW）	98.47	82.88	103.52	86.46	36.80	75.07	101.32
	MAPE（%）	3.02	2.56	3.01	2.40	1.09	2.52	3.42
Bagging	RMSE（MW）	71.84	61.31	107.76	145.06	97.76	54.78	76.71
	MAE（MW）	57.67	46.58	90.85	130.02	86.36	44.21	65.78
	MAPE（%）	1.68	1.39	2.59	3.67	2.54	1.43	2.15
Proposed	RMSE（MW）	**86.96**	**51.96**	**84.82**	**113.23**	**42.30**	**70.23**	**63.35**
	MAE（MW）	**76.97**	**41.51**	**66.24**	**97.38**	**35.39**	**61.99**	**50.05**
	MAPE（%）	**2.36**	**1.32**	**1.90**	**2.78**	**1.04**	**2.08**	**1.67**

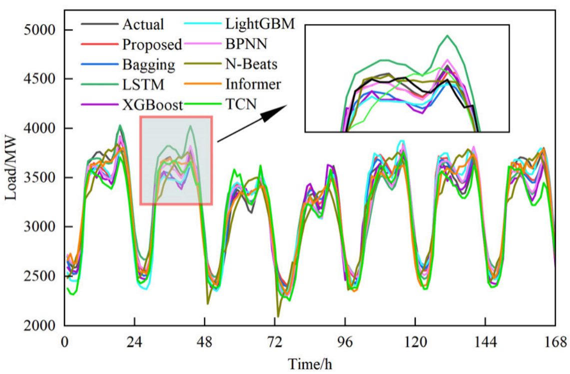

FIGURE 14. Comparison of forecast results.

TABLE 3. Comparison of prediction errors.

Methods	RMSE	MAE	MAPE
N-Beats	166.94	135.19	4.17
TCN	163.09	131.89	4.14
LSTM	135.91	106.36	3.33
LightGBM	123.65	101.47	3.18
XGBoost	126.88	101.01	3.12
Informer	115.39	86.65	2.71
BPNN	105.25	83.50	2.57
Bagging	92.60	74.50	2.21
Proposed	**73.26**	**61.36**	**1.88**

the introduction of multi-period features can improve the prediction accuracy, and the regression methods such as BPNN, LightGBM, XGBoost, and LSTM can simulate the trend of electricity consumption of different types of days in a week more accurately. The load forecasting effect of TCN and N-Beats, which use time series forecasting methods is average. Informer has good predictive accuracy as a more advanced model.

Table 3 gives a comparison of the average prediction error for one consecutive week, from which it can be seen that this paper's algorithm has the smallest prediction error in all the experiments, with MAPE as the main analysis, which is reduced by 42.64% and 38.78% compared to the single LSTM and XGBoost, respectively, which proves that the integrated learning can effectively learn the load multiple periodic features, and thus reduce the prediction error. Compared with the averaging method of the Bagging algorithm, the MAPE index is reduced by 13.57%, proving that introducing a weight optimization strategy and a weight update strategy in Bagging integrated learning can further improve the prediction accuracy. The baseline model is also compared, which is reduced by 26.84% and 39.74%

TABLE 4. Weekday 24-hour prediction error (MAPE).

Time	**Proposed**	Bagging	LSTM	XGBoost	BPNN	LightGBM	N-Beats	Informer	TCN
1:00	**1.09**	0.77	2.72	4.31	1.46	2.73	6.82	0.69	6.73
2:00	**0.14**	0.48	3.39	3.92	1.77	1.82	3.00	0.37	5.72
3:00	**0.45**	0.30	3.84	2.75	2.40	0.95	1.13	0.59	3.95
4:00	**0.33**	0.02	3.47	2.97	2.47	1.04	5.52	0.97	3.46
5:00	**0.23**	0.03	2.74	2.66	2.52	2.30	4.60	2.54	2.98
6:00	**0.14**	0.59	0.08	5.99	1.57	0.8	1.02	3.12	4.80
7:00	**0.16**	0.89	2.58	3.80	0.23	0.04	2.67	2.69	1.42
8:00	**0.69**	2.60	4.06	4.15	1.16	2.06	8.36	5.02	0.98
9:00	**0.61**	1.65	3.38	4.77	1.41	0.21	8.85	0.86	0.32
10:00	**0.10**	1.70	3.10	3.81	1.97	1.42	5.96	0.54	1.49
11:00	**1.08**	2.59	3.53	5.13	3.02	0.01	5.29	3.31	2.60
12:00	**2.41**	3.42	4.30	4.82	4.27	0.79	4.41	1.72	3.32
13:00	**3.69**	4.23	5.39	5.80	5.66	1.73	2.88	4.20	5.40
14:00	**3.91**	4.72	6.45	8.17	6.35	3.53	0.69	0.77	8.32
15:00	**4.19**	4.87	7.83	8.97	7.38	3.24	1.06	1.19	9.91
16:00	**3.36**	4.09	7.43	9.19	6.87	1.7	2.19	0.02	10.47
17:00	**3.07**	3.76	6.42	7.46	5.61	0.21	3.34	1.83	10.27
18:00	**2.99**	3.08	5.34	5.66	3.96	3.1	2.74	1.69	10.71
19:00	**3.23**	2.47	4.17	5.64	2.92	2.01	0.05	0.94	11.28
20:00	**1.54**	3.34	2.02	3.90	2.23	0.74	4.39	0.28	9.28
21:00	**1.55**	3.08	1.24	3.33	1.75	0.11	5.56	2.29	8.07
22:00	**2.85**	4.38	0.59	1.79	1.36	2.15	4.90	1.80	7.30
23:00	**4.43**	4.61	1.35	5.46	1.96	1.40	3.09	5.07	5.69
24:00	**3.43**	4.48	1.25	8.18	1.97	2.69	1.56	5.35	4.26

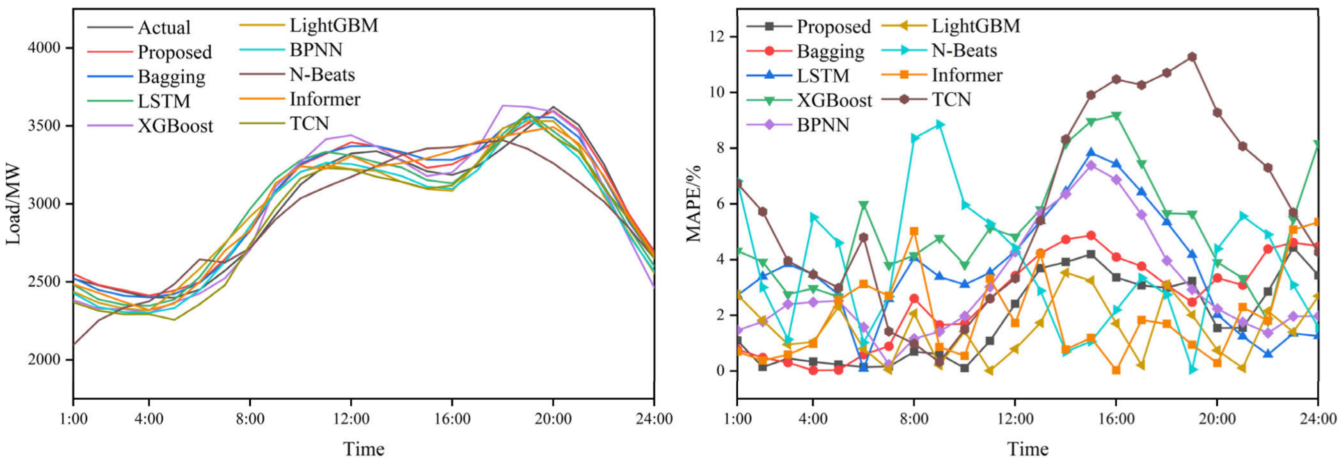

FIGURE 15. Comparison of weekday prediction results (left) and prediction error (right).

compared to the regression prediction methods BPNN and LightGBM. Compared with TCN, N-Beats, and Informer, the error is reduced by 53.86%, 54.20%, and 30.63% respectively. It proves that the GRO-Bagging day-ahead power curve forecasting model based on multi-cycle feature extraction is better and the prediction accuracy is more accurate compared to other models.

To further compare the errors of the algorithms, box plot and normal curve of absolute point error are given in this paper, as shown in Fig. 17, which shows that the method of this paper The proposed model shows a tight distribution with a median error around 50 MW. It also has the smallest interquartile range (IQR), indicating high accuracy and precision. The bagging model has a slightly wider distribution than the proposed model, with a median error of around 50 MW. However, it shows more variability and outliers. The LSTM model has a wider spread, with a median error higher than the proposed model and bagging. It shows considerable variability and several outliers. The XGBoost model and LSTM, Informer have similar distributions, the median error is about 100 MW, and there are more outliers. The LightGBM model shows a relatively large spread, with a median error slightly higher than the LSTM and XGBoost. It also has a higher number of outliers. The BPNN model shows a larger spread in error distribution, with a median error higher than the previous models and more outliers. The N-Beats model has a distribution similar to the BPNN, with a widespread and high median error. The number of outliers is also significant. The TCN model shows the widest distribution and highest

TABLE 5. Non-Working day 24-Hour prediction error (MAPE).

Time	**Proposed**	Bagging	LSTM	XGBoost	BPNN	LightGBM	N-Beats	Informer	TCN
1:00	**1.26**	0.24	1.45	5.38	3.64	3.26	16.83	1.37	5.93
2:00	**0.09**	1.20	3.66	6.06	5.63	4.52	9.09	1.96	6.61
3:00	**0.41**	1.06	3.47	4.56	5.37	4.41	4.21	2.68	5.92
4:00	**0.46**	0.01	2.23	4.16	4.13	3.48	1.03	3.35	4.45
5:00	**1.93**	0.98	0.47	1.27	2.76	1.62	3.66	1.52	5.93
6:00	**1.98**	1.79	3.33	1.01	0.09	5.48	7.95	2.00	3.79
7:00	**2.87**	2.15	6.56	1.88	2.23	6.89	2.00	4.99	3.62
8:00	**4.20**	4.45	9.43	0.20	5.35	7.61	0.15	4.40	1.05
9:00	**5.00**	5.28	8.01	6.24	4.95	4.77	0.94	6.95	1.57
10:00	**4.00**	4.29	4.98	4.58	2.57	2.63	2.82	3.75	1.24
11:00	**2.23**	2.43	2.63	5.09	0.44	0.06	4.33	0.72	0.56
12:00	**2.17**	1.42	0.40	3.52	2.03	2.98	4.50	0.50	3.07
13:00	**0.91**	0.99	2.16	0.92	3.55	3.79	2.87	2.97	4.91
14:00	**1.25**	1.62	1.43	0.16	3.09	4.27	0.99	0.56	4.20
15:00	**0.69**	2.32	1.75	0.94	3.06	3.53	4.56	2.59	3.45
16:00	**2.16**	3.08	1.68	0.55	2.80	3.18	5.57	4.81	2.10
17:00	**3.00**	2.91	0.50	3.44	0.74	0.74	4.56	4.87	0.58
18:00	**1.89**	3.70	2.70	8.08	1.70	3.77	1.48	2.18	1.85
19:00	**0.82**	1.96	2.75	3.85	1.77	1.23	3.90	0.66	2.61
20:00	**0.70**	1.87	4.28	0.85	5.13	2.51	9.98	3.57	5.24
21:00	**0.87**	2.26	4.70	1.19	6.02	3.83	10.39	3.40	4.85
22:00	**0.92**	2.51	4.23	4.65	5.71	6.29	7.34	2.30	4.67
23:00	**0.16**	1.50	2.92	5.16	4.53	4.07	2.86	0.54	1.04
24:00	**0.19**	1.49	3.53	8.66	4.73	5.22	1.63	0.88	1.63

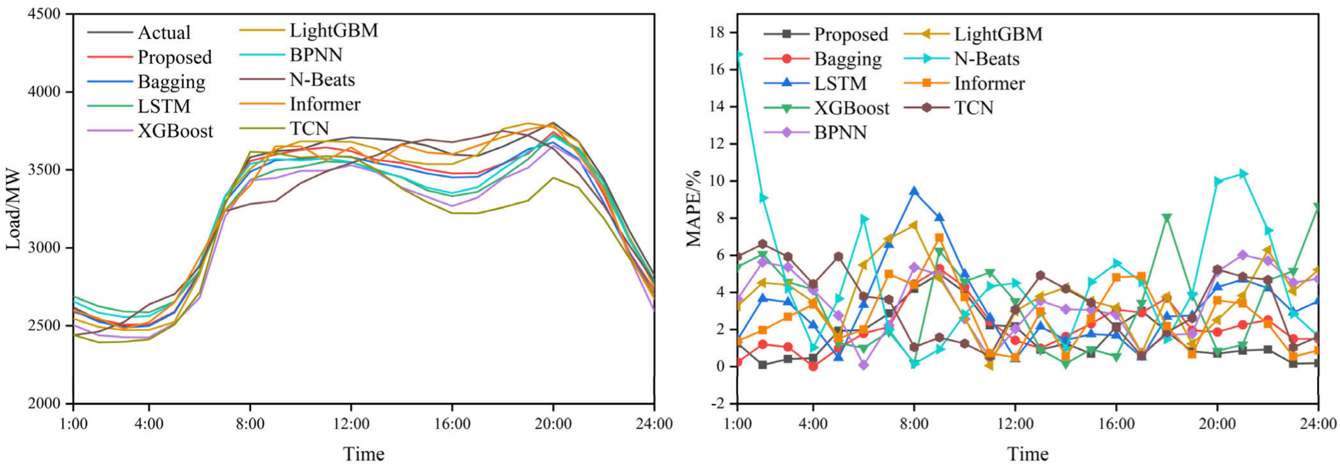

FIGURE 16. Comparison of non-weekday prediction results (left) and prediction error (right).

median error among all models, with many outliers indicating lower accuracy and precision.

For further analysis, the hour-by-hour prediction errors for weekdays (Wednesday, March 16, 2011) and non-weekdays (Sunday, March 13, 2011) are also given in this paper, as shown in Table 4 and Table 5, and the corresponding comparative graphs of the prediction results are shown in Figures 15 and 16. The results show that the forecasting model proposed in this paper is able to predict the upper and lower peak loads of the 24-hour electricity forecast more accurately, which indicates that the model proposed in this paper has a strong ability to forecast weekday and weekend loads.

G. COMPARISON OF TRAINING TIME

To assess the complexity of the proposed model, we compared the training times of all the predictive models involved in the comparison experiments as well as those presented in this paper. To avoid chance, we ran each model ten times with the optimal parameters obtained through grid search and used early stopping for the deep learning models to avoid overfitting. Table 6 summarizes the average ten training times of different prediction models, from the results in Table 6, it can be seen that XGBoost and LightGBM based on traditional machine learning have the lowest training time, followed by BPNN and LSTM, and the training time of the proposed method in this paper is 23.27s, which is similar

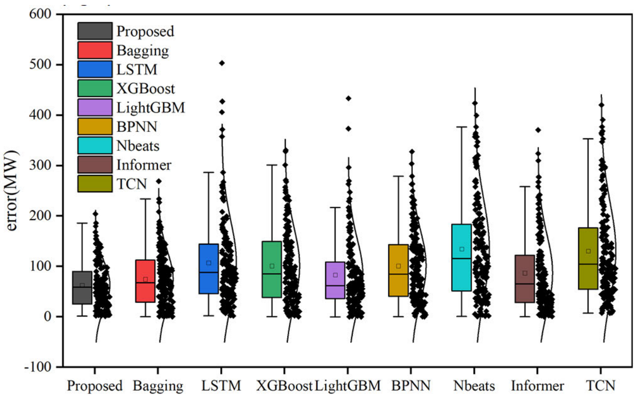

FIGURE 17. Distribution of errors between each predicted value and the true value.

to the Bagging method but much lower than that of TCN and methods such as N-Beats and Informer, etc. TCN and N-Beats have relatively more complex model structures and require more training time, and Informer requires the most training time, which is due to the fact that it has a more complex model structure and more parameters, resulting in a relatively high amount of computation. From the point of view of training time, the proposed method is in the middle of the range in terms of computation and is more efficient than the computationally intensive methods such as TCN and Informer.

TABLE 6. Comparison of training time of different models.

Methods	XGBoost	LightGBM	BPNN	LSTM	Bagging
Training time (s)	0.28	0.21	2.27	3.23	22.56
Methods	**Proposed**	N-Beats	TCN	Informer	/
Training time (s)	**23.27**	43.14	46.53	98.81	/

IV. CONCLUSION

In this paper, a GRO-Bagging day-ahead power curve forecasting model based on multi-cycle feature extraction is proposed from the multi-period nature of power loads, and the proposed method is experimentally verified by a real dataset, and the conclusions are as follows:

a) By extracting intra-periodic and inter-periodic features of historical loads under multi-periodicity, temporal pattern decoupling can be realized and new input features can be generated, which can positively contribute to load forecasting.

b) Using the Gold Rush Optimization Algorithm that introduces Tent chaotic mapping and elite strategy to solve the weights, as well as the introduction of error feedback for weight updating, possesses better prediction accuracy with better solution and faster convergence speed compared to the general integration methods.

c) The GRO-Bagging algorithm integrates the prediction of datasets with different cycle characteristics, which can effectively learn the multiple cycle change patterns of the data, and the prediction accuracy is higher.

In the future, we will further study the extraction and utilization of load multi-period features, as well as the dynamic weighting strategy considering model integration, to further realize the prediction modeling adapted to multiple time-varying loads.

REFERENCES

[1] T. Li, Y. Wang, and N. Zhang, "Combining probability density forecasts for power electrical loads," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1679–1690, Mar. 2020, doi: 10.1109/TSG.2019.2942024.

[2] B. Stephen, X. Tang, P. R. Harvey, S. Galloway, and K. I. Jennett, "Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1591–1598, Jul. 2017, doi: 10.1109/TSG.2015.2493205.

[3] S. Chapaloglou, A. Nesiadis, P. Iliadis, K. Atsonios, N. Nikolopoulos, P. Grammelis, C. Yiakopoulos, I. Antoniadis, and E. Kakaras, "Smart energy management algorithm for load smoothing and peak shaving based on load forecasting of an island's power system," *Appl. Energy*, vol. 238, pp. 627–642, Mar. 2019, doi: 10.1016/j.apenergy.2019.01.102.

[4] O. Rubasinghe, X. Zhang, T. K. Chau, Y. Chow, T. Fernando, and H. H. Iu, "A novel sequence to sequence data modelling based CNN-LSTM algorithm for three years ahead monthly peak load forecasting," *IEEE Trans. Power Syst.*, vol. 39, no. 1, pp. 1932–1947, Jan. 2024, doi: 10.1109/TPWRS.2023.3271325.

[5] X. Guo, Y. Gao, Y. Li, D. Zheng, and D. Shan, "Short-term household load forecasting based on long- and short-term time-series network," *Energy Rep.*, vol. 7, pp. 58–64, Apr. 2021, doi: 10.1016/j.egyr.2021.02.023.

[6] J. W. Taylor and P. E. McSharry, "Short-term load forecasting methods: An evaluation based on European data," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 2213–2219, Nov. 2007, doi: 10.1109/TPWRS.2007.907583.

[7] J. Zhang, Y.-M. Wei, D. Li, Z. Tan, and J. Zhou, "Short term electricity load forecasting using a hybrid model," *Energy*, vol. 158, pp. 774–781, Sep. 2018, doi: 10.1016/j.energy.2018.06.012.

[8] F. Liu, T. Dong, T. Hou, and Y. Liu, "A hybrid short-term load forecasting model based on improved fuzzy C-means clustering, random forest and deep neural networks," *IEEE Access*, vol. 9, pp. 59754–59765, 2021, doi: 10.1109/ACCESS.2021.3063123.

[9] T. Zhang, Y. Huang, H. Liao, and Y. Liang, "A hybrid electric vehicle load classification and forecasting approach based on GBDT algorithm and temporal convolutional network," *Appl. Energy*, vol. 351, Dec. 2023, Art. no. 121768, doi: 10.1016/j.apenergy.2023.121768.

[10] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—A novel pooling deep RNN," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Sep. 2018, doi: 10.1109/TSG.2017.2686012.

[11] S. Pei, H. Qin, L. Yao, Y. Liu, C. Wang, and J. Zhou, "Multi-step ahead short-term load forecasting using hybrid feature selection and improved long short-term memory network," *Energies*, vol. 13, no. 16, p. 4121, Aug. 2020, doi: 10.3390/en13164121.

[12] H. He, Z. Lu, C. Zhang, Y. Wang, W. Guo, and S. Zhao, "A data-driven method for dynamic load forecasting of scraper conveyer based on rough set and multilayered self-normalizing gated recurrent network," *Energy Rep.*, vol. 7, pp. 1352–1362, Nov. 2021, doi: 10.1016/j.egyr.2021.09.127.

[13] L. Yin and J. Xie, "Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems," *Appl. Energy*, vol. 283, Feb. 2021, Art. no. 116328, doi: 10.1016/j.apenergy.2020.116328.

[14] B. Zhang, J.-L. Wu, and P.-C. Chang, "A multiple time series-based recurrent neural network for short-term load forecasting," *Soft Comput.*, vol. 22, no. 12, pp. 4099–4112, Jun. 2018, doi: 10.1007/s00500-017-2624-5.

[15] R. Chen, C. S. Lai, C. Zhong, K. Pan, W. W. Y. Ng, Z. Li, and L. L. Lai, "MultiCycleNet: Multiple cycles self-boosted neural network for short-term electric household load forecasting," *Sustain. Cities Soc.*, vol. 76, Jan. 2022, Art. no. 103484, doi: 10.1016/j.scs.2021.103484.

[16] Z. Cao, C. Wan, Z. Zhang, F. Li, and Y. Song, "Hybrid ensemble deep learning for deterministic and probabilistic low-voltage load forecasting," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 1881–1897, May 2020, doi: 10.1109/TPWRS.2019.2946701.

[17] J. Lee and Y. Cho, "National-scale electricity peak load forecasting: Traditional, machine learning, or hybrid model?" *Energy*, vol. 239, Jan. 2022, Art. no. 122366, doi: 10.1016/j.energy.2021.122366.

[18] L. Wang, S. Mao, B. M. Wilamowski, and R. M. Nelms, "Ensemble learning for load forecasting," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 2, pp. 616–628, Jun. 2020, doi: 10.1109/TGCN.2020.2987304.

[19] X. Hu, X. Tang, Q. Zhang, Z. Chen, and Y. Zhang, "Combined machine learning forecasting method for short-term power load based on the dynamic weight adjustment," *Energy Rep.*, vol. 9, pp. 866–873, Sep. 2023, doi: 10.1016/j.egyr.2023.04.082.

[20] V. Álvarez, S. Mazuelas, and J. A. Lozano, "Probabilistic load forecasting based on adaptive online learning," *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3668–3680, Jul. 2021, doi: 10.1109/TPWRS.2021.3050837.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015. Accessed: May 3, 2024.

[22] Y. Lu, G. Wang, and S. Huang, "A short-term load forecasting model based on mixup and transfer learning," *Electr. Power Syst. Res.*, vol. 207, Jun. 2022, Art. no. 107837, doi: 10.1016/j.epsr.2022.107837.

[23] K. Zolfi, "Gold rush optimizer: A new population-based metaheuristic algorithm," *Oper. Res. Decisions*, vol. 33, no. 1, pp. 113–150, Jan. 2023, doi: 10.37190/ord230108.

[24] A. S. Khwaja, A. Anpalagan, M. Naeem, and B. Venkatesh, "Joint bagged-boosted artificial neural networks: Using ensemble machine learning to improve short-term electricity load forecasting," *Electr. Power Syst. Res.*, vol. 179, Feb. 2020, Art. no. 106080, doi: 10.1016/j.epsr.2019.106080.

[25] X. Deng, A. Ye, J. Zhong, D. Xu, W. Yang, Z. Song, Z. Zhang, J. Guo, T. Wang, Y. Tian, H. Pan, Z. Zhang, H. Wang, C. Wu, J. Shao, and X. Chen, "Bagging–XGBoost algorithm based extreme weather identification and short-term load forecasting model," *Energy Rep.*, vol. 8, pp. 8661–8674, Nov. 2022, doi: 10.1016/j.egyr.2022.06.072.

[26] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," *Int. J. Forecasting*, vol. 32, no. 3, pp. 896–913, Jul. 2016, doi: 10.1016/j.ijforecast.2016.02.001.

[27] H. H. Goh, B. He, H. Liu, D. Zhang, W. Dai, T. A. Kurniawan, and K. C. Goh, "Multi-convolution feature extraction and recurrent neural network dependent model for short-term load forecasting," *IEEE Access*, vol. 9, pp. 118528–118540, 2021, doi: 10.1109/ACCESS.2021.3107954.

[28] Y. Wang, J. Chen, X. Chen, X. Zeng, Y. Kong, S. Sun, Y. Guo, and Y. Liu, "Short-term load forecasting for industrial customers based on TCN-LightGBM," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 1984–1997, May 2021, doi: 10.1109/TPWRS.2020.3028133.

[29] Y. Hu, J. Li, M. Hong, J. Ren, R. Lin, Y. Liu, M. Liu, and Y. Man, "Short term electric load forecasting model and its verification for process industrial enterprises based on hybrid GA-PSO-BPNN algorithm—A case study of papermaking process," *Energy*, vol. 170, pp. 1215–1227, Mar. 2019, doi: 10.1016/j.energy.2018.12.208.

[30] M. Liu, H. Qin, R. Cao, and S. Deng, "Short-term load forecasting based on improved TCN and DenseNet," *IEEE Access*, vol. 10, pp. 115945–115957, 2022, doi: 10.1109/ACCESS.2022.3218374.

[31] B. N. Oreshkin, G. Dudek, P. Pełka, and E. Turkina, "N-BEATS neural network for mid-term electricity load forecasting," *Appl. Energy*, vol. 293, Jul. 2021, Art. no. 116918, doi: 10.1016/j.apenergy.2021.116918.

YAOXIAN LIU was born in 1990. He received the Ph.D. degree in electrical engineering from North China Electric Power University, in 2020. He is currently a Lecturer with the School of Electrical and Control Engineering, Shaanxi University of Science and Technology. His research interests include new power system demand response, load forecasting, non-invasive load monitoring, and decomposition.

KAIXIN ZHANG was born in 2000. He received the bachelor's degree in engineering from Shaanxi University of Science and Technology, in 2022, where he is currently pursuing the master's degree with the School of Electrical and Control Engineering. His research interests include load forecasting and load regulation.

SONGSONG CHEN was born in 1987. He received the Ph.D. degree. He is currently a Senior Engineer with China Electric Power Research Institute Company Ltd. His research interests include energy efficiency and intelligent electricity technology.

YING ZHOU was born in 1993. She received the master's degree. She is currently an Engineer with China Electric Power Research Institute Company Ltd. Her research interests include power system power quantity analysis and prediction.

JINGWEN CHEN was born in 1978. He received the Ph.D. degree in engineering from Shaanxi University of Science and Technology, in 2020. He is currently a Professor with the School of Electrical and Control Engineering, Shaanxi University of Science and Technology. His research interests include electric vehicle charging load optimization scheduling, microgrids, and hybrid energy storage technology.