

Article

Bayesian-Optimized GCN-BiLSTM-Adaboost Model for Power-Load Forecasting

Jiarui Li ¹, Jian Li ², Jiatong Li ¹ and Guozheng Zhang ^{2,*}

¹ School of Electrical Engineering, Tiangong University, Tianjin 300387, China; 18892279037@163.com (J.L.); iamagift2@163.com (J.L.)

² College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China; 2023021063@cauc.edu.cn

* Correspondence: zhangguozheng@tiangong.edu.cn

Abstract

Accurate and stable power-load forecasting is crucial for optimizing generation scheduling and ensuring the economic and secure operation of power grids. To address the issues of low prediction accuracy and poor robustness during abrupt load changes, this study proposes a Bayesian-optimized GCN-BiLSTM-Adaboost model (abbreviated as GCN-BiLSTM-AB). It combines Graph Convolutional Networks (GCN), Bidirectional Long Short-Term Memory Networks (BiLSTM), and a Bayesian-optimized AdaBoost framework. Firstly, the GCN is employed to capture the spatial correlation features of the input data. Then, the BiLSTM is employed to extract the long-term dependencies of the data time series. Finally, the AdaBoost framework is used to dynamically adjust the base learner weights, and a Bayesian method is employed to optimize the weight adjustment process and prevent overfitting. The experiment results on actual load data from a regional power grid show the GCN-BiLSTM-AB outperforms other compared models in prediction error metrics, with *MAE*, *MAPE*, and *RMSE* values of 1.86, 3.13%, and 2.26, respectively, which improve the prediction robustness during load change periods. Therefore, the proposed method shows that the synergistic effect of spatiotemporal feature extraction and dynamic weight adjustment improves prediction accuracy and robustness, which provides a new forecasting model with high precision and reliability for power system dispatch decisions.

Keywords: power-load forecasting; graph convolutional network (GCN); bidirectional long short-term memory networks (BiLSTM); Bayesian-optimized Adaboost



Academic Editor: Jen-Hao Teng

Received: 4 July 2025

Revised: 19 August 2025

Accepted: 20 August 2025

Published: 21 August 2025

Citation: Li, J.; Li, J.; Li, J.; Zhang, G. Bayesian-Optimized GCN-BiLSTM-Adaboost Model for Power-Load Forecasting. *Electronics* **2025**, *14*, 3332. <https://doi.org/10.3390/electronics14163332>

Copyright: © 2025 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

With the construction of smart grids [1] and the large-scale integration of renewable energy [2], the evolving power system is facing multiple challenges, including the continuous rise in electricity demand, the integration and utilization of renewable energy, and the urgent need to optimize system operation efficiency. Safe, reliable, and cost-effective power system operations are critical in global energy strategies. Accurate electricity demand forecasting plays a vital role in achieving these goals. The accuracy of power-load forecasting directly affects the economic efficiency and safety of power system dispatch decisions [3]. High-precision load forecasting is essential for guaranteeing the safe operation of the grid, maintaining system stability, and enhancing operational benefits [4]. Furthermore, accurate load forecasting can provide reliable data support for grid scheduling, effectively preventing potential risks of power shortages or surpluses, thereby ensuring voltage and frequency stability and improving power supply quality.

Power-load forecasting accuracy depends on various factors, including socioeconomic variables like population distribution and industrial energy efficiency, as well as environmental factors such as climate conditions [3,4]. Among these, complex and variable meteorological factors are key elements in load forecasting and have the greatest impact on prediction results [5]. Considering meteorological factors in forecasting can improve prediction accuracy.

In recent years, the frequent occurrence of extreme weather events has induced non-stationary abrupt changes in load curves. Consequently, numerous researchers have been conducting in-depth studies on the impacts of meteorological factors on power load and the characterization of spatiotemporal relationships among factors influencing power load. Scientifically analyzing and capturing these spatiotemporal relationships is crucial for enhancing the accuracy and stability of load forecasting, as well as for providing rational scheduling strategies.

The complex working conditions faced by modern power dispatch pose higher requirements for load forecasting. To address this, academia and industry have continuously improved the accuracy and robustness of forecasting models through diverse methodological innovations. The accuracy of load forecasting is determined by data quality, model performance, and load characteristics, and these interrelated factors have a synergistic effect on the prediction results. Traditional artificial intelligence methods have limitations in handling massive time-series data and high-dimensional multi-scale features. Their inherent gradient issues and overfitting phenomena can reduce the reliability of power forecasting [6]. Deep neural networks (such as CNN, LSTM, GRU, Transformer, etc.) have gradually become a new research hotspot in the field of improving power-load forecasting accuracy [7]. Li et al. [8] proposed a novel deep learning-based forecasting (DLSF) method, utilizing a CNN network model to accurately cluster input data and demonstrating its excellent performance in both accuracy and efficiency. Muzaffar et al. [9] applied LSTM networks to power-load forecasting and compared them with traditional methods, demonstrating the significant potential of LSTM in improving forecasting accuracy. L'Heureux et al. [10] optimized the model workflow, designed an N-dimensional space transformation layer, and innovated context feature extraction methods, ultimately developing an improved Transformer architecture specifically for load forecasting. However, it should be noted that a single intelligent forecasting model has inherent limitations. Traditional time-series analysis methods can lead to information loss when the input sequence is too long [11]. Based on this, Graves [12] proposed the bidirectional LSTM network, which can not only capture the temporal patterns of historical load data (such as daily periodic fluctuations) but also reversely learn the potential impact of future moments on the current state (such as pre-load adjustments before holidays), thus extracting temporal dependency features more comprehensively. Additionally, Graph Convolutional Networks (GCN) show marked superiority over conventional spatial feature extraction techniques, owing to their distinctive topological modeling proficiency. Chen et al. [13] used a combination of GCN and LSTM to extract spatiotemporal features from large-scale data for power-load forecasting and validated the effectiveness of the model. Model specificity leads to boundary constraints on the engineering applicability of single models, making it urgent to develop adaptive hybrid modeling methods [14].

Model ensemble-based forecasting methods provide an effective way to overcome the limitations of single models. This theoretical framework was first established by Bates and Granger in 1969, and their pioneering work laid the theoretical foundation for the subsequent development of combined forecasting techniques, continuously advancing research in this field [15]. In power-load forecasting research, accurately capturing spatiotemporal features is key to improving prediction accuracy. Ma et al. [16] proposed a

novel combined model that integrates Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) networks, effectively extracting spatiotemporal features from the data and incorporating an attention mechanism to improve the accuracy of power-load forecasting. Bouktif et al. [17] proposed combining RNN networks with LSTM networks and incorporating genetic algorithms (GA) and particle swarm optimization to improve the prediction accuracy of the model.

On the other hand, existing power-load forecasting models still suffer from the following limitations: (1) Inadequate capture of the spatiotemporal characteristics of data, which may degrade the model's prediction accuracy. (2) A lack of adaptive mechanisms for dynamically allocating weights to base learners, resulting in a significant decline in prediction robustness during periods of load fluctuation (e.g., extreme weather events). (3) Challenges in quantifying the impact of model uncertainty on prediction outcomes. Although researchers have applied various algorithms to mitigate these drawbacks, gaps persist in the existing literature.

To solve this, this paper introduces a GCN-BiLSTM hybrid model within the AdaBoost framework, further enhanced through Bayesian optimization:

(1) Construct a graph structure with meteorological factors as nodes, where each node represents different meteorological elements such as temperature and humidity, and use the GCN layer to extract nonlinear spatial correlation features among these factors.

(2) The BiLSTM network is responsible for learning the temporal dynamic characteristics of load and meteorological factors.

(3) A Bayesian optimization method based on Markov Chain Monte Carlo (MCMC) sampling is used to achieve probabilistic hyperparameter tuning and uncertainty quantification of prediction results.

(4) An Adaboost ensemble mechanism is introduced to dynamically adjust the weights of multiple GCN-BiLSTM weak learners based on uncertainty obtained from Monte Carlo sampling, with a focus on enhancing prediction robustness under extreme weather events.

Finally, by comparing the GCN-BiLSTM-AB model with single BiLSTM, GCN-BiLSTM, and GCN-BiLSTM-Adaboost models, the proposed method is validated in terms of prediction accuracy, computational efficiency, and reliability of probabilistic outputs, providing a more reliable basis for power system scheduling decisions.

The structure of this paper is organized as follows: Section 2 introduces the GCN-BiLSTM-AB hybrid model, which integrates Graph Convolutional Networks (GCN) for spatial feature extraction, Bidirectional Long Short-Term Memory (BiLSTM) networks for temporal learning, and a Bayesian-optimized AdaBoost framework. Section 3 elaborates on the experimental setup, including data preprocessing procedures and evaluation metrics. Section 4 assesses the model's performance against baseline models using quantitative indicators (mean absolute error [MAE], mean absolute percentage error [MAPE], root mean square error [RMSE]) and robustness tests under extreme load conditions; it also includes an analysis of the graph construction method based on the Spearman correlation coefficient. Section 5 summarizes the key findings, discusses their practical implications for power grid operations, and elaborates on the limitations of the proposed model as well as potential future improvement strategies.

2. Method

2.1. Model Framework

The power-load forecasting research framework proposed in this paper involves a specific process, as shown in Figure 1. The model uses GCN to capture spatial dependencies and BiLSTM to extract long-term temporal dependencies in power system features. The GCN-BiLSTM model serves as the base learner, where AdaBoost integrates 10 such weak

learners by dynamically adjusting their weights based on prediction errors. A Bayesian approach is introduced to quantify prediction uncertainty through Monte Carlo sampling, which simultaneously prevents overfitting. The weights of weak learners are further refined according to their uncertainty estimates. Ultimately, the weighted aggregation of all 10 learners' predictions yields the final output, significantly enhancing both predictive stability and accuracy.

The research framework for short-term power-load forecasting involves a specific process, including data preprocessing and GCN-BiLSTM-Adaboost (with the introduction of Bayesian methods), as shown in Figure 1:

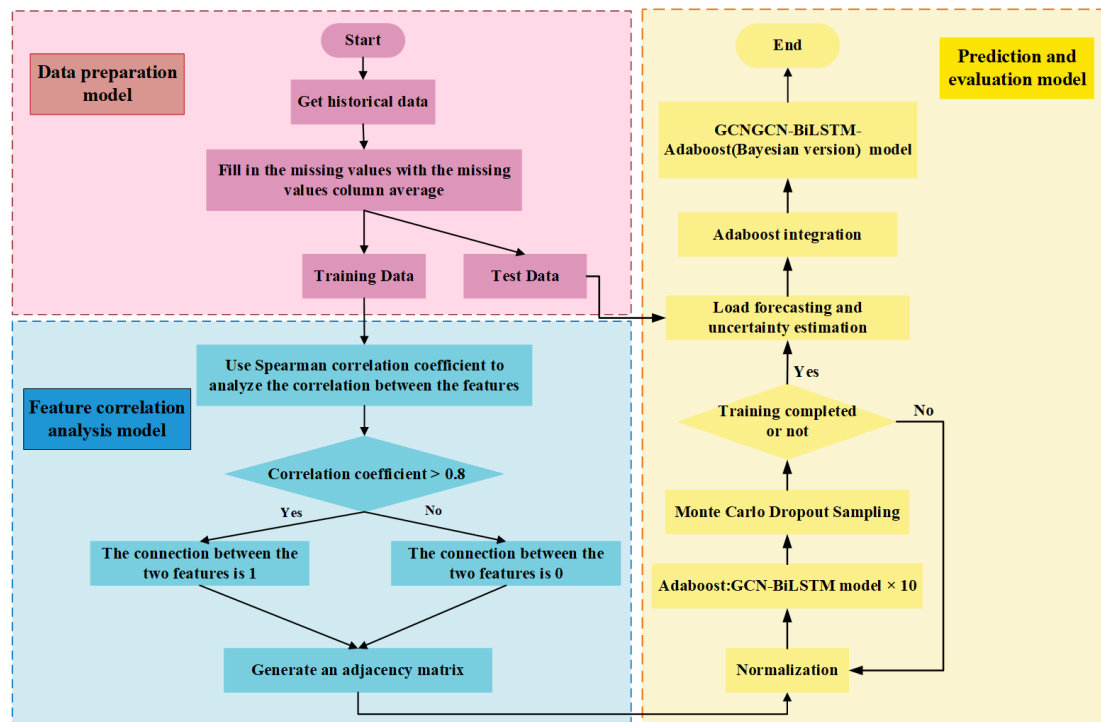


Figure 1. Model flow chart.

(1) Data preprocessing:

The input dataset comprises 9 feature columns, including load value, humidity, temperature, wind speed, atmospheric pressure, precipitation, visibility, water vapor pressure, and perceived temperature.

For handling missing values, the mean value of the respective column containing the missing data is employed for imputation.

Subsequently, the data are normalized, and a sliding window (with a 24-h window size) is constructed to predict the load value for the subsequent hour.

(2) Feature correlation analysis and adjacency matrix construction:

The Spearman correlation coefficient is employed to analyze the correlation between input features in the dataset. Within the adjacency matrix of the GCN, a correlation coefficient exceeding 0.8 is defined as an edge (i.e., a connection weight of 1) between two features, while all other correlations are assigned a weight of 0. This approach is used to construct the adjacency matrix.

(3) Prediction and model evaluation:

GCN-BiLSTM is used to capture the spatial and temporal dependency relationships in the dataset.

Dropout is used to randomly discard some neurons during the training phase to prevent model overfitting.

Adaboost is utilized to integrate the results of multiple GCN-BiLSTM weak predictors, with the weight of each weak predictor adjusted based on its prediction error and uncertainty.

Bayesian methods are realized via Monte Carlo Dropout to estimate the uncertainty of prediction outcomes.

2.2. GCN Method

The core idea of Graph Convolutional Networks (GCN) is to aggregate information from nodes and their neighboring nodes through graph convolution operations [18]. In this study, the nodes within the GCN layer represent the 8-dimensional feature vectors corresponding to meteorological factors at each time step, while the edges between nodes denote the connectivity relationships among these features. The primary objective of the GCN is to extract the interdependencies between features in the dataset.

Graph data can be represented as $G = \{X, E, A\}$. In this case, X represents the set of nodes, E represents the set of edges, and A is the adjacency matrix. This paper determines the connection relationships between nodes in GCN by calculating the correlation relationships between input features [19], the detailed analysis process is presented in Section 4.1. Since there is no obvious normal distribution relationship or linear correlation between the features, the Spearman correlation coefficient is chosen to analyze the relationships between the features [20].

Assuming all negative correlation coefficients are taken as absolute values, a Spearman correlation coefficient absolute value less than 0.8 is typically interpreted as indicating weak or no correlation between two variables. Conversely, when the absolute value of the Spearman correlation coefficient is greater than or equal to 0.8, a connection relationship exists between the two nodes. The value of Spearman's correlation coefficient is shown in Figure 2.

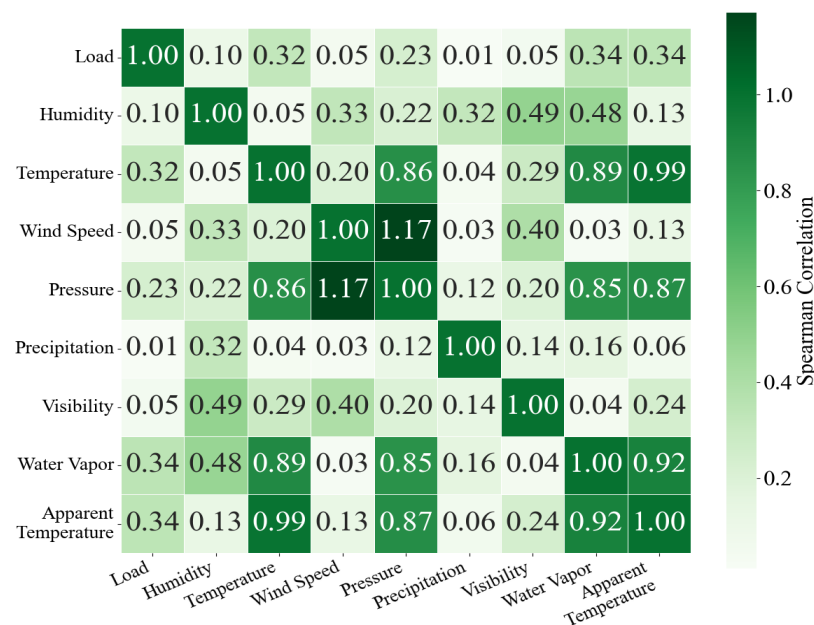


Figure 2. Spearman correlation coefficient between features.

The GCN layer can be represented by the following equation:

$$H^{(1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(0)} W^{(0)}\right) \quad (1)$$

where $H^{(1)}$ is the input feature matrix, $W^{(0)}$ is the weight matrix, $\tilde{A}$ is the adjacency matrix with self-loops added, $\tilde{D}$ is the degree matrix, $\sigma(\cdot)$ is the nonlinear activation function.

GCN enhances spatial feature representation and can better capture the complex relationships between meteorological factors. The output of the GCN is passed to BiLSTM (Bidirectional Long Short-Term Memory) to capture the long-term dependencies in the time series. The structure of GCN is shown in Figure 3.

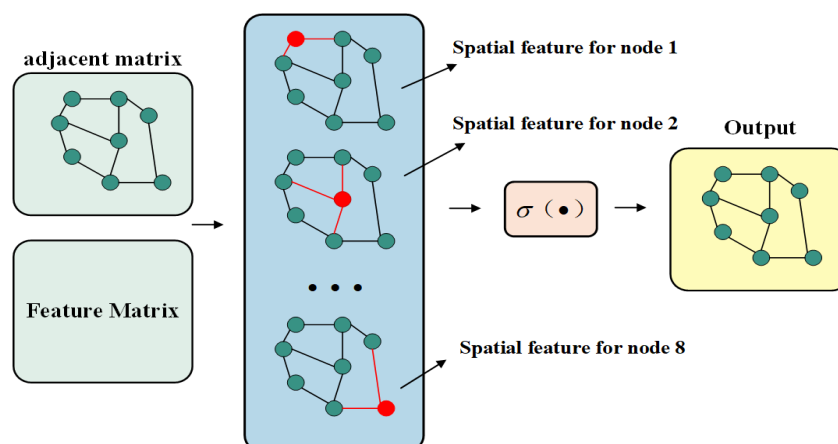


Figure 3. GCN structure.

2.3. BiLSTM Method

The cited paper [21] mentions that Long Short-Term Memory (LSTM) was proposed by Hochreiter S and Schmidhuber J. By introducing gating mechanisms (input gate, forget gate, and output gate), LSTM addresses the vanishing gradient problem of traditional RNNs [22]. The core formula is as follows:

(1) The forget gate determines what information needs to be forgotten from the cell state C_{t-1} at time $t - 1$, as shown in Equation (2). The forget gate reads the hidden layer state h_{t-1} and input sequence x_t at time t , and outputs a value between 0 and 1, where 1 means retaining all information, and 0 means discarding all information.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

where f_t is the forgetting gate state at time t ; W_f and b_f are the remaining relics, respectively, forget the weight and bias of the gate; σ is the bipolar sigmoid activation function.

(2) The input gate reads the input x_t at time t and decides the information to be saved in the neurons. Then, the tanh layer generates a temporary state for the memory unit at time t . Finally, the cell state is updated to obtain the new cell state C_t , as shown in Equations (3)–(5).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (4)$$

$$C_t = (f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t) \quad (5)$$

where i_t is the input gate state at time t , and the control x_t is passed to C_t amount of information; W_i and b_i are the weight and bias of the input gate, respectively. W_c and b_c are the weight matrix and bias term of the cell state, respectively. $\tanh$ is a double curved tangent activation function, $\otimes$ is the Hadamard product.

(3) The output gate can choose important information to be output from the current state. The sigmoid layer first decides which part of the neuron state needs to be output. The neuron state to be output passes through the tanh layer and is multiplied by the output

of the sigmoid layer to obtain the output value h_t , which is also the input value for the next hidden layer, as shown in Equations (6) and (7).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t \otimes \tanh(C_t) \quad (7)$$

where o_t is the output gate state at time t , W_o and b_o are the weight matrix and bias term for the output gate, respectively.

Its structure is shown in Figure 4:

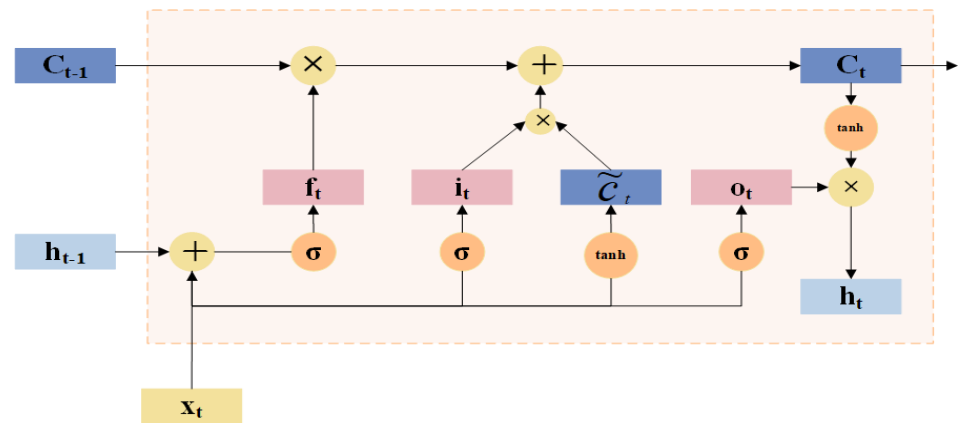


Figure 4. LSTM structure.

While unidirectional LSTM effectively leverages historical load data, it is important to note that it does not consider future information in its predictions.

Bidirectional Long Short-Term Memory (BiLSTM) network differs from traditional LSTM in that it simultaneously takes into account both the forward and backward information of the sequence, enabling a more comprehensive capture of long-term dependencies in the time series. The BiLSTM structure is shown in Figure 5.

In this paper, BiLSTM can effectively capture long-distance dependencies between time steps, improving prediction accuracy. The final output is the concatenation of the forward and backward LSTM outputs:

$$h_t = [\vec{h}_t, \overleftarrow{h}_t] \quad (8)$$

where $\vec{h}_t$ is the output of the forward LSTM, and $\overleftarrow{h}_t$ is the output of the backward LSTM.

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}) \quad (9)$$

$$\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t-1}) \quad (10)$$

The LSTM unit represents the computation process of the unidirectional LSTM network. $\vec{h}_t$ is the forward LSTM hidden state at time step t . $\overleftarrow{h}_t$ is the backward LSTM hidden state at time step t . W_y and b_y are the weight matrix and bias term, respectively. $x_1, x_2, x_3, \dots, x_t$ represent $t_1 \sim t_i$ the input data at each time step ($i \in [1 \sim t]$). $y_1, y_2, y_3, \dots, y_t$ represent $t_1 \sim t_i$ the output data at each time step ($i \in [1 \sim t]$).

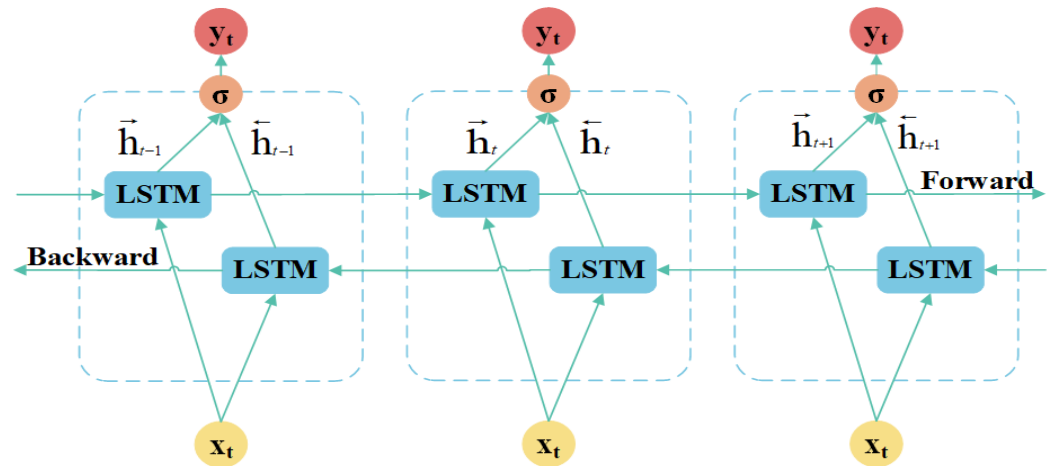


Figure 5. BiLSTM structure.

2.4. AdaBoost Algorithm

Adaboost (Adaptive Boosting) is an ensemble learning method that builds a strong predictor by combining multiple weak predictors. The fundamental principle of Adaboost lies in iteratively training weak learners and modifying sample weights according to each learner's performance, enabling subsequent learners to concentrate on previously misclassified instances. Ultimately, Adaboost integrates the results of all weak predictors through weighted voting to produce the final prediction results [23,24].

In this study, the GCN-BiLSTM hybrid model acts as a weak learner. The AdaBoost framework is introduced to integrate 10 such weak learners. Based on prediction results, it dynamically adjusts the sample weights within the weak learners and the weights of the weak learners themselves. This enhances the accuracy and robustness of the final predictions. The specific steps are as follows:

Step 1: Initialize sample weights, with each sample assigned an equal initial weight, $D_1(i) = \frac{1}{m}$, where m is the number of training samples (1200), and i refers to the i -th sample.

Step 2: Train the k -th weak learner using the current sample weights $D_k(i)$, and for each sample i , calculate the current prediction error e_i .

$$e_i = |\hat{y}_i - y_i| \quad (11)$$

where y_i is the true value, $\hat{y}_i$ is the predicted value.

Step 3: Calculate the weight adjustment factor weight_sum and accumulate the weights of samples with large errors:

$$\text{weight_sum} = \sum_{i|e_i > 0.3} D_k(i) \quad (12)$$

Calculate the weight of the weak learner α_k using an exponential decay formula: The weight of the current weak learner is calculated, which is related to its performance in training, and weak learners with good performance will be assigned higher weights.

$$\alpha_k = \frac{0.5}{\exp(\text{weight_sum})} \quad (13)$$

Step 4: Update sample weights. Samples with large prediction errors ($e_i > 0.3$) will be assigned higher weights to receive more attention in the next iteration.

$$D_{k+1}(i) = D_k(i) \times 1.1 \quad (14)$$

For other samples, keep the weights unchanged:

$$D_{k+1}(i) = D_k(i) \quad (15)$$

Normalize the sample weights so that the sum of the sample weights is 1:

$$D_{k+1}(i) \leftarrow \frac{D_{k+1}(i)}{\sum_i D_{k+1}(i)} \quad (16)$$

The weights of weak learners do not need to be normalized here, because a Bayesian method will be introduced in the subsequent testing phase to further adjust the weights of weak learners, and then the normalization of weak learner weights will be performed.

The structure of the algorithm is shown in Figure 6.

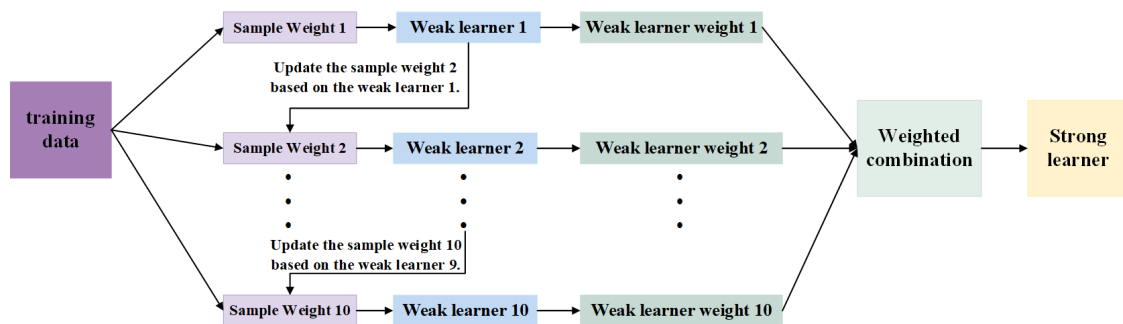


Figure 6. Adaboost structure.

It is worth emphasizing that the AdaBoost algorithm employed in this study has been modified from the traditional AdaBoost framework, with the specific improvements outlined as follows:

(1) The proposed framework directly models spatiotemporal features using GCN-BiLSTM, which enables more effective capture of complex patterns compared to the decision trees or shallow models utilized in traditional AdaBoost.

(2) Unlike the traditional AdaBoost algorithm, which adjusts the weights of all misclassified samples uniformly, this study only increases the weights of samples with prediction errors exceeding 0.3. This modification allows subsequent weak learners to focus on learning critical hard-to-predict samples (e.g., peak electricity consumption periods, load mutations induced by extreme weather) while omitting weight adjustments for samples with minor errors. This avoids excessive model sensitivity to noise or non-significant fluctuations. It should be noted that the threshold value of 0.3 should be adjusted based on error tolerance criteria and data characteristics in practical applications.

(3) The introduction of an “uncertainty” mechanism for further refinement of weak learner weights will be elaborated in Section 2.5.

In summary, these modifications to the traditional Adaboost algorithm enhance the model’s sensitivity to sudden changes and periodic variations in load, while mitigating overfitting during stable load periods.

2.5. Bayesian Method

Owing to data uncertainty (i.e., errors induced by inherent noise in the dataset) and model uncertainty (i.e., uncertainty stemming from insufficient training data or inadequate model interpretability of data patterns), this study incorporates Bayesian methods to quantify prediction uncertainty. It should be noted that the “prediction uncertainty” in this study specifically refers to the variance of predictions from each GCN-BiLSTM weak learner.

The Bayesian method is a probability-based statistical inference technique that calculates the posterior distribution for prediction and uncertainty estimation by incorporating prior distributions and observational data [25]. In this study, the Bayesian method is realized using Monte Carlo Dropout. Monte Carlo Dropout is an approximation to Bayesian inference that estimates the uncertainty of predictions by repeatedly applying Dropout during the testing phase to simulate the posterior distribution of the model parameters [26].

Specifically, the Bayesian approach is implemented in two distinct phases. During the training phase, Dropout is employed to randomly deactivate neurons in each forward pass, thereby mitigating overfitting. In the testing phase, each GCN-BiLSTM weak learner generates 100 stochastic predictions y_n through Monte Carlo Dropout sampling, from which we compute both the mean prediction $\bar{y}$ and variance—the latter serving as the quantitative uncertainty measure for the weak learner.

$$\text{Uncertainty} = \frac{1}{100} \sum_{n=1}^{100} (y_n - \bar{y})^2 \quad (17)$$

Each weak learner adjusts its training-derived weight W_m (computed during the AdaBoost training phase) based on its prediction uncertainty, as specified in Equation (18). Concretely, weak learners demonstrating higher predictive uncertainty undergo weight attenuation, whereas those with more certain predictions obtain magnified weighting. (High uncertainty means high variance. This shows the weak learner's predictions are unstable, so we reduce its weight in the ensemble).

$$W'_m = \frac{W_m}{1 + \text{uncertainty}} \quad (18)$$

To ensure the weights sum to unity, a normalization step is applied to the weights W'_m as shown in Equation (19), yielding the final weak learner weights W''_m .

$$W''_m = \frac{W'_m}{\sum_{m=1}^{10} W'_m} \quad (19)$$

The final prediction is obtained by computing the weighted sum of the weak learners' predictions.

The advantages of introducing the Bayesian method are as follows:

- (1) Quantifying uncertainty, improving model reliability.
- (2) Introducing uncertainty weighting in AdaBoost, enhancing the stability of final predictions.
- (3) Obtaining a more reasonable predictive distribution by calculating the mean and variance through multiple samplings.

3. Data Processing

3.1. Data Description

The dataset utilized in this experiment is derived from hourly electricity load data collected over one year in a specific region (spanning from 00:00 on 1 January 2018, to 23:00 on 28 December 2018), with detailed information presented in Table 1.

The input dataset comprises ten columns, where the labels for columns 1 to 10 are as follows: date (formatted as year-month-day hour), load value, humidity, temperature, wind speed, atmospheric pressure, precipitation, visibility, water vapor pressure, and perceived temperature.

Table 1. Data example.

Date	Load (MW)	Humidity (%)	Temperature (°C)	Pressure (Pa)
1 January 2018 0:00	174	53	−3.6	1024.4
Wind Speed (m/s)	Precipitation (mm)	Visibility (km)	Vapor Pressure (hPa)	Sensible Temperature (°C)
0.4	0	5.25	2.48	−6.32

3.2. Data Standardization

Data normalization constitutes a critical preprocessing step for optimizing the model training process, with its core value manifested in three aspects: accelerating the convergence of the objective function optimization trajectory, suppressing gradient anomalies during parameter updates, and enhancing the stability of numerical computations. In practical applications of power-load forecasting, model input features frequently exhibit heterogeneous dimensional characteristics (e.g., unit differences between temperature, humidity, and load values). To eliminate the interference of inter-feature dimensional discrepancies on the prediction model, while improving the algorithm's generalization capability and computational efficiency, the original data must undergo scale normalization. This study employs the min-max normalization method, whereby each feature is linearly mapped to the unit interval [0, 1]. Its mathematical expression is as follows:

$$X_i^* = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \quad (20)$$

where X_i represents the original measurement value of the sampling point, X_i^* represents the normalized value, and $X_{\max}$, $X_{\min}$ are the maximum and minimum values in the actual data. After inputting the normalized data into the model and obtaining the normalized prediction result, the result is re-normalized using Equation (16).

$$x^* = y^*(x_{\max} - x_{\min}) + x_{\min} \quad (21)$$

where y^* is the normalized power-load prediction value, x^* is the de-normalized power-load prediction value, and $x_{\max}$ and $x_{\min}$ are the maximum and minimum values in the actual data.

3.3. Evaluation Index

To verify the effectiveness of the model and evaluate the prediction performance, this paper selects Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE) as evaluation metrics. The formulas for these metrics are provided in Equations (17)–(19):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (22)$$

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (23)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (24)$$

where y_i represents the actual value, $\hat{y}_i$ represents the predicted value, and n represents the number of samples. The smaller the values of MAE , $MAPE$, and $RMSE$, the more accurate the load prediction.

4. Case Study

In this section, the advantages of the Spearman correlation coefficient-based graph construction method are verified by comparing it with several alternative graph construction methods, including k-nearest neighbors, learned graphs, and mutual information. Additionally, the effectiveness of the proposed model is validated from the following two perspectives:

(1) The prediction accuracy of the proposed model is compared with that of baseline models (LSTM, GRU, CNN-LSTM, GCN-LSTM, and CNN-BiLSTM) over one-day and one-week forecasting horizons. The results demonstrate its distinct advantage in prediction accuracy.

(2) By comparing the load forecasting performance of BiLSTM, GCN-BiLSTM, GCN-BiLSTM-Adaboost, and the proposed model over one-day and one-week horizons, the contributions of AdaBoost integration and Bayesian methods to enhancing the model's prediction stability are validated.

The aforementioned forecasting performance is evaluated using three metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error ($MAPE$), and Root Mean Square Error ($RMSE$).

4.1. Comparison of Methods for Constructing Graph Structures

There are many methods for constructing graphic structures [27], like k-nearest neighbors (KNN) [28], learned graphs [29], mutual information [30] and so on. This paper compares the one-week prediction results of the Spearman correlation coefficient with several graph construction methods, including k-nearest neighbors, learned graphs, and mutual information, as shown in Table 2, verifying the superiority of using the Spearman correlation coefficient for graph construction.

Table 2. Performance comparison.

Model	A Week		
	MAE	$MAPE$	$RMSE$
Spearman	0.34	0.68%	0.43
KNN	0.42	0.82%	0.55
Learned Graphs	0.49	0.97%	0.65
Mutual Information	0.42	0.81%	0.54

As can be seen from Table 2, compared with KNN, the graph construction method using Spearman correlation coefficient has improved MAE , $MAPE$, and $RMSE$ by 0.08, 0.14%, and 0.12, respectively. Compared with Learned Graphs, MAE , $MAPE$, and $RMSE$ have increased by 0.15, 0.29%, and 0.22, respectively. Compared with Mutual Information, MAE , $MAPE$, and $RMSE$ have increased by 0.08, 0.13%, and 0.11, respectively.

The k-nearest neighbors (KNN) method connects features solely based on the numerical similarity of meteorological data, which may easily introduce spurious correlations. In contrast, the Spearman correlation coefficient identifies genuine correlations among meteorological features through statistical relevance, and the resulting graph structure is more consistent with actual physical laws. This enables it to more accurately reflect the combined impact of meteorological factors on load.

Learned graphs require large volumes of data and complex training processes, making them prone to overfitting. By comparison, the Spearman-based approach directly constructs graph structures based on the inherent correlations of meteorological factors, eliminating the need for training. It features simple computation and yields stable, reliable results.

Mutual information is sensitive to data distribution and sample size, often leading to estimation biases in small datasets. In contrast, the Spearman correlation coefficient is computed via simple rank statistics and can reliably capture monotonic relationships, regardless of data distribution characteristics.

4.2. Hyperparameter Settings

The proposed GCN-BiLSTM-AB-based power-load forecasting model was implemented and optimized using the PyTorch framework in Python. All experiments in this study were conducted on a workstation equipped with an NVIDIA GeForce RTX 4060Ti GPU (Nvidia Corporation, Santa Clara, CA, USA), a 13th Generation Intel® Core™ i7 CPU, and 32 GB of RAM (Intel Corporation, Santa Clara, CA, USA). The Adam optimization algorithm was employed for network parameter updates, with an initial learning rate set to 0.001. The deep learning model was trained over 1800 epochs.

The detailed structural parameters of the model are presented in Table 3. The input layer of the model accepts raw time-series data consisting of 24 time steps and 8 feature dimensions (24×8) for target prediction. Graph convolutional networks (GCN) are utilized for spatial feature extraction, while bidirectional long short-term memory (BiLSTM) networks are employed for temporal feature extraction. During model training, a Dropout rate of 0.2 was applied to mitigate overfitting. Finally, the feature sequence is transformed into a 1×512 vector, and AdaBoost integration is used to dynamically adjust weights, ultimately outputting the power-load forecast for the subsequent hour. The hyperparameter design of the aforementioned model is configured to fit the provided dataset and ensure the model's prediction accuracy.

Table 3. Model structure parameters.

Types of Layer Structures	Input Dimension	Output Dimension
Input layer	24×8	—
GCN layer	24×8	24×128
Dropout	24×128	24×128
BiLSTM layer	24×128	1×512
Output layer	1×512	1×1

4.3. A Comparative Analysis with the Traditional Model

Figures 7 and 8 show the comparison of forecasting results between the proposed model and traditional power-load forecasting models, while Figures 9 and 10 present the comparison of forecasting errors. The forecasting results are presented with hourly intervals.

From Figures 7 and 8, it can be seen that the GCN-BiLSTM-AB model outperforms traditional power-load forecasting models in terms of prediction accuracy, whether forecasting daily load or the load for the upcoming week.

The results in Figures 9 and 10 allow for a quantitative description of the forecasting performance. Based on MAE measurements, compared with LSTM, GRU, CNN-LSTM, GCN-LSTM, and CNN-BiLSTM, the prediction accuracy improved by 2.41 and 1.27 for one day, 0.92 and 0.87 for one week, 1.39 and 0.44, and 1.40 and 1.65, respectively. Based on MAPE measurements, compared with LSTM, GRU, CNN-LSTM, GCN-LSTM, and CNN-BiLSTM, the prediction accuracy improved by 3.63% and 2.80%, 1.42% and 1.73%, 2.35% and 0.95%, and 2.35% and 3.55%, respectively. Based on RMSE measurements, compared

with LSTM, GRU, CNN-LSTM, GCN-LSTM, and CNN-BiLSTM, the prediction accuracy improved by 2.8370 and 1.73, 1.16 and 1.27, 1.62 and 0.59, and 1.83 and 2.18, respectively.

One day power load prediction results comparison

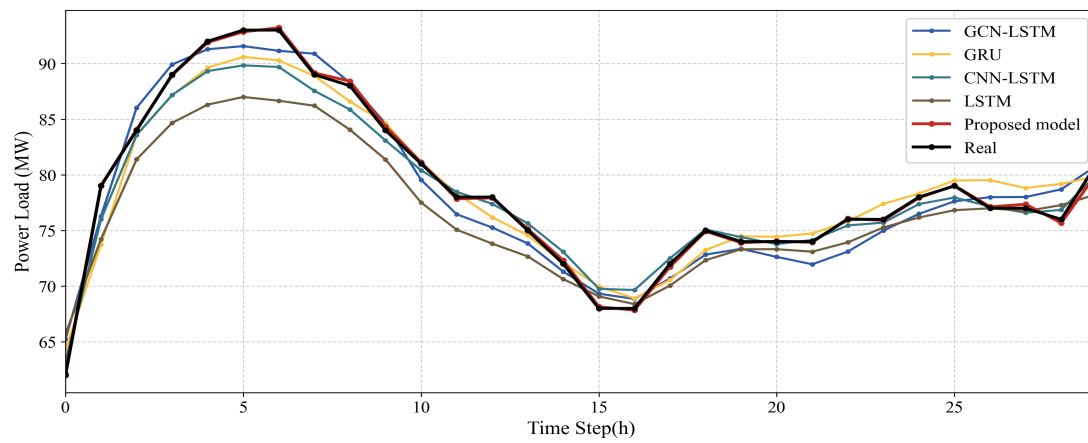


Figure 7. One day power-load prediction results comparison.

One week power load prediction results comparison

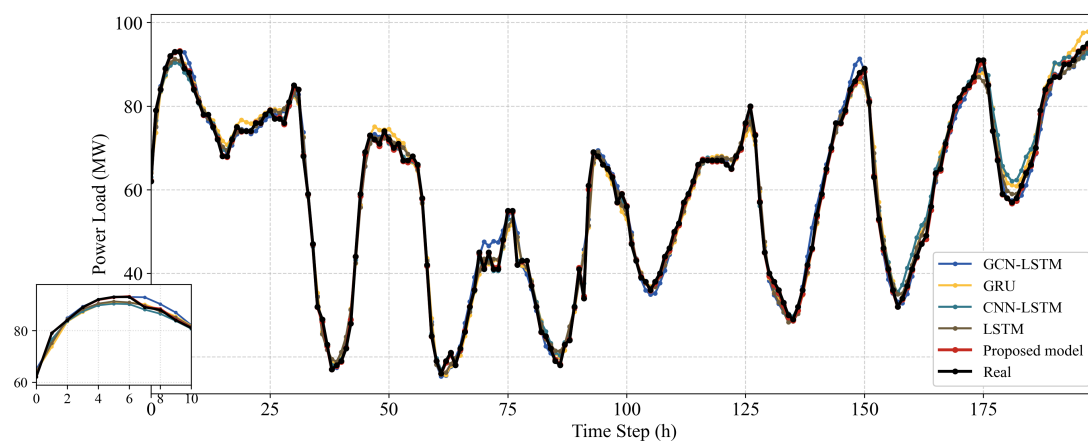


Figure 8. One week power-load prediction results comparison.

Prediction Error Comparison of one day

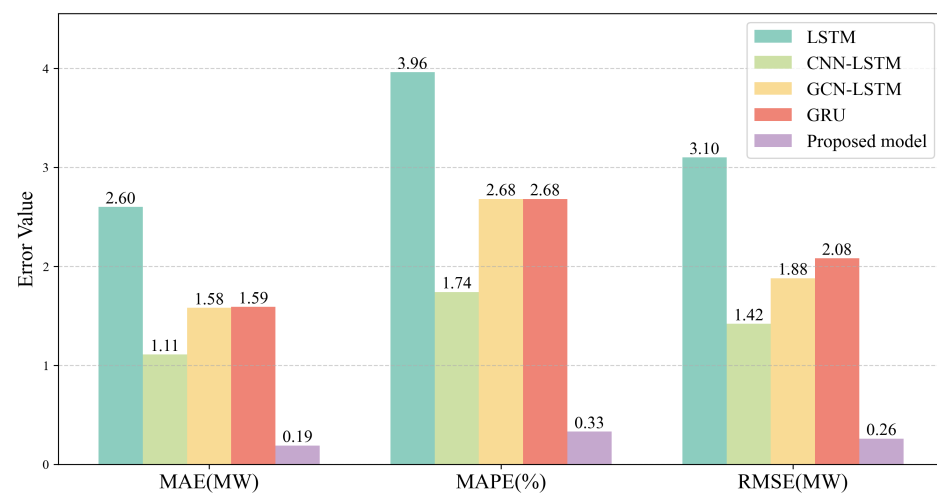


Figure 9. Prediction error comparison of one day.

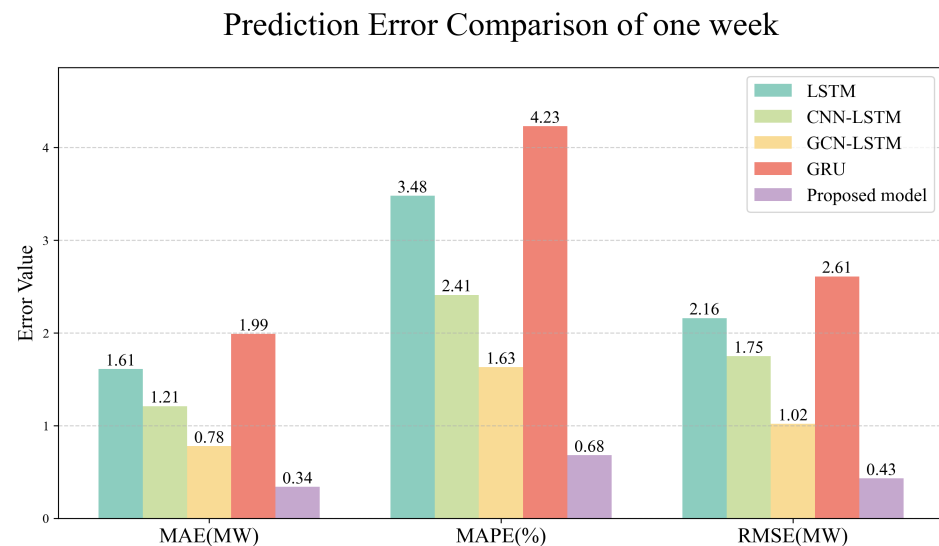


Figure 10. Prediction error comparison of one week.

Therefore, the comparison results show that GCN-BiLSTM-AB can better capture the spatiotemporal features of the input data, leading to more accurate predictions.

4.4. Model Prediction Robustness Analysis

To verify the effectiveness of using AdaBoost for model integration and incorporating Bayesian methods to enhance prediction robustness, the proposed model is compared with BiLSTM, GCN-BiLSTM, and GCN-BiLSTM-Adaboost models.

Figure 11 shows the one-day forecasting result comparison between the proposed model and the BiLSTM, GCN-BiLSTM, and GCN-BiLSTM-Adaboost models. Table 4 presents the forecasting errors, and Figure 12 illustrates the one-week forecasting results of the proposed model.

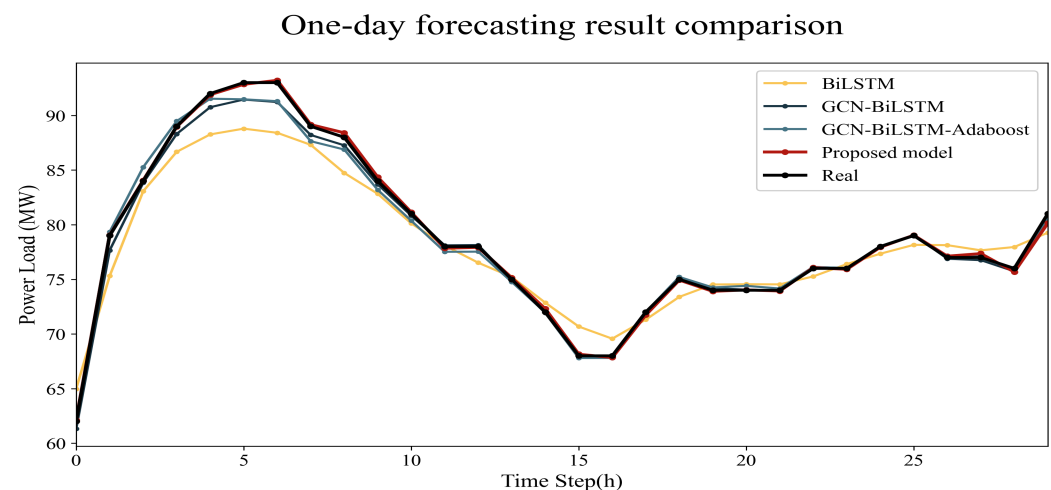


Figure 11. One-day forecasting result comparison.

As illustrated in Figure 11, during periods of daily load abrupt changes, other models exhibit degraded forecasting performance. In contrast, the GCN-BiLSTM-AB model retains relatively accurate predictions even at load turning points.

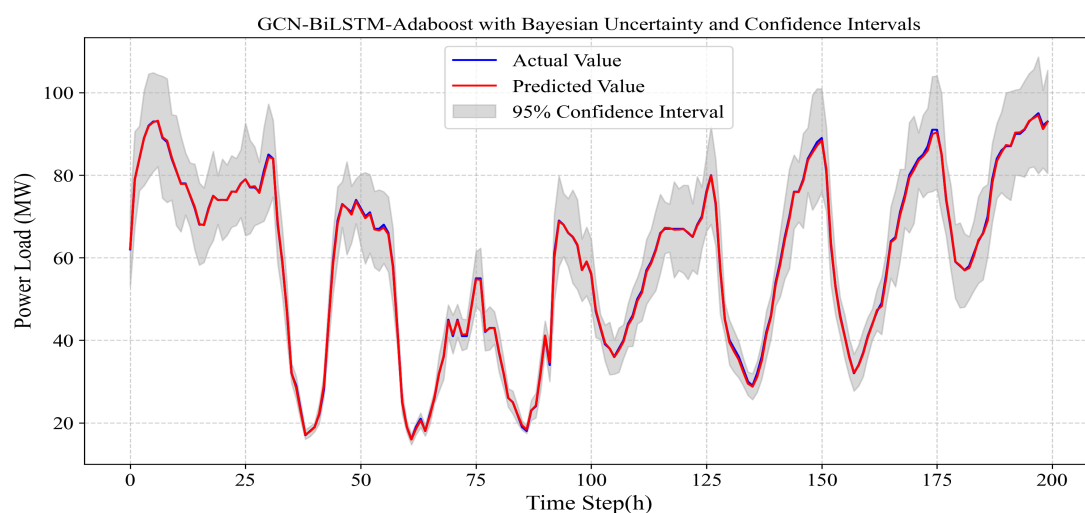
It can also be observed that as the model evolves from the standalone BiLSTM, to GCN-BiLSTM, then to GCN-BiLSTM-Adaboost, and finally to the integration of Bayesian methods into GCN-BiLSTM-Adaboost, the three types of prediction errors consistently decrease. This trend indicates a gradual improvement in the model's forecasting performance.

Table 4. Model performance comparison (A week/A day).

Model	A Week			A Day		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
BiLSTM	2.65	5.27%	3.41	1.86	3.12%	2.26
GCN-BiLSTM	1.99	4.23%	2.61	0.70	1.19%	1.02
GCN-BiLSTM-A	1.93	3.78%	2.57	0.26	0.41%	0.37
Proposed model	0.34	0.68%	0.43	0.19	0.33%	0.26

Figure 12 shows the weekly power-load forecasting results of the proposed model, as well as the prediction uncertainty within the 95% confidence interval. The model proposed in this study takes 5 min and 56 s to run. It can be seen that the weekly power-load forecasts align well with the actual values, enabling relatively accurate power-load prediction.

This confirms that the AdaBoost ensemble algorithm can dynamically adjust the weights of weak predictors, while the Bayesian method—implemented via Monte Carlo Dropout—effectively mitigates overfitting and quantifies prediction uncertainty. By incorporating uncertainty-based weighting into AdaBoost, the stability of final predictions is enhanced, enabling the model to retain accuracy even during abrupt power-load changes induced by events such as extreme weather.

**Figure 12.** One-week prediction result.

The 95% confidence interval in the Figure 12 is the prediction uncertainty range generated by Monte Carlo Dropout sampling. Its statistical significance is that under repeated sampling conditions, the true value has a 95% probability of falling within this interval range. By visualizing the confidence interval, the model not only provides point predictions but also presents risk quantification indicators.

5. Conclusions

This paper proposes a GCN-BiLSTM-based power-load forecasting model, integrating models through the AdaBoost algorithm and incorporating Bayesian methods to enhance the robustness of the model's predictions. The key findings of this study are:

(1) The BiLSTM network excels at extracting temporal dependency features, and the GCN network uniquely models topological relationships. The integrated GCN-BiLSTM network efficiently captures temporal and spatial features from input data, enhancing prediction accuracy.

(2) The Bayesian method is a probabilistic statistical inference approach capable of quantifying uncertainties in data and models. In this paper, the Bayesian method is implemented through Monte Carlo Dropout, which randomly drops neurons during the prediction process to prevent overfitting. Simultaneously, multiple sampling is used to calculate means and variances, yielding a more reasonable predictive distribution.

(3) The Adaboost algorithm uses GCN-BiLSTM as the weak learner, sets thresholds to dynamically adjust sample weights, incorporates “uncertainty” into the final weighted fusion of weak learners, and reduces the weights of predictions with higher uncertainty. These improvements not only retain the core idea of the Adaboost algorithm that gradually focuses on hard samples but also achieve more accurate weight allocation according to the characteristics of power-load forecasting, improving the prediction accuracy for key scenarios while enhancing the overall robustness of the model.

(4) The Spearman correlation coefficient balances efficiency (fast calculation), robustness (noise resistance), and interpretability (clear physical meaning) in power-load forecasting, and is particularly suitable for constructing graph structures based on statistical priors in multivariate time-series forecasting. However, KNN, Learned Graphs, and Mutual Information have limited performance in practical applications due to issues such as overly strong assumptions, complex calculations, and poor stability. Compared with the other three methods, the prediction results obtained by the graph construction method using the Spearman correlation coefficient show an average overall improvement in *MAE*, *MAPE*, and *RMSE* by 0.10, 0.19%, and 0.15, respectively.

The research results show that, compared to traditional power-load forecasting models, the proposed GCN-BiLSTM-AB hybrid network effectively captures spatiotemporal features of data, achieving more accurate predictions. Additionally, the network exhibits high robustness and generalization capabilities during sudden power-load fluctuations, ensuring stable predictions even in scenarios such as extreme weather events causing abrupt load changes. The predictive capabilities of GCN-BiLSTM-AB can assist power dispatch in making scientific and reasonable decisions, thereby reducing costs, improving efficiency, and enhancing the overall stability of the power system.

While the proposed GCN-BiLSTM-AB demonstrates superior performance, several limitations warrant further investigation: (1) The static graph construction may not capture dynamic feature interactions during extreme events; (2) Computational overhead increases linearly with the number of weak predictors ($K = 10$) and Monte Carlo samples (100).

Regarding the above limitations, the future work that can be done is as follows:

(1) Introduce dynamic graph neural networks (DGNN) or temporal attention mechanisms to enable the adjacency matrix to be adaptively adjusted over time.

(2) Replacing MC dropout with Bayesian neural networks (BNNs) via Laplace approximation, which provides uncertainty estimates without iterative sampling;

(3) Implementing surrogate models (e.g., Gaussian processes) to approximate the posterior distribution of weak predictors, reducing the need for direct sampling;

(4) Adopting quantization-aware training to compress the ensemble model for edge deployment.

Additionally, multi-modal data fusion (e.g., grid topology maps) will be explored to enhance robustness under complex weather conditions.

Author Contributions: G.Z.: Conceptualization, Methodology; J.L. (Jiarui Li): Methodology, Software, Validation, Writing—original draft; J.L. (Jian Li): Conceptualization, Methodology; J.L. (Jiatong Li): writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are not publicly available due to privacy restrictions. Data may be made available upon reasonable request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kwilinski, A.; Lyulyov, O.; Dzwigol, H.; Vakulenko, I.; Pimonenko, T. Integrative smart grids' assessment system. *Energies* **2022**, *15*, 545. [\[CrossRef\]](#)
2. Kumar, G.B.; Sarojini, R.K.; Palanisamy, K.; Padmanaban, S.; Holm-Nielsen, J.B. Large scale renewable energy integration: Issues and solutions. *Energies* **2019**, *12*, 1996. [\[CrossRef\]](#)
3. Shah, S.A.H.; Ahmed, U.; Bilal, M.; Khan, A.R.; Razzaq, S.; Aziz, I.; Mahmood, A. Improved electric load forecasting using quantile long short-term memory network with dual attention mechanism. *Energy Rep.* **2025**, *13*, 2343–2353. [\[CrossRef\]](#)
4. Rui, H.; Lingli, Z.; Feng, G.; Yuhong, W.; Yalan, Y.; Xiaofeng, X. Short-term power load forecasting method based on variational modal decomposition for convolutional long-short-term memory network. *Mod. Electr. Power* **2024**, *41*, 97–105.
5. Fahad, M.U.; Arbab, N. Factor affecting short term load forecasting. *J. Clean Energy Technol.* **2014**, *2*, 305–309. [\[CrossRef\]](#)
6. Peng, X.; Li, C.; Jia, S.; Zhou, L.; Wang, B.; Che, J. A short-term wind power prediction method based on deep learning and multistage ensemble algorithm. *Wind Energy* **2022**, *25*, 1610–1625. [\[CrossRef\]](#)
7. Almalaq, A.; Edwards, G. A review of deep learning methods applied on load forecasting. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 511–516.
8. Li, L.; Ota, K.; Dong, M. Everything is image: CNN-based short-term electrical load forecasting for smart grid. In Proceedings of the 2017 14th International Symposium on Pervasive Systems, Algorithms and Networks & 2017 11th International Conference on Frontier of Computer Science and Technology & 2017 Third International Symposium of Creative Computing (ISPAN-FCST-ISCC), Exeter, UK, 21–23 June 2017; pp. 344–351.
9. Muzaffar, S.; Afshari, A. Short-term load forecasts using LSTM networks. *Energy Procedia* **2019**, *158*, 2922–2927. [\[CrossRef\]](#)
10. L'Heureux, A.; Grolinger, K.; Capretz, M.A. Transformer-based model for electrical load forecasting. *Energies* **2022**, *15*, 4993. [\[CrossRef\]](#)
11. Han, J.; Zeng, P. Short-term power load forecasting based on hybrid feature extraction and parallel BiLSTM network. *Comput. Electr. Eng.* **2024**, *119*, 109631. [\[CrossRef\]](#)
12. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM networks. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; Volume 4, pp. 2047–2052.
13. Chen, H.; Zhu, M.; Hu, X.; Wang, J.; Sun, Y.; Yang, J.; Li, B.; Meng, X. Multifeature Short-Term Power Load Forecasting Based on GCN-LSTM. *Int. Trans. Electr. Energy Syst.* **2023**, *2023*, 8846554. [\[CrossRef\]](#)
14. Li, W.Q.; Chang, L. A combination model with variable weight optimization for short-term electrical load forecasting. *Energy* **2018**, *164*, 575–593. [\[CrossRef\]](#)
15. Bates, J.M.; Granger, C.W. The combination of forecasts. *J. Oper. Res. Soc.* **1969**, *20*, 451–468. [\[CrossRef\]](#)
16. Ma, Z.; Mei, G. A hybrid attention-based deep learning approach for wind power prediction. *Appl. Energy* **2022**, *323*, 119608. [\[CrossRef\]](#)
17. Bouktif, S.; Fiaz, A.; Ouni, A.; Serhani, M.A. Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting. *Energies* **2020**, *13*, 391. [\[CrossRef\]](#)
18. Wei, M.; Wen, M.; Zhang, Y. A novel spatial electric load forecasting method based on LDTW and GCN. *IET Gener. Transm. Distrib.* **2024**, *18*, 491–505. [\[CrossRef\]](#)
19. Song, L.; Jin, Y.; Lin, T.; Zhao, S.; Wei, Z.; Wang, H. Remaining useful life prediction method based on the spatiotemporal graph and GCN nested parallel route model. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 3511912. [\[CrossRef\]](#)
20. Chok, N.S. Pearson's Versus Spearman's and Kendall's Correlation Coefficients for Continuous Data. Ph.D. Thesis, University of Pittsburgh, Pittsburgh, PA, USA, 2010.
21. Yang, M.; Wang, J. Adaptability of financial time series prediction based on BiLSTM. *Procedia Comput. Sci.* **2022**, *199*, 18–25. [\[CrossRef\]](#)
22. Memarzadeh, G.; Keynia, F. Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm. *Electr. Power Syst. Res.* **2021**, *192*, 106995. [\[CrossRef\]](#)
23. Ying, C.; Qi-Guang, M.; Jia-Chen, L.; Lin, G. Advance and prospects of AdaBoost algorithm. *Acta Autom. Sin.* **2013**, *39*, 745–758.
24. Nirmal, S.; Patil, P.; Kumar, J.R.R. CNN-AdaBoost based hybrid model for electricity theft detection in smart grid. *e-Prime-Adv. Electr. Eng. Electron. Energy* **2024**, *7*, 100452. [\[CrossRef\]](#)
25. Jin, X.B.; Zheng, W.Z.; Kong, J.L.; Wang, X.Y.; Bai, Y.T.; Su, T.L.; Lin, S. Deep-learning forecasting method for electric power load via attention-based encoder-decoder with bayesian optimization. *Energies* **2021**, *14*, 1596. [\[CrossRef\]](#)

26. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 1050–1059.
27. Wang, Y.; Wu, M.; Li, X.; Xie, L.; Chen, Z. A survey on graph neural networks for remaining useful life prediction: Methodologies, evaluation and future trends. *Mech. Syst. Signal Process.* **2025**, *229*, 112449. [[CrossRef](#)]
28. Sieranoja, S.; Fränti, P. Constructing a high-dimensional k NN-graph using a z-order curve. *J. Exp. Algorithmics JEA* **2018**, *23*, 1–21. [[CrossRef](#)]
29. Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; Huang, J. Graph representation learning via graphical mutual information maximization. In *WWW '20: Proceedings of The Web Conference 2020, Taipei, Taiwan, 20–24 April 2020*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 259–270.
30. Wang, H.; Fu, Y.; Yu, T.; Hu, L.; Jiang, W.; Pu, S. Prose: Graph structure learning via progressive strategy. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 25–29 August 2023; pp. 2337–2348.

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.