




Article

A Combined Method for Short-Term Load Forecasting Considering the Characteristics of Components of Seasonal and Trend Decomposition Using Local Regression

Sile Hu ^{1,2} , Yuan Wang ³, Wenbin Cai ³, Yuan Yu ², Chao Chen ¹, Jiaqiang Yang ^{1,*} , Yucan Zhao ¹  and Yuan Gao ¹

¹ College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China; 12110087@zju.edu.cn (S.H.); 222601119@zju.edu.cn (C.C.); zhaoyucan@zju.edu.cn (Y.Z.); 22210051@zju.edu.cn (Y.G.)

² Inner Mongolia Power (Group) Co., Ltd., Hohhot 010020, China; 13947181808@163.com

³ Inner Mongolia Electric Power Economic and Technological Research Institute, Hohhot 010090, China; wangyuan126126@163.com (Y.W.); 18686016708@163.com (W.C.)

* Correspondence: yangjiaq@zju.edu.cn; Tel.: +86-0571-8795-1243

Abstract: In response to the complexity and high volatility of original load data affecting the accuracy of load forecasting, a combined method for short-term load forecasting considering the characteristics of components of seasonal and trend decomposition using local regression (STL) is proposed. The original load data are decomposed into a trend component, seasonal component, and residual component using STL. Then, considering the characteristics of each component, a long short-term memory (LSTM) neural network, a convolutional neural network (CNN), and Gaussian process regression (GPR) are used to predict the trend component, seasonal component, and residual component, respectively. The final outcome of the load forecasting is obtained by summing the forecasted results of each individual component. A specific case study is conducted to compare the proposed combined method with LSTM, CNN, GPR, STL-LSTM, STL-CNN, and STL-GPR prediction methods. Through comparison, the proposed combined method exhibits lower errors and higher accuracy, demonstrating the effectiveness of this method.

Keywords: short-term load forecasting; seasonal and trend decomposition using local regression; long and short-term memory network; convolutional neural network; Gaussian process regression



Citation: Hu, S.; Wang, Y.; Cai, W.; Yu, Y.; Chen, C.; Yang, J.; Zhao, Y.; Gao, Y. A Combined Method for Short-Term Load Forecasting Considering the Characteristics of Components of Seasonal and Trend Decomposition Using Local Regression. *Appl. Sci.* **2024**, *14*, 2286. <https://doi.org/10.3390/app14062286>

Academic Editors: Gaetano Zizzo and Roberto Zivieri

Received: 11 January 2024

Revised: 4 March 2024

Accepted: 7 March 2024

Published: 8 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The primary purpose of the power system is to meet the demand for electricity. The ongoing advancement of the power grid has led to increasingly intricate fluctuations in power load [1], rendering research on power load forecasting a pivotal component of grid management. Load forecasting provides a clear understanding of future demand levels and patterns, forming the basis for the planning and decision-making process. By forecasting future electricity demand, power companies can reasonably plan the generation, transmission, and distribution facilities to avoid power shortages and reduce power outages. Load forecasting also helps optimize resource allocation, reduce operating costs, and improve overall system efficiency. It can be said that load forecasting is an indispensable part of power system planning and scheduling, as it relates to the stability and sustainability of power supply, and has significant impacts on economic and social development [2]. Therefore, load forecasting has always been a critical issue in the field of electricity.

Load forecasting is classified into traditional classic forecasting methods and intelligent forecasting methods. Initially, traditional classic forecasting methods based on statistical theory were widely utilized, primarily encompassing time series analysis [3] and regression analysis [4], among others. These methods heavily depend on historical data and mathematical models for prediction. While they feature simplified models, low computational complexity, and rapid prediction speed, their lack of robustness makes

them susceptible to random disruptions, resulting in reduced reliability when employed for intricate regional load forecasting. With the advancement of intelligent algorithms, some electricity load forecasting methods based on these algorithms have emerged. These algorithms comprise support vector machines (SVMs) [5], long short-term memory (LSTM) neural networks [6], gated recurrent units (GRUs) [7], and convolutional neural networks (CNNs) [8], which are adept at handling the nonlinear relationships and complex features of the load. These methods can effectively handle nonlinearity and multidimensional data, but they are sensitive to outliers, and facing data mutations or breakpoints can lead to performance degradation [9].

However, the high complexity of the load sequence limits the accuracy of load prediction using a single algorithm. In recent years, signal decomposition techniques such as wavelet transform [10] and Empirical Mode Decomposition (EMD) [11] have been widely applied in the field of load forecasting. These techniques decompose the load sequence into multiple components and individually predict each component to enhance feature extraction and prediction accuracy of the load sequence. In reference [12], a short-term load forecasting model based on EMD is established. EMD is used to decompose load data into a series of local features with different frequencies and time scales. The components are then predicted using a GRU and superimposed to obtain the final load forecast. In reference [13], a load decomposition using Variational Mode Decomposition (VMD) is utilized to obtain components with different frequencies, featuring smooth and approximately sinusoidal waveforms. The components are predicted and superimposed using an LSTM model, achieving effective load prediction and avoiding the mode-mixing phenomenon caused by EMD. Reference [14] proposes a prediction method through dual decomposition. It employs Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEM-DAN) to decompose load data, generating multiple Intrinsic Mode Functions (IMFs). It then utilizes VMD to perform a secondary decomposition on the IMF with the maximum sample entropy. A Support Vector Regression (SVR) model is established to predict each decomposed subsequence separately, and these subsequences are then superimposed to obtain the final load forecast.

Although the above methods all achieved effective load prediction, they did not consider the specific characteristics of the load, and they focused more on the decomposition of the signal's high-frequency and low-frequency components, resulting in poor interpretability. In reference [15], a method is introduced that utilizes the seasonal and trend decomposition using a local regression (STL) algorithm to decompose the load sequence based on seasonality and trend. It then uses the Light Gradient Boosting Machine algorithm to predict the components. This method fully considers the trend, seasonality, and volatility of the load. However, it uses the same model to predict components with different characteristics and does not consider the matching between component characteristics and prediction methods, resulting in insufficient feature extraction for the load.

In response to the above-mentioned problem, this paper introduces a novel short-term load combination forecasting method that utilizes STL. The approach integrates Loess decomposition technology with hybrid model forecasting methods. Through the decomposition of the original load data into a trend component, seasonal component, and residual component via STL, appropriate models are then applied to predict the characteristics of each component. Specifically, the trend component is forecasted using the LSTM model, the seasonal component utilizing the CNN model, and the residual component using the Gaussian process regression model. The forecast results from these models are then aggregated to derive the final load forecast. By presenting specific examples and conducting a comparative analysis with existing methods such as LSTM, CNN, GPR, STL-LSTM, STL-CNN, and STL-GPR, the efficacy and superiority of the proposed method in practical applications are highlighted. This method ensures a more precise capture and prediction of the intricate dynamics of power load compared to traditional standalone models or methods that overlook component characteristics, thereby enhancing prediction accuracy. By introducing STL decomposition and hybrid model techniques, this research

offers a fresh perspective and tool for load forecasting studies, particularly in handling load data exhibiting complex seasonal and trend patterns.

2. The Proposed Combined Method and Principles

2.1. The Proposed Combined Method

The load sequence can be extracted using STL to identify a trend component, seasonal component, and residual component. Specifically, the trend component signifies the evolving pattern of the load, the seasonal component illustrates the cyclic variations in the load, and the residual component captures the stochastic and uncertain aspects of load fluctuations. Each component manifests distinctive characteristics of change. Employing a single model for predicting each component might hinder the ability to capture the unique attributes of the individual load components, consequently reducing the precision of load forecasting.

The trend component exhibits gradual changes concealed within the dynamically shifting load time series, rendering them challenging to capture. For this reason, the network model needs to have long-term memory and the capacity to analyze and extract the inter-relations of information at distinct time junctures. LSTM has flexible memory and generalization capabilities, making it suitable for predicting relatively smooth trend components. Within the load sequence, there are fixed similar patterns and regularities, namely the periodic patterns of load changes. While LSTM can transmit information from different time nodes, capturing global features, it lacks perception of the local features of the load. On the other hand, the key feature of a CNN is local connectivity and weight sharing, having the ability to convey information from different time points. Given this characteristic, this paper applies a CNN to the prediction of the load's seasonal component. GPR is a non-parametric method able to capture the characteristics of sequences with complex nonlinear patterns and significant uncertainty, and it provides useful uncertainty quantification, making it suitable for predicting residual components with high randomness. In conclusion, in the proposed combined method, LSTM, a CNN, and GPR are used to predict the trend component, seasonal component, and residual component, respectively.

The proposed combined method consists of three steps: load decomposition, component prediction, and component summation. The specific prediction method flowchart is shown in Figure 1. First, the load sequence is decomposed into three components using STL: the trend component, the seasonal component, and the residual component. Next, LSTM, CNN, and GPR models are constructed to forecast the trend, seasonal, and residual components, respectively. Finally, the predicted results of the three components are aggregated to obtain the final load forecast.

2.2. Principle of STL

STL is a time series decomposition algorithm based on local weighted regression. In the proposed combined method, STL is used to decompose load data into three components: a seasonal component, a trend component, and a residual component. The relationship between each component and the original load data is shown in formula (1).

$$Y_t = S_t + T_t + R_t \quad (1)$$

where Y_t represents the value of the original load data at time t ; S_t represents the value of the seasonal component at time t ; T_t represents the value of the trend component at time t ; R_t represents the value of the residual component at time t .

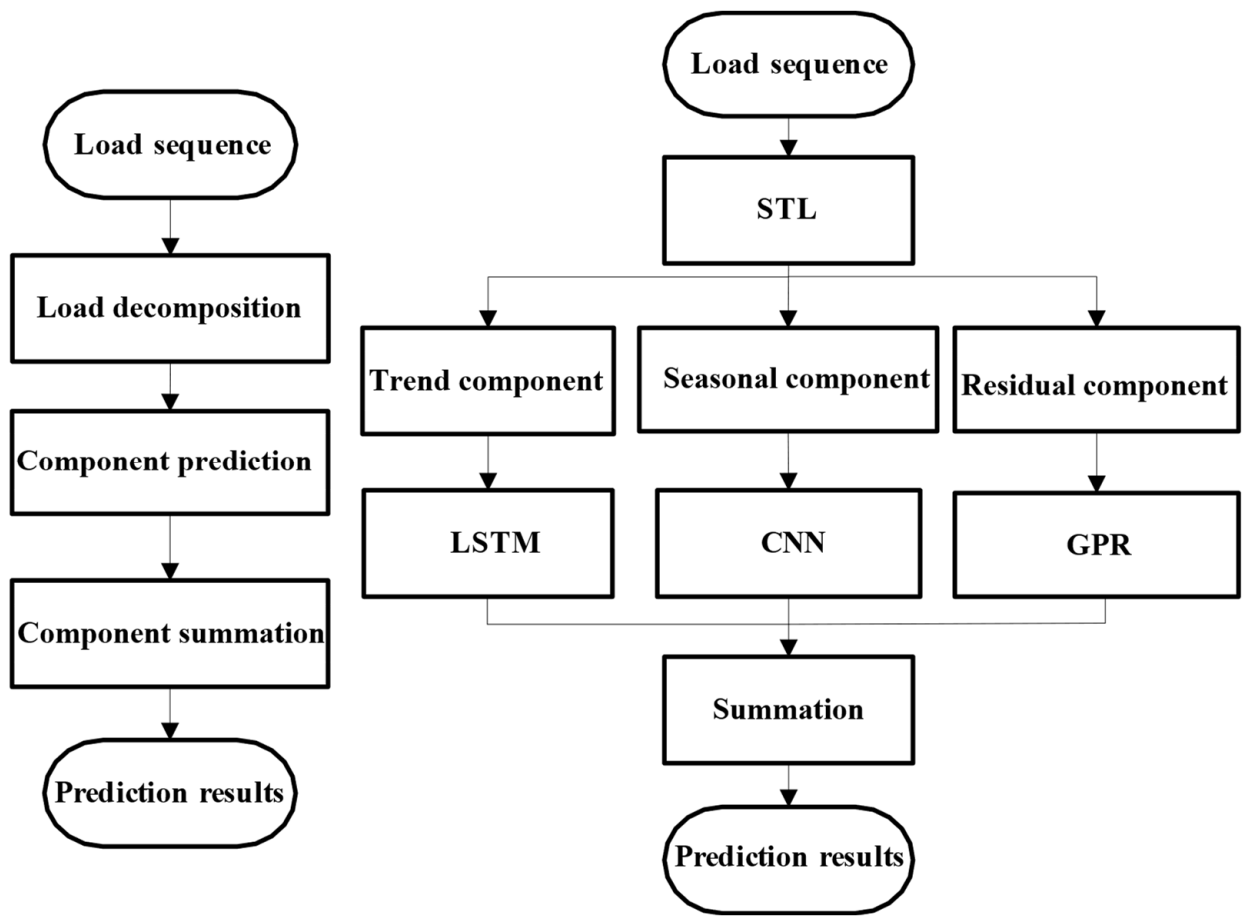


Figure 1. Flowchart of the proposed combined method.

STL uses locally estimated scatterplot smoothing (Loess) to extract smooth estimates for three components; it consists of two recursive processes, the inner loop and the outer loop, and has good robustness. The inner loop, based on Loess, smooths the seasonal and trend components, while the outer loop computes robust weights based on the residual component to reduce the influence of time series outliers on the residuals [16]. The calculation formula for robust weights is shown in formulas (2) and (3).

$$\rho_t = B\left(\frac{|R_t|}{6\text{median}(|R_t|)}\right) \tag{2}$$

$$B(a) = \begin{cases} (1 - a^2)^2 & \text{for } 0 \leq a < 1 \\ 0 & \text{for } a \geq 1 \end{cases} \tag{3}$$

where ρ_t represents the robust weight at time t ; $\text{median}(\cdot)$ denotes the median function; $B(a)$ denotes the Bisquare function; a represents the independent variable of the Bisquare function.

In the next inner loop, the neighborhood weights in Loess will be updated based on the robust weights, following the update method shown in formula (4).

$$v_{t,k+1} = v_{t,k}\rho_t \tag{4}$$

where $v_{t,k}$ represents the neighborhood weight at time t in the k -th inner loop.

The flowchart of STL is shown in Figure 2.

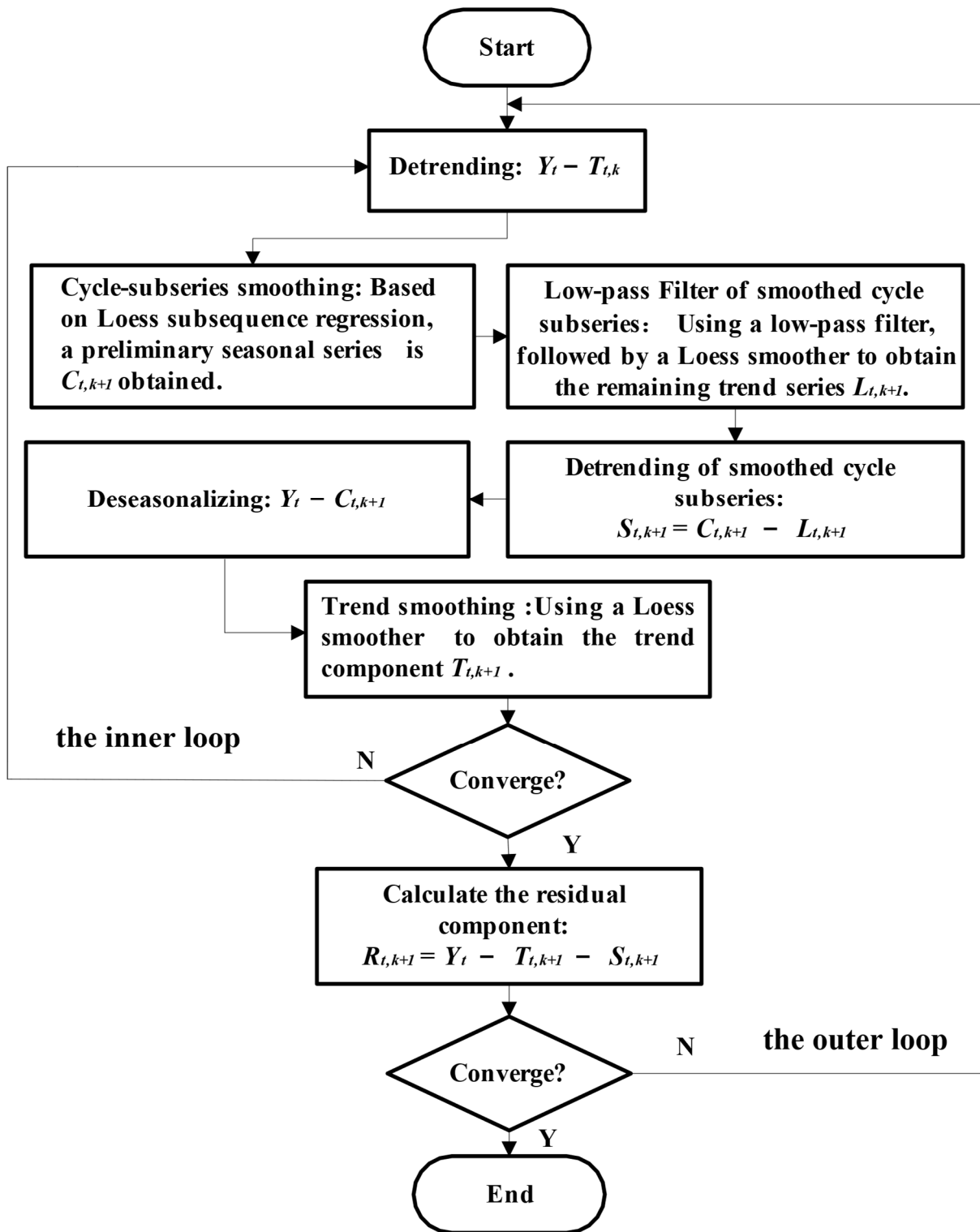


Figure 2. Flowchart of STL.

2.3. Principle of LSTM

LSTM is a specific type of recurrent neural network (RNN) that specializes in processing sequences of data with varying lengths, such as time series, speech, or text. Its distinctive architecture enables it to retain long-term dependencies, which effectively mitigates the vanishing gradient issue prevalent in traditional RNNs. An LSTM unit comprises three essential components: the input gate, the forget gate, and the output gate, which

manage the flow of information into and out of the cell state, a critical element of the LSTM. A schematic diagram of the LSTM loop unit structure is shown in Figure 3.

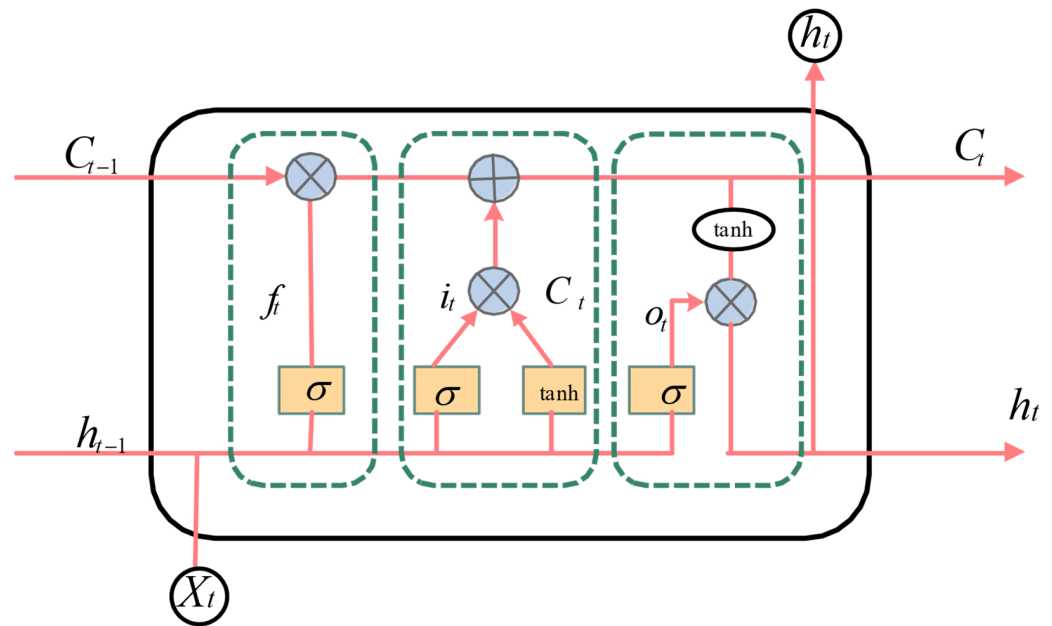


Figure 3. Schematic diagram of the LSTM loop unit structure.

The input gate determines the extent to which the new input is integrated into the cell state by utilizing a sigmoid activation function to filter inputs, as well as a tanh function to provide a new candidate value, adjusted by the sigmoid’s output. The forget gate, on the other hand, assesses the relevance of past information to the present cell state by employing a sigmoid function to allocate weights to the information, with values close to 1 indicating ‘retain’ and values near 0 indicating ‘discard’. The cell state serves as the memory track of the LSTM, extending throughout the entire unit chain, allowing for the addition or removal of information based on the input and forget gates. The output gate determines the content of the subsequent hidden state, containing information that is forwarded to the next LSTM unit and dictates the final output of the LSTM. The interplay of these gates enables the LSTM to capture and retain long-term dependencies, thus overcoming the constraints of standard RNNs, making it particularly effective for language modeling, machine translation, and speech recognition tasks [17]. In LSTM, the detailed calculations between the modules are shown in formula (5).

$$\begin{cases} f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \\ o_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \\ h_t = o_t \cdot \tanh(c_t) \end{cases} \quad (5)$$

where x_t is the input at time t ; f_t is the calculation result of the forget gate at time t ; i_t is the calculation result of the input gate at time t ; \tilde{c}_t is the candidate cell state information at time t ; c_t is the cell state information at time t ; o_t is the calculation result of the output gate at time t ; h_t is the hidden state information at time t ; σ is the sigmoid activation function; W_f is the weight matrix of the forget gate; W_i is the weight matrix of the input gate; W_o is the weight matrix of the output gate; W_c is the weight matrix of the memory cell; b_f is the bias of the forget gate; b_i is the bias of the input gate; b_c is the bias of the activation function of the input gate; b_o is the bias of the output gate.

2.4. Principle of CNN

A CNN is a type of feedforward neural network that is structured with five modules: the input layer, the convolutional layers, the activation layers, the pooling layers, and the fully connected layer [18,19]. The layered architecture of CNNs excels at extracting spatial and temporal features from load data, facilitating precise predictions.

The input layer serves as the starting point for data processing in load forecasting, handling historical load figures. These data are formatted to meet the requirements of the CNN, laying the groundwork for subsequent feature extraction. The convolutional layers are responsible for feature extraction from the input data. By applying various filters, these layers identify critical patterns in the data, such as cyclical fluctuations in power usage, which are essential for comprehending and predicting load patterns. Following the convolutional layers, the activation layers (typically ReLU) introduce nonlinearity, assisting the network in learning more complex data patterns, which is particularly beneficial in capturing and understanding irregular load variations in forecasting. Pooling layers are employed to reduce the dimensionality of the feature data, decreasing computational load while avoiding overfitting. In load forecasting, pooling layers refine and compress feature information, concentrating the model on the most crucial data characteristics. Finally, the fully connected layers represent the ultimate step in a CNN, integrating the features extracted by all previous layers to make the final prediction.

The core of CNN lies in its convolutional layers, which extract diverse feature maps from the input data through convolutional computations with different kernels, as represented by formula (6).

$$X^{m+1} = \sigma(\sum W^m \cdot X^m + B^m) \quad (6)$$

where X^{m+1} represents the input to the next layer, σ signifies the activation function, W^m denotes the weight of the current layer, X^m represents the input to the current layer, and B^m indicates the bias.

2.5. Principle of GPR

GPR is a non-parametric regression method based on Bayesian theory, and its core idea is to view data points as a whole Gaussian process. To apply Gaussian processes to regression analysis, the regression function $\{f(x)|x \in D\}$ can be regarded as a Gaussian process [20]. According to the properties of Gaussian processes, a Gaussian process can be fully determined by its mean function $m(x)$ and kernel function $k(x, x')$. Regarding $f(x)$ as a Gaussian process means that for any selected number of points from D , the probability distribution of $f(x)$ follows a Gaussian distribution, as represented by formula (7).

$$f(x) \sim GP(m(x), k(x, x')) \quad (7)$$

The mean function and the covariance function can be expressed as represented by formula (8).

$$\begin{cases} m(x) = \mathbb{E}(f(x)) \\ k(x, x') = \mathbb{E}[(m(x) - f(x))(m(x') - f(x')))] \end{cases} \quad (8)$$

Although Gaussian process regression does not require specifying the form of the regression function, it is necessary to determine the forms of the mean function and the kernel function. In GPR, selecting an appropriate kernel function is crucial as it determines how the model interprets the relationships between data points. In this article, the mean function a is set to 0, and the Gaussian kernel is chosen as the kernel function for GPR, as shown in formula (9).

$$k(x, x') = \sigma^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right) \quad (9)$$

where σ^2 is the variance parameter, determining the overall range of function values; l is the length scale parameter, controlling the rate at which function values change with the distance between input points.

For practical modeling, the observed values should be the sum of the regression function $f(x)$ and an independently and identically distributed Gaussian white noise ε , as shown in formula (10).

$$y = f(x) + \varepsilon, \varepsilon \sim N(0, \sigma_n^2) \quad (10)$$

Therefore, the prior distribution of the noisy observed values is

$$\mathbf{y} \sim N(0, \mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I}_n). \quad (11)$$

where vector \mathbf{y} refers to the observed values $[y_1, y_2, y_3, \dots, y_n]$; vector \mathbf{x} refers to the input vector $[x_1, x_2, x_3, \dots, x_n]$; and $\mathbf{K}(\mathbf{x}, \mathbf{x}) = (k(x_i, x_j))_{n \times n}$ is a covariance matrix, where each matrix element is computed through the kernel function, which is specifically represented by formula (8). \mathbf{I}_n is an n -dimensional identity matrix, while $\sigma_n^2 \mathbf{I}_n$ represents the covariance matrix of the noise.

In this GPR model, σ_n^2 is a parameter, and in formula (8), there are also parameters σ^2 and l . These parameters can all be obtained by maximizing the marginal likelihood function. Denoting all the parameters in the kernel function as $\boldsymbol{\theta} = [\sigma^2, l]$, the logarithm of the marginal likelihood (LML) is given by formula (12).

$$LML = \log p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}, \sigma_n^2) = \frac{1}{2} \log(\det(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I}_n)) - \frac{1}{2} \mathbf{y}^T [\mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbf{I}_n]^{-1} \mathbf{y} - \frac{n}{2} \log 2\pi \quad (12)$$

By applying maximum likelihood estimation to the training dataset \mathbf{x} and \mathbf{y} , the parameters in the GPR model can be determined. Once the GPR model has been established, predictions can be made on the test dataset \mathbf{x}^* .

3. Experimental Analysis

3.1. Evaluation Indicators

In order to evaluate the forecasting accuracy of the prediction methods, this paper adopts three evaluation indicators: mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean square error (RMSE). The smaller the values of these indicators, the smaller the prediction errors and the higher the forecasting accuracy of the prediction methods. Their calculation formulas are shown in formulas (13) to (15).

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100\% \quad (13)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \quad (14)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2} \quad (15)$$

where N represents the total number of samples; \hat{y}_i is the predicted load at time i ; y_i is the actual load at time i .

3.2. Experimental Dataset

This paper uses the actual electricity load measured in a province in northern China from 1 January–December 2021 as the experimental dataset, with a sampling interval of 1 h, totaling 8760 samples. According to a 4:1 ratio, the dataset is divided into a training set and a testing set, with the first 7008 load data samples used as the training set and the remaining 1752 load samples used as the testing set.

3.3. Load Decomposition

STL is used to decompose the 8760 load data samples. Figure 4 shows the load data samples for the first two weeks (336 h) and their STL results. From Figure 4, it is observed

that the original load sequence, after STL, is decomposed into a relatively smooth-changing trend component, a seasonal component with a periodicity of approximately 24 h, and a residual component with significant fluctuations.

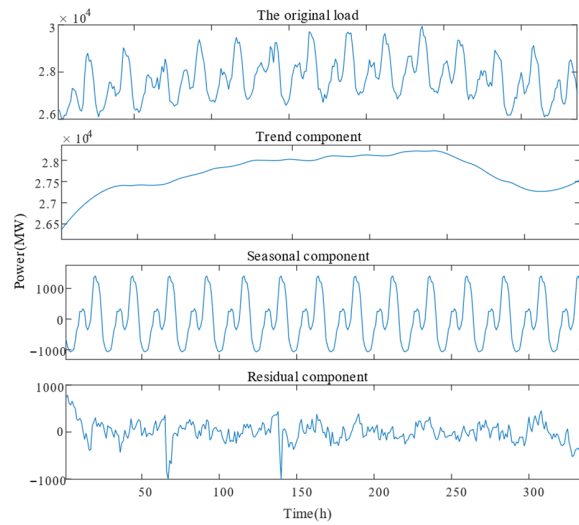


Figure 4. Values of the original load and its components for the previous 336 h.

3.4. Component Prediction

After load decomposition, different models are used to train and predict the three components. The predictive results of these three components under different models will be individually presented next.

3.4.1. Trend Component Prediction Based on LSTM

In the proposed combined method, the trend component is input into the LSTM model. Within the internal structure of LSTM, the load data at each time step undergo processing through the input gate, forget gate, and output gate. These gating units determine the cell state and hidden state at the current time step. At the same time, the data at each time step, together with the hidden state obtained from the previous time step, are used to update the cell state and pass it to the next time step. Ultimately, LSTM utilizes this information to predict the trend component of future loads. In the trend component prediction based on LSTM, the number of LSTM layers is set to three, with 64 units per layer, and the activation function is set to tanh.

Figure 5 shows the prediction results of the trend component of the test set load data for the first two weeks (336 h). It can be seen that the LSTM model demonstrates a smooth and accurate prediction of the trend component, with a high degree of fit between the actual and predicted value curves.

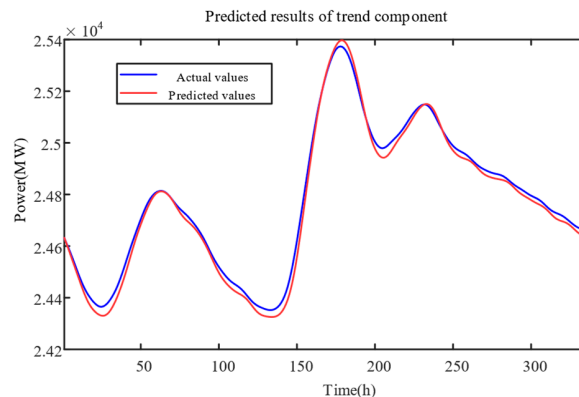


Figure 5. Predicted results of trend component for the previous 336 h.

To further illustrate the accuracy of LSTM in predicting the trend component, Figure 6 shows the MAPE values of the trend component of the load data at each moment in the first two weeks (336 h). It demonstrates MAPE values ranging between 0 and 0.003, indicating a high level of prediction accuracy. Furthermore, the curve of the MAPE value is notably smooth.

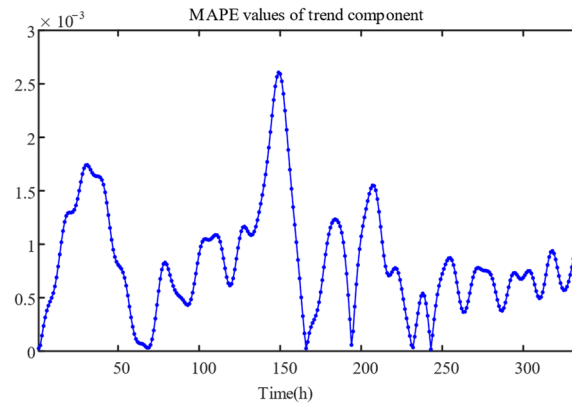


Figure 6. MAPE values of trend component at each timestamp for the previous 336 h.

3.4.2. Seasonal Component Prediction Based on CNN

In the proposed combined method, the seasonal component is first reshaped into a two-dimensional structure and then input into the CNN model for feature extraction and downsampling through convolutional and pooling layers. Subsequently, the fully connected layer processes the feature map and outputs the forecasted results of the seasonal component. Overall, in the CNN model, the load data undergo processing through modules such as convolution, pooling, flattening, and fully connected layers, enabling the prediction of the future load seasonal component. In the CNN-based seasonal component prediction, both convolutional layers utilize 32 filters with a filter size of 3, a stride of 1, and ‘same’ padding, with a tanh activation function. A max pooling layer with a pool size of 2 is added between the two convolutional layers. The fully connected layer has 50 neurons with a tanh activation function, and the output layer has 1 neuron.

Figure 7 shows the prediction results of the seasonal component of the test set load data for the first two weeks (336 h). It can be observed that the CNN model aptly captures the regularity and periodicity of the seasonal component, enabling effective forecasting for this component.

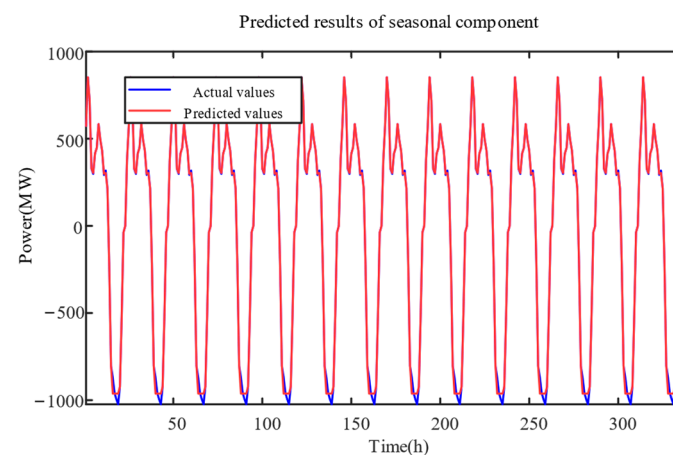


Figure 7. Predicted results of seasonal component for the previous 336 h.

To further demonstrate the precision of the CNN in predicting the seasonal component, Figure 8 displays the MAPE values of the seasonal component at each moment during

the initial two weeks (336 h). It illustrates that the forecast results have MAPE values ranging from 0 to 0.4, with most values falling within the 0–0.1 range, indicating a high level of prediction accuracy. Additionally, the MAPE value curve shows a certain degree of periodicity.

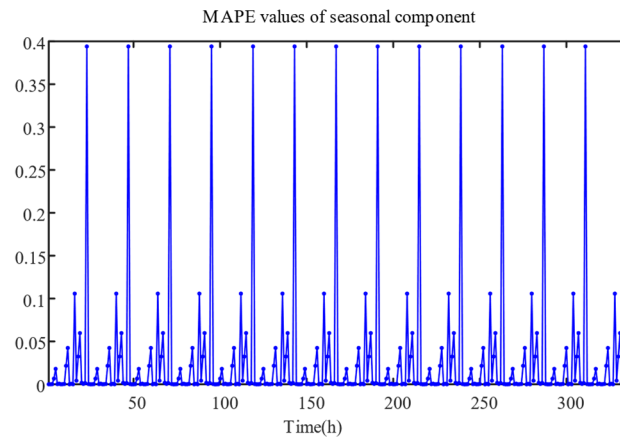


Figure 8. MAPE values of seasonal component at each timestamp for the previous 336 h.

3.4.3. Residual Component Prediction Based on GPR

In the proposed combined method, the residual component is input into the GPR model. The GPR model estimates the potential load distribution based on the residual component and provides predictions for future load residual components. In the prediction of residual components based on GPR, the noise level is set as 0.3, the mean function is set to 0, and the kernel function selected is the Gaussian kernel, as shown in formula (6). The hyperparameters in the Gaussian kernel function are automatically optimized through the maximization of the log marginal likelihood function, as shown in formula (12).

Figure 9 shows the prediction results of the residual component of the test set load data for the first two weeks (336 h). The GPR model demonstrates its effectiveness in predicting residual components characterized by substantial variability and randomness.

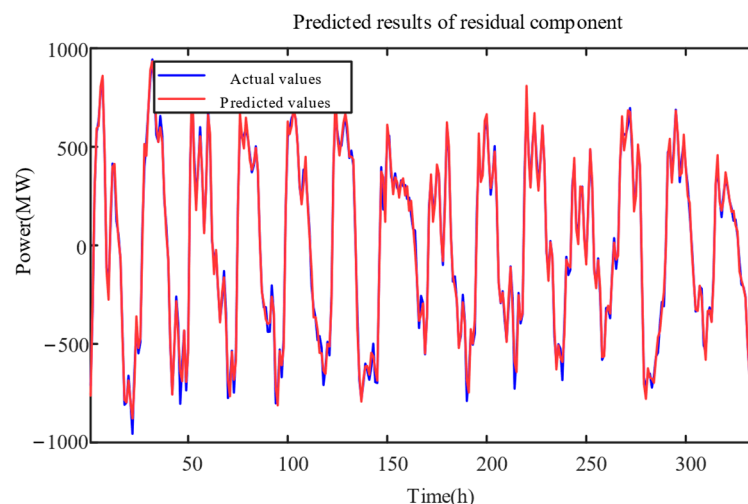


Figure 9. Predicted results of residual component for the previous 336 h.

To further demonstrate the precision of GPR in predicting the residual component, Figure 10 depicts the MAPE values for the residual component of the load data at each moment during the initial two weeks (336 h). It indicates that, with a few exceptions, the MAPE values of the forecast results generally range from 0 to 0.5, signifying a high level of prediction accuracy.

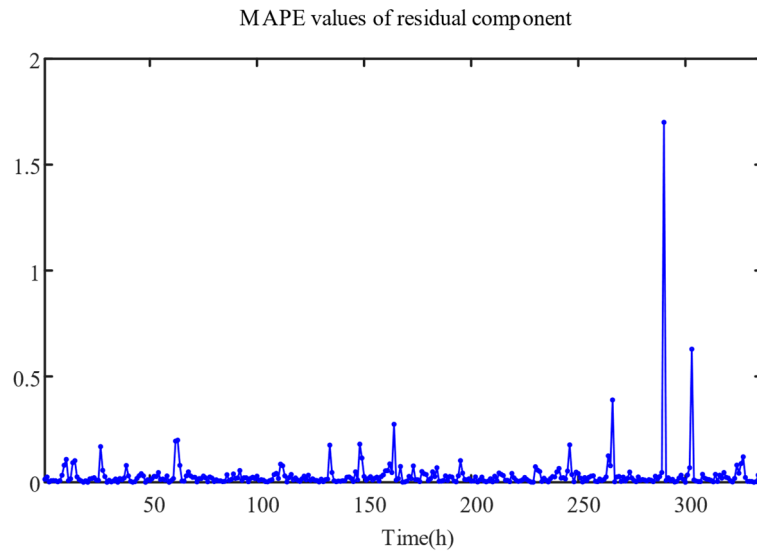


Figure 10. MAPE values of residual component at each timestamp for the previous 336 h.

Based on the aforementioned analysis, the LSTM, CNN, and GPR models all demonstrate excellent performance in predicting their respective components, achieving high accuracy.

3.5. Comparison of Prediction Results and Methods

The prediction results of each component in Section 3.4 are combined to obtain the final result of load prediction. Figure 11 presents the prediction results of the test set load data for the first two weeks (336 h).

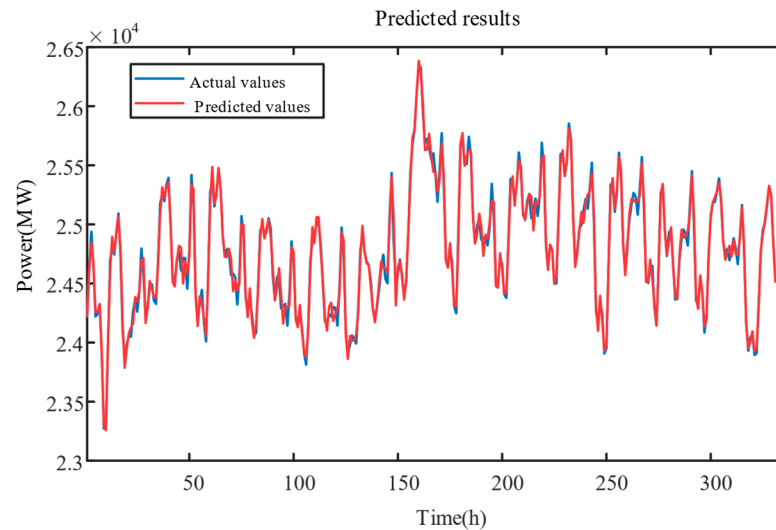


Figure 11. Predicted results of the proposed combined method for the previous 336 h.

To validate the superiority of the proposed combined method, it is compared with multiple prediction methods through experimental comparisons. Initially, the proposed combined method is compared with naive forecasting methods without decomposition, including CNN, LSTM, and GPR. The parameter configurations for each prediction method align with those of the proposed combined method, while the partitioning of the dataset into training and testing sets remains consistent. Figure 12 shows the comparison of the predicted results with naive forecasting methods for the load data for the first three days (72 h) of the testing set. The proposed combined method is depicted in Figure 12, demonstrating a significantly superior level of proximity between the predicted blue curve and the actual red curve compared to other forecasting methods.

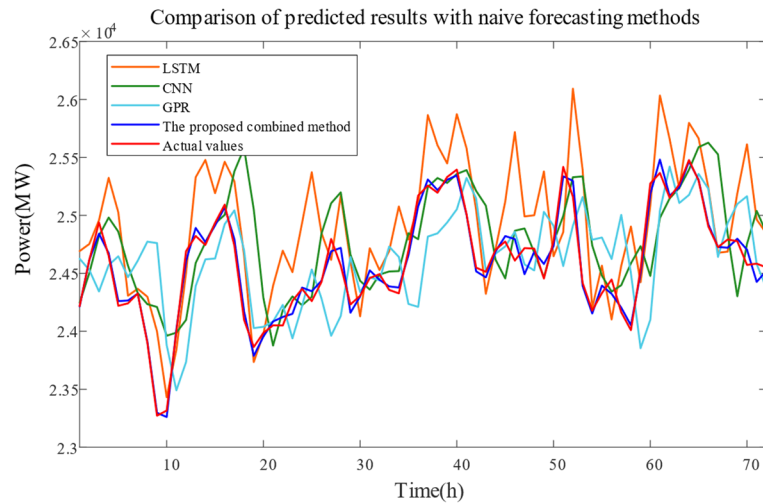


Figure 12. Comparison of predicted results with naive forecasting methods.

To verify the superiority of the proposed combined method in using different prediction methods for different components of STL, the proposed combined method is compared with three other methods that use the same model for predicting the components obtained from STL, namely STL-LSTM, STL-CNN, and STL-GPR. The parameter settings for each prediction method are consistent with those used in the combined prediction method, and the division of the training and testing sets is also consistent. Figure 13 displays the comparison of the prediction results of the load data for the first three days (72 h) of the testing set under each prediction method. It can be observed that all methods in the figure can effectively predict the load, but the predicted values of the proposed combined method are closer to the actual values compared to the other methods.

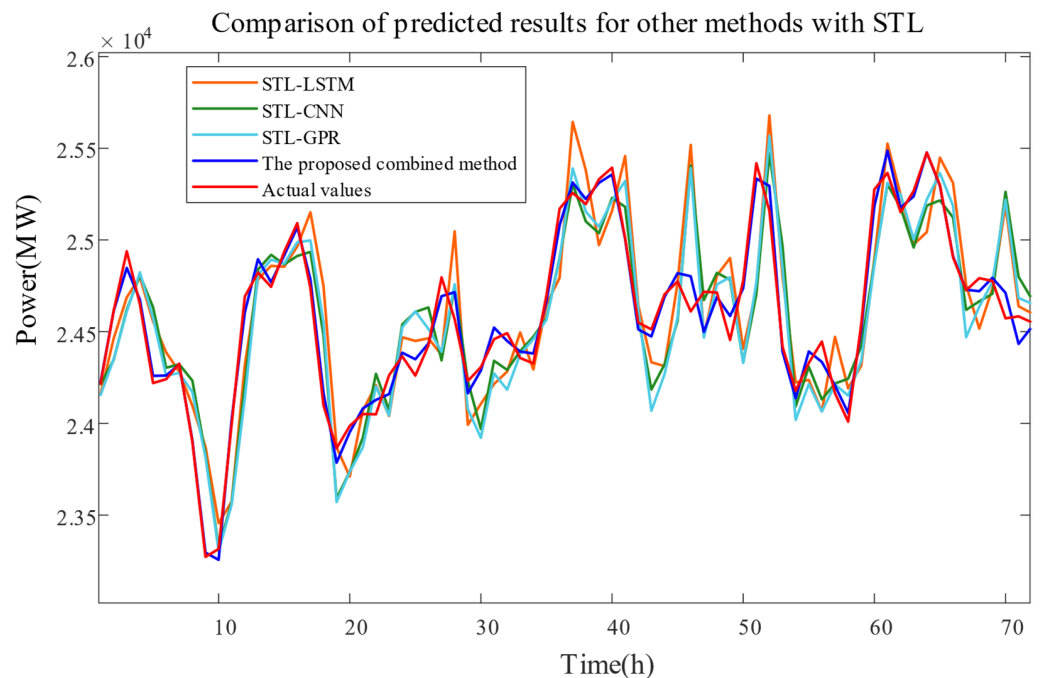


Figure 13. Comparison of predicted results for other methods with STL.

To provide a more intuitive comparison of the results and further validate the superiority of the combined prediction method proposed in this paper, the MAPE, MAE, and RMSE evaluation indicators were used to compare and analyze the various methods mentioned in Figures 12 and 13. The evaluation indicator values for the prediction results of each

method are shown in Table 1. A comparison of the three evaluation indicators in Table 1 reveals that the proposed combined method exhibits lower values for all three evaluation indicators compared to the other methods, indicating higher prediction accuracy.

Table 1. Values of evaluation indicators for prediction results.

Prediction Method	MAPE	MAE (MW)	RMSE (MW)
The proposed combined method	0.14%	151.3506	265.1293
LSTM	2.52%	1421.0175	1756.1287
CNN	1.69%	893.1694	1014.3561
GPR	1.74%	961.3262	1225.6124
STL-LSTM	0.81%	350.3907	432.0894
STL-CNN	0.60%	288.1016	397.1486
STL-GPR	1.05%	521.2655	660.9917

4. Conclusions

The paper proposes a combined method for short-term load forecasting that considers the characteristics of components of STL. The proposed combined method utilizes STL and applies different prediction methods to different components. Through specific case analysis, the following conclusions are drawn:

- (1) Compared to the LSTM, CNN, GPR, STL-LSTM, STL-CNN, and STL-GPR models, the proposed combined method demonstrates higher prediction accuracy, indicating that applying different prediction methods to the characteristics of the STL components can effectively improve prediction accuracy.
- (2) The proposed combined method comprehensively considers the trend, periodic, and random characteristics of the load sequence. It can effectively extract the trend and periodic characteristics of the load sequence while taking into account the uncertainty of load variations.
- (3) The proposed combined forecasting method, which integrates STL with deep learning technologies, has shown considerable improvements in predicting accuracy and stability. Although this method offers advantages over traditional forecasting models, there is still room for enhancement when dealing with the complexity and dynamics of the power system. Therefore, future research will concentrate on integrating a broader range of data sources, including meteorological conditions, economic indicators, and electricity market prices, with the aim of enhancing the model's predictive capabilities for power load fluctuations by incorporating both macro and micro factors. Additionally, the exploration of novel data decomposition techniques and the adoption of more sophisticated machine learning algorithms will be a primary focus of future research aiming to reveal the intrinsic characteristics of data, thereby boosting the accuracy and reliability of predictions. Through thorough research and implementation in these key areas, it is anticipated that not only will the overall performance of short-term power load forecasting be significantly enhanced but also more robust and precise decision support will be provided for the effective and reliable operation of the power system. These efforts will further enhance the adaptability and robustness of forecasting models in response to the intricate dynamics of the power system, establishing a solid foundation for addressing the challenges encountered by future energy systems.

Author Contributions: Conceptualization, S.H. and J.Y.; methodology, C.C.; software, Y.Y. and Y.W.; validation, Y.Z. and Y.G.; formal analysis, C.C.; data curation, S.H. and W.C.; writing—original draft preparation, J.Y.; writing—review and editing, S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Inner Mongolia Electric Power (Group) Co., Ltd., Science and Technology Project 2023-4-6.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Due to the confidentiality requirements of the State Grid company, the research data cannot be disclosed.

Conflicts of Interest: Authors Sile Hu and Yuan Yu were employed by the company Inner Mongolia Electric Power (Group) Co., Ltd. The remaining authors declare that the re-search was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Hong, T.; Fan, S. Probabilistic Electric Load Forecasting: A Tutorial Review. *Int. J. Forecast.* **2016**, *32*, 914–938. [[CrossRef](#)]
2. Kim, N.; Park, H.; Lee, J.; Choi, J.K. Short-Term Electrical Load Forecasting with Multidimensional Feature Extraction. *IEEE Trans. Smart Grid* **2022**, *13*, 2999–3013. [[CrossRef](#)]
3. Montero-Manso, P.; Hyndman, R.J. Principles and Algorithms for Forecasting Groups of Time Series: Locality and Globality. *Int. J. Forecast.* **2021**, *37*, 1632–1653. [[CrossRef](#)]
4. Kuster, C.; Rezgui, Y.; Mourshed, M. Electrical Load Forecasting Models: A Critical Systematic Review. *Sustain. Cities Soc.* **2017**, *35*, 257–270. [[CrossRef](#)]
5. Zendejboudi, A.; Baseer, M.A.; Saidur, R. Application of Support Vector Machine Models for Forecasting Solar and Wind Energy Resources: A Review. *J. Clean. Prod.* **2018**, *199*, 272–285. [[CrossRef](#)]
6. Cai, C.; Tao, Y.; Zhu, T.; Deng, Z. Short-Term Load Forecasting Based on Deep Learning Bidirectional LSTM Neural Network. *Appl. Sci.* **2021**, *11*, 8129. [[CrossRef](#)]
7. Abumohsen, M.; Owda, A.Y.; Owda, M. Electrical Load Forecasting Using LSTM, GRU, and RNN Algorithms. *Energies* **2023**, *16*, 2283. [[CrossRef](#)]
8. Hu, L.; Wang, J.; Guo, Z.; Zheng, T. Load Forecasting Based on LVMD-DBFCM Load Curve Clustering and the CNN-IVIA-BLSTM Model. *Appl. Sci.* **2023**, *13*, 7332. [[CrossRef](#)]
9. Quilumba, F.L.; Lee, W.-J.; Huang, H.; Wang, D.Y.; Szabados, R.L. Using Smart Meter Data to Improve the Accuracy of Intraday Load Forecasting Considering Customer Behavior Similarities. *IEEE Trans. Smart Grid* **2015**, *6*, 911–918. [[CrossRef](#)]
10. Li, S.; Wang, P.; Goel, L. Short-Term Load Forecasting by Wavelet Transform and Evolutionary Extreme Learning Machine. *Electr. Power Syst. Res.* **2015**, *122*, 96–103. [[CrossRef](#)]
11. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.C.; Tung, C.C.; Liu, H.H. The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-Stationary Time Series Analysis. *Proc. R. Soc. London. Ser. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [[CrossRef](#)]
12. Semero, Y.K.; Zhang, J.; Zheng, D. EMD-PSO-ANFIS-Based Hybrid Approach for Short-Term Load Forecasting in Microgrids. *IET Gener. Transm. Distrib.* **2020**, *14*, 470–475. [[CrossRef](#)]
13. He, F.; Zhou, J.; Feng, Z.K.; Liu, G.; Yang, Y. A Hybrid Short-Term Load Forecasting Model Based on Variational Mode Decomposition and Long Short-Term Memory Networks Considering Relevant Factors with Bayesian Optimization Algorithm. *Appl. Energy* **2019**, *237*, 103–116. [[CrossRef](#)]
14. Li, W.; Shi, Q.; Sibtain, M.; Li, D.; Mbanze, D.E. A Hybrid Forecasting Model for Short-Term Power Load Based on Sample Entropy, Two-Phase Decomposition and Whale Algorithm Optimized Support Vector Regression. *IEEE Access* **2020**, *8*, 166907–166921. [[CrossRef](#)]
15. Fang, Z.; Zhan, J.; Cao, J.; Gan, L.; Wang, H. Research on Short-Term and Medium-Term Power Load Forecasting Based on STL-LightGBM. In Proceedings of the 2022 2nd International Conference on Electrical Engineering and Control Science (IC2ECS), Nanjing, China, 16–18 December 2022; pp. 1047–1051. [[CrossRef](#)]
16. Lin, C.; Weng, K.; Lin, Y.; Zhang, T.; He, Q.; Su, Y. Time Series Prediction of Dam Deformation Using a Hybrid STL-CNN-GRU Model Based on Sparrow Search Algorithm Optimization. *Appl. Sci.* **2022**, *12*, 11951. [[CrossRef](#)]
17. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
18. Hong, Y.-Y.; Apolinario, G.F.D.; Cheng, Y.-H. Week-ahead Daily Peak Load Forecasting Using Hybrid Convolutional Neural Network. *IFAC-PapersOnLine* **2023**, *56*, 372–377. [[CrossRef](#)]
19. Imani, M. Electrical Load-Temperature CNN for Residential Load Forecasting. *Energy* **2021**, *227*, 120480. [[CrossRef](#)]
20. Darab, C.; Antoniu, T.; Beleiu, H.G.; Pavel, S.; Birou, I.; Micu, D.D.; Ungureanu, S.; Cirstea, S.D. Hybrid Load Forecasting Using Gaussian Process Regression and Novel Residual Prediction. *Appl. Sci.* **2020**, *10*, 4588. [[CrossRef](#)]

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