

Research Article

Load Forecasting Method Based on Improved Deep Learning in Cloud Computing Environment

Kai Zhang ¹, Wei Guo ², Jian Feng ¹ and Mei Liu ³

¹State Grid Hebei Electric Power Co., Ltd., Shijiazhuang 050021, China

²Market Service Center, State Grid Hebei Electric Power Co., Ltd., Shijiazhuang 050021, China

³Beijing Tsingsoft Technology Co., Ltd., Beijing 100085, China

Correspondence should be addressed to Wei Guo; yxzx_guow@he.sgcc.com.cn

Received 4 August 2021; Revised 8 September 2021; Accepted 11 September 2021; Published 12 October 2021

Academic Editor: Yi-Zhang Jiang

Copyright © 2021 Kai Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For the problems of low accuracy and low efficiency of most load forecasting methods, a load forecasting method based on improved deep learning in cloud computing environment is proposed. Firstly, the preprocessed data set is divided into several data partitions with relatively balanced data volume through spatial grid, so as to better detect abnormal data. Then, the density peak clustering algorithm based on spark is used to detect abnormal data in each partition, and the local clusters and abnormal points are merged. The parallel processing of data is realized by using spark cluster computing platform. Finally, the deep belief network is used for load classification, and the classification results are input into the empirical mode decomposition-gating recurrent unit network model, and the load prediction results are obtained through learning. Based on the load data of a power grid, the experimental results demonstrate that the mean prediction error of the proposed method is basically controlled within 3% in the short term and 0.023 MW, 19.75%, and 2.76% in the long term, which are better than other comparison methods, and the parallel performance is good, which has a certain feasibility.

1. Introduction

With the vigorous development of social and economic level, the power consumption of industry, commerce, and residents in the power grid shows the characteristics of rapid growth. Due to the different production processes and peak and valley periods of power consumption among different industries, there are certain differences in load characteristics among different users [1, 2]. At the same time, the load characteristics of power users also change with seasonal changes, weather changes, characteristic days, power consumption areas, and other factors. Some loads fluctuate violently in a short time and there is a great peak valley difference [3]. Accurately analyze the load characteristics and make corresponding prediction, which can not only improve the power supply economic benefits of the power grid but also help the power grid provide strong decision support in optimizing energy structure and rational allocation of resources [4, 5]. Therefore, how to

comprehensively analyze the power load characteristics to achieve accurate load forecasting has become an important and difficult research.

On the other hand, due to the extensive use of sensors and smart meters in the power grid, it can quickly obtain massive power consumption data with high dimension and fine granularity, thus forming user power big data. The load big data contains rich user characteristic information, which can be deeply mined to give full play to its maximum value [6, 7]. At present, the commonly used load data analysis and application methods can include two types. The first type of method is time series analysis method, such as multiple linear regression, autoregressive moving average model, and other methods. This method requires that the time series used for calculation be relatively stable. It is generally suitable for long-term load forecasting with stable growth, and it is difficult to be applied to short-term load with frequent fluctuations [8, 9]. The second type is artificial intelligence technology represented by neural network and

its improved combination method. This method has good overall prediction performance but does not consider the characteristics of different time periods and different load types, and its efficiency needs to be improved when dealing with huge data volume and complex data structure [10].

In view of the unsatisfactory performance of existing methods in power grid load forecasting, a power grid load forecasting method based on improved deep learning in cloud computing environment is proposed. Its innovations are summarized as follows:

- (1) Due to the high time complexity of the fast density peak algorithm, the proposed method divides the data set to be clustered into multiple data partitions with relatively balanced data volume through spatial grid and designs a parallel algorithm by using spark parallel programming model. The power load is detected in parallel in the data partition corresponding to each computing node, and the detected abnormal point set is combined, so as to reduce the computational complexity and ensure the accuracy of abnormal data detection.
- (2) For full use of the massive data of power grid load and considering the periodicity and regularity of load itself, the proposed method finds the commonness between load data through fuzzy c-means clustering algorithm and inputs it into deep belief network (DBN) to classify the daily load to be measured.
- (3) For the high nonlinearity and instability of power load series caused by the superposition of various influencing factors, the proposed method combines empirical mode decomposition (EMD) and gated recurrent unit (GRU) to predict load. It avoids the random errors caused by modeling and forecasting the decomposed subsequences, respectively, so as to predict the load more accurately and quickly.

2. Related Research

Load forecasting is a process to predict the future load change and explore the dynamic and internal law of load data by analyzing the historical load series [11]. In recent years, academia has carried out a lot of research work on load forecasting methods. Most of the existing research results include three categories: traditional methods, artificial intelligence methods, and combined prediction methods [12].

Among them, traditional load forecasting method includes time series method and multiple linear regression method [13], which can show good performance in processing general load data, but when analyzing high-dimensional and complex power load data, the prediction accuracy is lacking, and the analysis efficiency is not high. The artificial intelligence and combination method play a great role in improving the performance of power grid load forecasting. Artificial intelligence methods include gray theory and support vector machine. For example, [14] constructs a gray correlation analysis model based on

interval gray effective information transformation, optimizes the resolution of traditional gray, and puts forward a multi-variable gray model to predict interval gray series, so as to effectively deal with the data prediction problem in big data. Reference [15] proposed a load forecasting method based on deep learning method for heat load demand, which has higher accuracy than the linear model of automatic feature selection, but the amount of calculation is relatively large. Reference [16] proposed a sequence to sequence recurrent neural network with attention to power load forecasting to capture the time dependence in load data. When dealing with huge and complex load data, a single artificial intelligence method can achieve good analysis performance in one aspect, but the overall load forecasting performance is not ideal.

The development of artificial intelligence technology promotes the continuous optimization of load forecasting. The combined forecasting method includes model combination in forecasting mechanism and weighted combination of forecasting results [17]. For example, [18] proposed two data preprocessing methods based on wavelet transform to extract data features, combined with least squares support vector machine learning engine and improved virus colony search algorithm to achieve accurate load prediction, but the prediction time range is limited. Reference [19] proposed a load forecasting method based on gated recurrent unit (GRU) combined with deep learning idea. Based on deep learning, different types of load influencing factors are processed, and the gated cyclic unit network is introduced to process the historical load series with time series characteristics, so as to finally complete the load forecasting. However, this method depends on data timing and needs more data preprocessing. Reference [20] proposed a short-term load forecasting combination method combining fuzzy time series and convolutional neural networks (CNN). By using the images created by the sequence values of multivariate time series and combined with CNN model, the relevant important parameters are automatically determined and extracted to accurately complete the load forecasting. Reference [21] proposed long-term and short-term memory, multilayer perceptron, and CNN to learn the relationship in the time series, but it is highly dependent on the load data in terms of capture time. Reference [22] realizes fast and accurate short-term load forecasting based on stacking factor condition limited Boltzmann machine and condition limited Boltzmann machine, but the model is complex and the processing efficiency is poor. The role of cloud computing environment on big data processing efficiency has not been deeply considered.

To sum up, the existing methods are difficult to give consideration to both prediction efficiency and accuracy in the process of load forecasting. Therefore, a power grid load forecasting method based on improved deep learning algorithm in cloud computing environment is proposed. In spark cluster computing platform, fuzzy C-means clustering is used to mine the relationship between data, DBN is used to realize load classification, and the classified load information is input into EMD-GRU combination model to complete load forecasting.

3. Proposed Method

3.1. Overall Framework. Spark is a fast and general cluster computing platform. Its main feature is that it can perform operations in memory and has high processing efficiency. Under the spark framework, combined with the cluster analysis method in data mining technology, a load forecasting method based on improved deep learning is proposed, and its process is shown in Figure 1.

Firstly, the preprocessed load data is partitioned through spark to better detect abnormal data. Then, in each partition, the density peak clustering algorithm is used to detect abnormal data, eliminate bad data, and reduce the impact of data on load forecasting. At the same time, combined with the main influencing factors and typical load types, DBN is used to classify different loads and determine the load category for forecast days. Finally, the load type and original load for forecast days are input into the EMD-GRU model for learning and training, so as to quickly match the corresponding model to realize load forecasting.

3.2. Load Data Preprocessing. In the power grid, the load data is measured by various sensors, and individual data is often lost or distorted due to collection, transmission, storage, and other factors. Before starting the prediction, the data needs to be preprocessed [23].

For the processing of missing data, the average method is generally used to fill in the missing value. The calculation is as follows:

$$L_t = \frac{L_{t-1} + L_{t+1}}{2}, \quad (1)$$

where L_t is the filling value at time t ; L_{t-1} and L_{t+1} represent the load values of the previous time and the next time, respectively. When the data at the beginning and end of the sequence is missing, the trend extrapolation method can be used for completion; and it is necessary to distinguish the authenticity of data values to improve the accuracy of prediction, that is, to identify and correct abnormal data. Abnormal data can be processed by rough Sugar Set Theory and wavelet theory.

The dimensions of the data are processed uniformly, and the values of load influencing factors are normalized to the $[0, 1]$ interval; that is,

$$\bar{x}_i = \frac{x_i - x_{\min}}{x_i - x_{\max}}, \quad (2)$$

where x_i and \bar{x}_i are normalized values before and after normalization; x_{\max} and x_{\min} are the maximum value and minimum value in the data sequence, respectively.

3.3. Load Data Partition. The parallel detection algorithm of abnormal data based on Spark density peak clustering can divide the data space into spatial grids. After division, when calculating the local density of data objects, only the sample data objects in the grid cells and the sample data objects in adjacent grid cells need to be considered, which greatly reduces the time complexity of the algorithm [24].

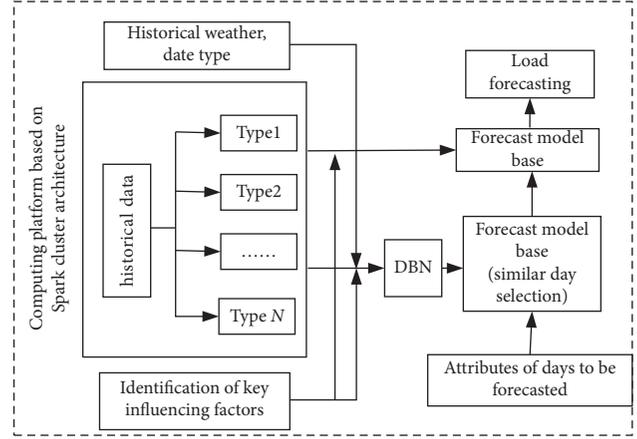


FIGURE 1: Load forecasting process based on improved deep learning in spark architecture method.

In order to divide the data evenly and with the load of each computing node being relatively balanced, to better detect abnormal data, the parallel abnormal data detection algorithm based on Spark density peak clustering uses K-dimensional tree (KD-tree) algorithm to divide the data space into multiple grid cells with roughly the same number of data objects.

When partitioning a data set, some data objects need to be assigned to multiple different partitions at the same time. This is because for the critical points in the grid cell after partitioning and because some adjacent data points within the neighborhood of their density intercept are not in the grid cell but in the adjacent grid cell, if their local density ρ_i is calculated directly. The local density error will be too large, resulting in too large error in anomaly detection. Therefore, in order to calculate the local density ρ_i of these critical points, some data objects need to be allocated to multiple different partitions at the same time [25].

Because the data partition and grid cell are one-to-one correspondence, each data partition corresponds to a grid cell, and each grid cell corresponds to a data partition. In a data partition, the local density and minimum distance of any data object in the data partition can be calculated. The pseudocode of the data partition algorithm is shown in Algorithm 1.

3.4. Abnormal Data Detection. Abnormal data detection is carried out in each partition to eliminate bad data and reduce the impact of data on load forecasting. In the abnormal data detection, firstly set the abnormal value judgment rules and then carry out local clustering anomaly detection in different partitions and merge the local clustering and abnormal points. Finally, the Spark parallel programming model is used to realize the parallel detection of abnormal data.

3.4.1. Abnormal Value Judgment Rules. Based on the density outlier detection method, it is considered that the cluster density of normal sample points is higher than that of outlier

Parameter implication:

Input: X is the data set; n_{\max} is the maximum number of sample data objects in the grid cell.

Output: Partitions is the data partition obtained after the data set is divided.

Begin

(1) Obtain multidimensional data space D_S through sample data set X .

(2) The KD-tree algorithm is used to divide the multidimensional data space D_S into multiple grid cells with relatively balanced size and no coincidence.

(3) The sample data objects are allocated to grid cells, and then the number of sample data objects contained in each grid cell is calculated.

(4) Initialize an empty Queue, add data space D_S to the Queue, and initialize an empty grid cell set D .

(5) Pop up the Queue header element S from the queue, and calculate the number n of sample data objects contained in S .

(6) **If** $n < n_{\max}$, **then**

Add S to D ;

(7) **If** $n \geq n_{\max}$, **then**

Calculate the variance of each dimension of the data object in the m -dimensional space in the spatial area S , select the dimension with the largest variance as the segmentation dimension, divide S into two subspaces $S1$ and $S2$ with an equal number of data objects, and then add $S1$ and $S2$ to the queue to wait for further division.

(8) **If** Queue is empty, **then**

The result of spatial meshing of data set X is set D ;

Otherwise Skip to step 5.

(9) According to the divided spatial grid set D , all data partitions of the data set can be obtained.

End

ALGORITHM 1: Pseudocode of data partition algorithm.

samples [26]. The proposed method combines the local outlier factor (LOF) algorithm with the density peak clustering algorithm for outlier detection. The specific formula is as follows:

$$\rho_a = \begin{cases} 0, & \text{LOF}_k(a) > 1, \\ \sum_{\text{dist}(x_a, x_j) \leq \sigma} \exp \left[- \left(\frac{\text{dist}(x_a, x_j)}{\text{dist}_{\text{cut off}}} \right)^2 \right], & \text{LOF}_k(a) \leq 1, \end{cases} \quad (3)$$

where σ is the density intercept, and the range less than σ from the data object becomes the density intercept neighborhood of the data object; $\text{LOF}_k(a)$ represents the mean value of the local reachable density ratio of the neighborhood point of point a to point a ; if this ratio is closer to 1, it indicates that the difference between the local reachable density of a and its neighborhood point density is small; $\text{dist}(x_a, x_j)$ is the reachable distance from x_a to x_j ; $\text{dist}_{\text{cut off}}$ is the intercept.

The mathematical description of determining that the sample point is an abnormal sample is as follows:

$$\begin{cases} \rho_a = 0, \\ \delta_a > \delta_{\Theta}, \end{cases} \quad \delta_a + \gamma_a, \quad (4)$$

$$\delta_{\Theta} = \frac{1}{N} \sum_{a=1}^N$$

where δ_a and δ_{Θ} are the relative distance and its threshold, respectively; γ_a is the empirical parameter; N is the total number of samples.

3.4.2. Local Clustering Anomaly Detection in Partition and Outlier Merging. In order to enable each computing node to perform local clustering anomaly detection on its corresponding data partition in parallel, it is necessary to optimize the original density peak clustering. In order to get rid of the intervention of subjective human factors, the original density peak clustering algorithm uses an auxiliary function to select the cluster center. The mathematical expression is as follows:

$$\gamma_i = \rho_i * \hat{\delta}_i, \quad (5)$$

where ρ_i is the local density of sample points and $\hat{\delta}_i$ is the minimum distance of samples.

For local clustering in the partition, a cluster center threshold needs to be given, and the γ value of each sample data object in the data partition is compared with the given cluster center threshold. If the γ value of the sample data object is greater than the set threshold, the data object is regarded as the candidate object of the cluster center.

In order to achieve the goal that each computing node can independently carry out clustering anomaly detection on the corresponding sample data partition, the data partition stage divides the sample data set into several overlapping data partitions, which also contain some common sample data objects. In the stage of local outlier merging and local cluster merging, the algorithm can find out all local clusters to be merged by evaluating the characteristics of these common data objects (i.e., critical points and expansion points). If the outlier sample points repeatedly appear in two or more data partitions, only one outlier sample point needs to be retained to eliminate the duplicate outlier sample points; form a set of global abnormal sample points.

3.4.3. Parallelization of Anomaly Data Detection Algorithm Based on Density Peak Clustering. In the case of massive power load data, the single-machine version of abnormal data detection algorithm is inefficient and cannot meet the requirements of abnormal data detection in power system. Therefore, the Spark parallel programming model is used to parallelize anomaly data detection algorithm to improve its efficiency.

The parallel detection algorithm of density peak clustering anomaly data based on spark mainly includes three important stages, namely, data partition and local clustering in the partition, anomaly detection and local clustering, and anomaly point merging. The algorithm has a lot of distance and density calculation, connection operation, and low efficiency. The execution sequence of the single-machine version of the algorithm can continue to the next stage only after the operation of the previous stage is completed. For this purpose, a clustering algorithm based on peak density is proposed, as shown in Figure 2.

In the Map stage, firstly, the power load data set is divided into multiple grid cells with approximately the same number of data objects by KD-tree algorithm, and then the data partition and grid cells are allocated one by one by data partition algorithm. In the Combine stage, the local clustering anomaly detection algorithm is executed in each partition to obtain the local clustering results of the data partition and the abnormal sample set in the data partition. In the Reduce phase, local cluster merging and outlier merging algorithms are implemented to connect local cluster markers to obtain the clustering results of global clusters and the global set of outlier sample points.

3.5. Load Forecasting Based on Improved Deep Learning in Spark Architecture. In load forecasting, the training data mainly includes load data and meteorological data [27, 28]. For traditional shallow learning methods, on the one hand, due to the simple structure model, it is difficult to learn the complex nonlinear mapping relationship in the training data. On the other hand, in order to avoid local minima, only a small number of features can be selected for training, which fundamentally limits its application scope and prediction accuracy.

The deep learning method based on Spark memory computing framework fully considers the selection of more dimensions and more extensive features, makes maximum use of massive data, and forecasts the load more accurately and quickly. At the same time, considering the characteristics of periodicity and regularity of power load itself, first find the commonness in massive load data by mining the typical load curve of historical load, then classify the load curve categories of forecast days through the load classifier based on DBN, and finally apply the corresponding typical load curve as a feature to the load predictor based on EMD-GRU. The flow chart of load forecasting based on improved deep learning under spark platform is shown in Figure 3.

3.5.1. Historical Load Clustering Based on Spark. After clustering the load curve by fuzzy C-means clustering (FCMC), the load characteristics of n distribution

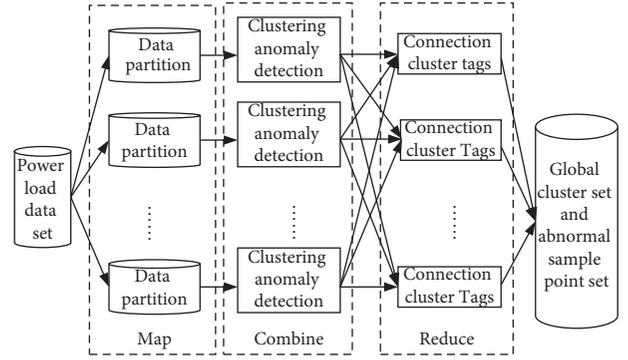


FIGURE 2: Algorithmic parallel framework.

transformer in the distribution network are classified into c homogeneous clusters. In order to solve the optimal membership matrix U and clustering center matrix Ψ , the following objective functions can be constructed according to the clustering criteria:

$$J(U, \Psi, \lambda) = \sum_{i=1}^n \sum_{k=1}^c (u_i^k)^\tau (\text{dist}_i^k)^2 + \sum_{i=1}^n \lambda_i \left(\sum_{k=1}^c u_i^k - 1 \right), \quad (6)$$

where $u_i^k \in [0, 1]$ represents the degree that the i -th distribution transformer belongs to the k -th cluster center, and the sum of membership degrees of one distribution transformer to all clusters is equal to 1; $(\text{dist}_i^k)^2$ is the Euclidean distance between the i -th distribution transformer and the k -th cluster center; $\tau \in [0, 2]$ is the weighted index; λ_i is the Lagrange multiplier of equality constraint. The iterative formula to minimize the objective function is calculated as follows:

$$u_i^k = \frac{1}{\sum_{q=1}^c (\text{dist}_i^k / d_i^q)^{2/(\tau-1)}}, \quad (7)$$

$$\varphi_k = \frac{\sum_{i=1}^n (u_i^k)^\tau x_i}{\sum_{i=1}^n (u_i^k)^\tau},$$

where x_i is the data volume of the i -th distribution transformer; φ_k is the cluster center quantity of fuzzy clustering.

In order to maintain the consistency and rapidity of the whole load forecasting process, the fuzzy C-means clustering algorithm based on spark memory computing environment is used. The specific steps include the following:

- (1) *Drive*. The main task is to initialize the basic functions of the program and drive each subtask through the function method in Spark. After the cluster is started, each node in the cluster will load each row of data in the data set file into Spark as a resilient distributed data set (RDD) and copy the shared data to each node in the cluster.
- (2) *Map Task*. The RDD of the load data set is calculated by line-by-line scanning, and the distance between the data object and each central point in the global variable cluster center is calculated. By comparison,

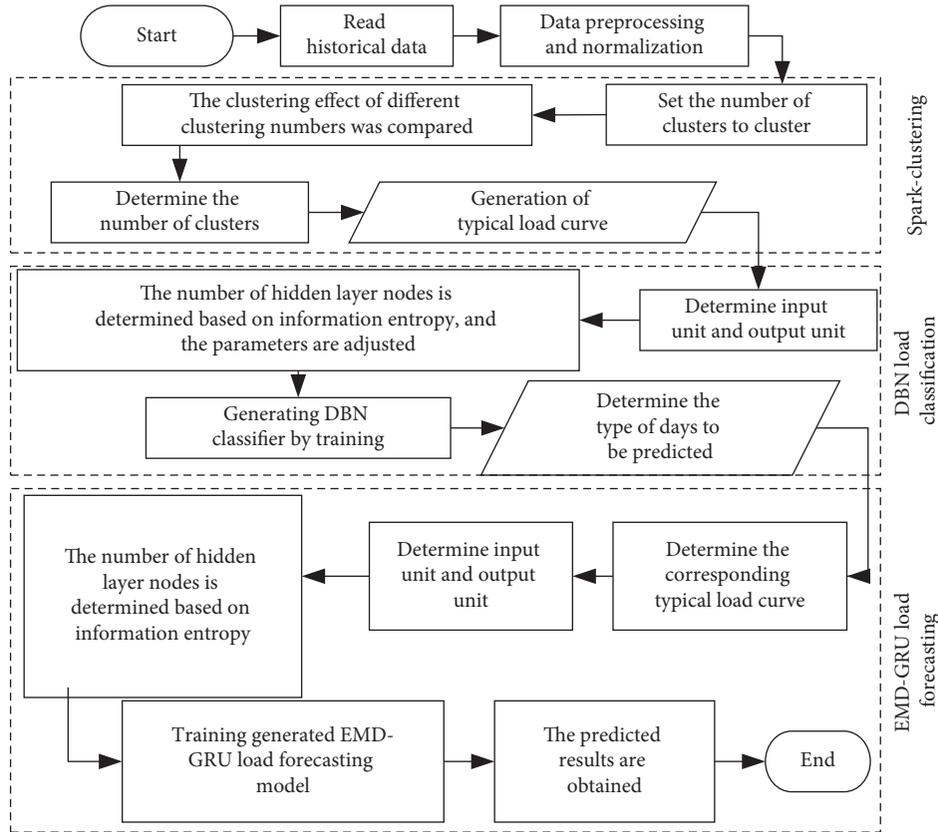


FIGURE 3: Overall framework.

the data object is assigned to the cluster with the smallest distance from the central point, and the final output is the key value pair <key, value >, where key is the cluster center and value is the data object belonging to the cluster center.

- (3) *Combine Task*. After the data set is mapped, a large number of RDD intermediate data sets will be generated. In order not to make the network communication a bottleneck, the values belonging to the same key are averaged locally, the local results <key, value> are obtained, and then the data are transmitted to the master node for processing to reduce the traffic.
- (4) *Reduce Task*. Summarize and merge the local results of the Combine process from the calculation node and return the result RDD in the form of an array to generate a global result. The number of Combine data points in each calculation node is different. The counter is used to count the data points to obtain a weight. During Reduce calculation, the weight and local results are used to calculate the global result.

3.5.2. Classification of Daily Load Types to be Forecasted Based on DBN. In order to obtain more accurate load forecasting results, the load categories on the forecast day shall be classified before the actual load forecasting. Then, the typical load curve of the corresponding category is

extracted, and the load in the curve is taken as a relevant input, so as to reflect the hidden commonness of massive load in the process of load forecasting. Because this process is essentially equivalent to extracting the characteristics of historical load and classifying the typical load curve and the probability generation model DBN formed by the stacking of restricted Boltzmann machine (RBM) has very strong learning ability in classification, the proposed method selects DBN to determine the load category of the day to be predicted.

The specific steps of classifying forecast days based on DBN are as follows:

- (1) Determine the input element and output element of DBN. Generally, the selected input elements include influencing factors related to the daily load to be predicted, such as load data on the same regular date as the day to be predicted, as well as its category, meteorological data, date attribute, and so forth. For the date attribute in the input element, convert it to 8 binary inputs. The setting of output element depends on the result of FCMC.
- (2) Based on information entropy theory, the number of neurons in the hidden layer of DBN is determined.
- (3) Stack RBM to form DBN and adjust the relevant parameters of the network. The parameter adjustment is mainly obtained through the continuous verification and analysis of the model effect.

- (4) Save the training data to HDFS and convert it to RDD format.
- (5) The DBN model is trained in parallel through data parallel. The specific method is to establish multiple data slices in spark cluster, create multiple copies of neural network model, train each slice at the same time, and cache the intermediate results and extract the training speed of the model from memory. After the training of each copy, the calculated parameter adjustment value is transmitted to the model parameter server, and a new parameter is applied to the parameter server for the next step of training.
- (6) The trained DBN model is used to determine the load curve category of the day to be predicted.

3.5.3. Load Forecasting Based on EMD-GRU. There will be some high-frequency noise components in the load subsequence based on EMD decomposition, which will affect the overall prediction accuracy. Therefore, an EMD-GRU load forecasting model based on feature selection is proposed. Feature selection of decomposed subsequences can not only avoid multiple prediction errors and improve prediction accuracy but also reduce prediction workload and model complexity.

The overall framework of load forecasting based on EMD-GRU is shown in Figure 4.

Firstly, the raw load sequence is decomposed by EMD into intrinsic mode functions (IMF) and residuals containing different characteristics of the original load series. Then, the original feature set is analyzed and screened by Pearson correlation coefficient method. Finally, combined with the selected time series characteristics and the raw load sequence, it is input into GRU model to realize power grid load forecasting. The Pearson correlation coefficient method is calculated as follows:

$$\eta = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (8)$$

where η is the correlation coefficient, x_i and y_i are the sample points, \bar{x} and \bar{y} are the sample means, and n is the number of samples.

4. Experiment and Analysis

4.1. Experimental Environment and Data Set. The Spark cluster built in the experiment is composed of 8 PCs with the same configuration. Each PC has 8G memory, 2T hard disk, and dual-core Intel i7 CPU, the main frequency is 4.7 GHz, and it runs Centos Linux operating system. One machine is the Master node, which is responsible for resource allocation and job scheduling of the whole cluster, and the other 7 machines are slave nodes, which are mainly used to store data and run tasks. The Spark cluster topology is shown in Figure 5.

The experimental data comes from the load data and influencing factor data collected by a regional power grid. The amount of data is TB and it has high dimensions. It is

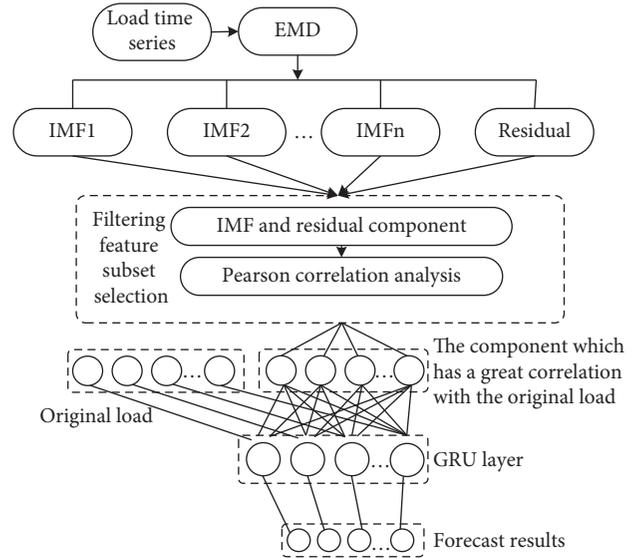


FIGURE 4: Overall block diagram of load forecasting based on EMD-GRU.

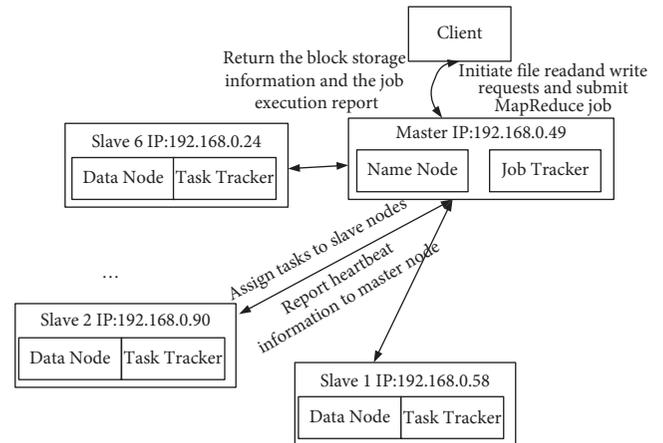


FIGURE 5: Topology of Spark cluster.

mainly structured and semistructured data, which is in line with the characteristics of power big data. The training sample is the power consumption data from May 1, 2020, to August 25, 2020, and the sampling interval is 1 h. Taking the 24-hour load data from May 1 to May 25, 2021, as the test sample, the load forecasting effect is evaluated by relative error, average error E_{ME} , root mean square error E_{RMSE} , and average absolute percentage error E_{MAPE} . The evaluation indexes are calculated as follows:

$$e_{ME} = \frac{1}{n+1} \sum_{i=0}^n (y(i) - \hat{y}(i)),$$

$$e_{RMSE} = \sqrt{\frac{\sum_{i=0}^n (y(i) - \hat{y}(i))^2}{n+1}}, \quad (9)$$

$$e_{MAPE} = \frac{1}{n+1} \sum_{i=0}^n \left| \frac{y(i) - \hat{y}(i)}{\hat{y}(i)} \right| \times 100\%,$$

where $y(i)$ and $\hat{y}(i)$ are the true load value and predicted value at time i .

4.2. Effectiveness Experiment. In order to demonstrate the effectiveness of the proposed method, the proposed method and the method in [22] are used to predict the power load, respectively. The load forecasting results in a short time are shown in Figure 6.

As can be seen from Figure 6 that the predicted value of the proposed method is closer to the real value, while the prediction deviation of the method in [22] is obvious, in order to quantitatively analyze the accuracy of load forecasting by the proposed method, six time points are selected for experimental comparison, and the results are shown in Table 1.

It can be seen from Table 1 that the accuracy of the proposed method has reached a high level, and the average prediction error is basically controlled within 3%, while the maximum relative error of the method in [22] is 6.35%.

4.3. Prediction Accuracy Analysis. In order to better demonstrate the load forecasting accuracy of the proposed method in a long time, the load within 10 days is predicted. The results of the proposed method and the load forecasting error evaluation index in [22] are shown in Table 2. Among them, January 1-3 is the new year's day holiday, January 4-8 are the working days, and January 9-10 are the weekend, which can better reflect the prediction effect under each load mode.

As can be seen from Table 2 that, in long-term load forecasting, the error evaluation index value of the proposed method is better than the comparison method. e_{ME} , e_{RMSE} , and e_{MAPE} are 0.023 MW, 19.75%, and 2.74% respectively. The mean value meets the assessment index of the State Grid, and the overall deviation control is better. Because the proposed method uses the density peak clustering algorithm to detect abnormal data and improve the quality of data set and uses the combination of DBN and EMD-GRU to realize load forecasting, it can ensure high forecasting accuracy. However, the model in [22] is complex and the control of data quality is insufficient, so the overall forecasting accuracy is lacking. In addition, the prediction accuracy of the proposed method in some time periods is not ideal, but the prediction errors are within the national grid standard.

In addition, the load forecasting error curve of the proposed method and the methods in [16], [18], and [22] within 48 h is shown in Figure 7.

As can be seen from Figure 7, the prediction error of the proposed method within 24 h is less than that of other comparison methods, with a minimum of about 1.31% and a maximum of about 3.38%. Except for individual points, the prediction error at most times is less than 3%. The proposed method ensures the accuracy of load data through data preprocessing, data partition detection, and other operations. On the basis of load classification by DBN, EMD-GRU model is used to reduce the load forecasting error to a great extent. Reference [22] was based on the method of using stacking condition limited Boltzmann machine to realize fast

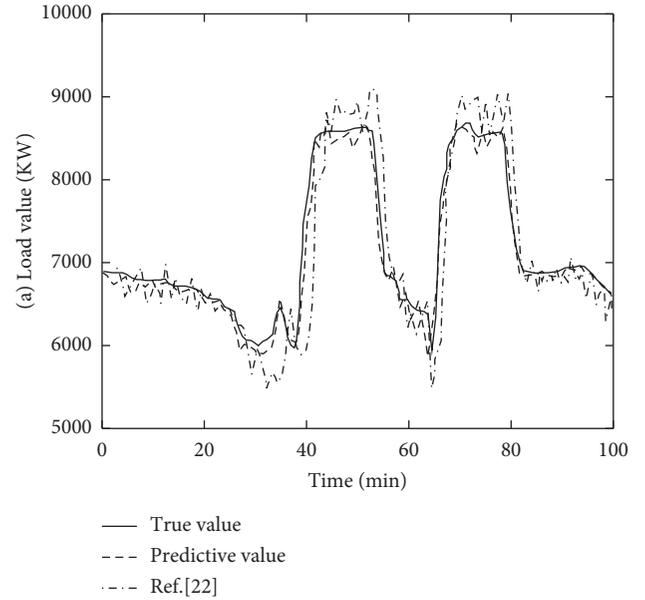


FIGURE 6: Load forecasting results of different methods in short time.

TABLE 1: Comparison results of different load forecasting methods.

No.	True value (MW)	Predictive value (MW)	
		Ref. [22]	The proposed method
1	8.254	9.106	7.922
2	8.739	9.054	8.623
3	7.825	8.234	7.711
4	7.241	8.338	7.489
5	8.632	9.104	8.532
6	8.963	9.272	8.645

and accurate short-term load forecasting, but it performed poorly in long-term load forecasting, and the growth rate of prediction error in the later stage was obvious. Reference [18] combined the least squares support vector machine learning engine and the improved virus colony search algorithm to realize load forecasting and used the wavelet transform method to extract data features. The prediction error is small in a short time, and the prediction error increases with the passage of time, due to its lack of full consideration of influencing factors. Reference [16] uses sequence to sequence cyclic neural network with attention to predicting power load. The model is single, the overall performance is poor, and it depends too much on time attributes, up to 4.68%.

4.4. Parallel Performance Analysis. With the increase of the amount of input data, the traditional serial processing method is difficult to meet the requirements of load forecasting. The parallel algorithm based on Spark can calculate the whole forecasting task in parallel to improve the computing efficiency. The speedup ratio is an important standard to measure the parallel efficiency of a parallelized system. Amdahl's law points out that the better the parallel

TABLE 2: Comparison results of different load forecasting methods.

Date	e_{ME} (MW)		e_{RMSE} (%)		e_{MAPE} (%)	
	The proposed method	Ref. [22]	The proposed method	Ref. [22]	The proposed method	Ref. [22]
1	-0.172	-0.279	19.73	22.57	3.41	4.25
2	0.357	0.750	21.14	20.62	3.25	3.97
3	0.136	0.621	16.75	16.30	2.59	2.45
4	0.173	0.237	23.57	25.79	2.77	3.72
5	-0.437	-0.324	21.49	25.41	2.09	3.34
6	0.574	-0.349	17.74	24.75	2.33	3.47
7	-0.275	0.427	19.27	19.54	2.74	2.95
8	-0.234	-0.732	17.54	22.73	3.12	3.57
9	0.314	0.449	22.73	21.74	2.47	2.75
10	-0.205	0.407	14.15	19.35	2.59	2.99
Average value	0.023	0.139	19.75	21.90	2.74	3.37

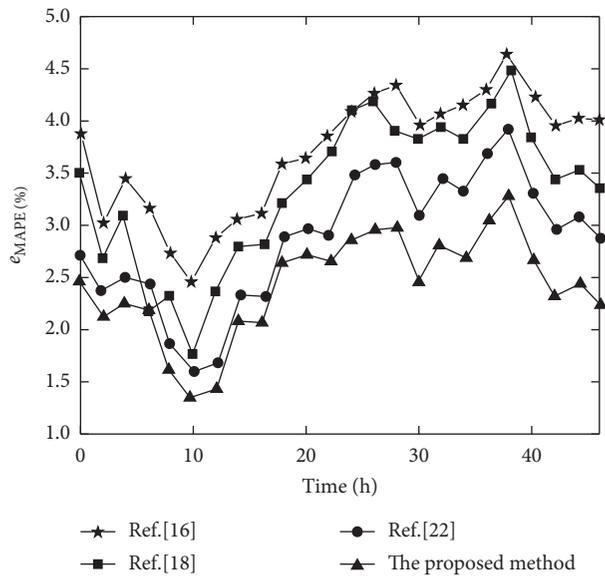


FIGURE 7: Comparison of load forecasting errors of different methods.

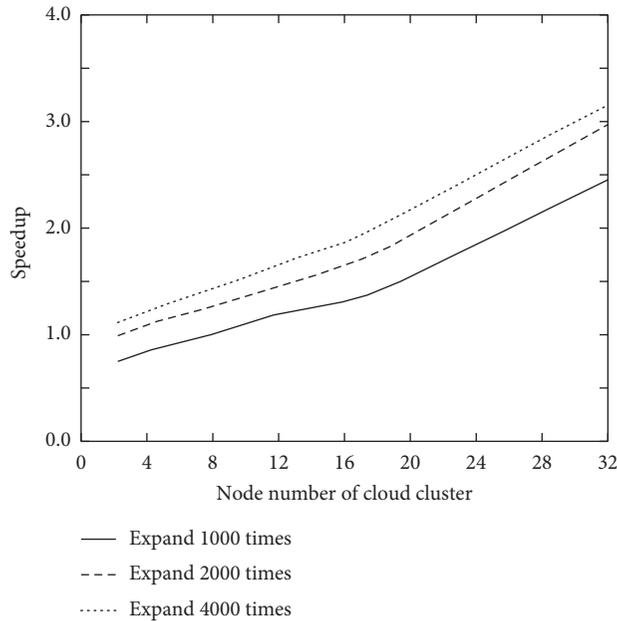


FIGURE 8: Parallelization performance results.

computing system, the speedup ratio is closer to the number of processors in the parallelized system. Therefore, in the parallel performance experiment, when the number of cluster nodes changes from 2 to 32, the acceleration ratio results of the cloud computing platform are shown in Figure 8.

As can be seen from Figure 8, when there are more than 16 cloud cluster nodes, additional consumption such as network transmission between nodes will increase, so the acceleration ratio will deteriorate with the increase of cloud cluster nodes. However, with the increase of the amount of data, the speedup ratio of this method still increases almost linearly, indicating good parallel performance.

5. Conclusions

The load forecasting method based on improved deep learning in cloud computing environment is proposed in this paper. The experimental results show that this method has high prediction accuracy and good parallel performance. However, in load forecasting based on deep learning algorithm, the proposed method still relies on manual selection when selecting the relevant features of load forecasting and does not make full use of the feature extraction ability of unsupervised learning in deep learning. In the next research, we can study how to automatically extract the relevant features of load forecasting.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] Q. Wang and F. Jiang, "Integrating linear and nonlinear forecasting techniques based on grey theory and artificial intelligence to forecast shale gas monthly production in Pennsylvania and Texas of the United States," *Energy*, vol. 178, no. 6, pp. 781–803, 2019.
- [2] S. Sengar and X. Liu, "An efficient load forecasting in predictive control strategy using hybrid neural network," *Journal of Circuits, Systems, and Computers*, vol. 29, no. 1, pp. 567–576, 2020.
- [3] D.-J. Bae, B.-S. Kwon, C.-H. Moon, S.-H. Woo, and K.-B. Song, "Short-term load forecasting algorithm on weekdays considering the amount of behind-the-meter generation," *Journal of the Korean Institute of Illuminating and Electrical Installation Engineers*, vol. 34, no. 11, pp. 37–43, 2020.
- [4] S. O. Khomutov and N. A. Serebryakov, "Creation of a short-term load forecasting neural network model of electrical engineering complex of section regional electric grid 6-35 kV," *Transportation Systems and Technology*, vol. 6, no. 1, pp. 80–91, 2020.
- [5] G. Fernandez, V. Krishnasamy, J. Ali, J. Sathik, and S. Aleem, "Internet of things based real-time electric vehicle load forecasting and charging station recommendation," *ISA Transactions*, vol. 97, no. 4, pp. 431–447, 2020.
- [6] D. Ma, "Short-term load forecasting based on variable weighted synthesis of different kernel SVM," *Statistics and Applications*, vol. 9, no. 1, pp. 73–80, 2020.
- [7] P. D. Lê, D. M. Bui, D. A. Phm et al., "Applying statistical analysis for assessing the reliability of input data to improve the quality of short-term load forecasting for a Ho Chi Minh City distribution network," *Science & Technology Development Journal - Engineering and Technology*, vol. 2, no. 4, pp. 223–239, 2020.
- [8] A. Io, B. Sbe, and B. Ho, "A combined deep learning application for short term load forecasting," *Alexandria Engineering Journal*, vol. 60, no. 4, pp. 3807–3818, 2021.
- [9] M. Sun, T. Zhang, Y. Wang, G. Strbac, and C. Kang, "Using bayesian deep learning to capture uncertainty for residential net load forecasting," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 188–201, 2020.
- [10] F. Z. Abera and V. Khedkar, "Machine learning approach electric appliance consumption and peak demand forecasting of residential customers using smart meter data," *Wireless Personal Communications*, vol. 111, no. 1, pp. 65–82, 2020.
- [11] Q. Chen, M. Xia, T. Lu, X. Jiang, W. Liu, and Q. Sun, "Short-term load forecasting based on deep learning for end-user transformer subject to volatile electric heating loads," *IEEE Access*, vol. 7, pp. 162697–162707, 2019.
- [12] L. Wen, K. Zhou, S. Yang, and X. Lu, "Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting," *Energy*, vol. 171, pp. 1053–1065, 2019.
- [13] D. Kim, H. J. Jo, M. S. Kim, J. H. Roh, and J.-B. Park, "Short-term load forecasting based on deep learning model," *The Transactions of the Korean Institute of Electrical Engineers*, vol. 68, no. 9, pp. 1094–1099, 2019.
- [14] J. Ye, Y. Dang, and Y. Yang, "Forecasting the multifactorial interval grey number sequences using grey relational model and GM (1, N) model based on effective information transformation," *Soft Computing*, vol. 24, no. 7, pp. 5255–5269, 2020.
- [15] G. Suryanarayana, J. Lago, D. Geysen, P. Aleksiejuk, and C. Johansson, "Thermal load forecasting in district heating networks using deep learning and advanced feature selection methods," *Energy*, vol. 15, no. 3, pp. 141–149, 2018.
- [16] L. Sehovac and K. Grolinger, "Deep learning for load forecasting: sequence to sequence recurrent neural networks with attention," *IEEE Access*, vol. 8, pp. 36411–36426, 2020.
- [17] A. Ghasemi-Marzbali, "A developed short-term electricity price and load forecasting method based on data processing, support vector machine, and virus colony search," *Energy Efficiency*, vol. 13, no. 7, pp. 1–18, 2020.
- [18] H. J. Sadaei, P. C. de Lima e Silva, F. G. Guimarães, and M. H. Lee, "Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series," *Energy*, vol. 175, no. 5, pp. 365–377, 2019.
- [19] Z. Wang, B. Zhao, and W. Ji, "Short-term load forecasting method based on GRU-NN model," *Dianli Xitong Zidonghua/Automation of Electric Power Systems*, vol. 43, no. 5, pp. 53–58, 2019.
- [20] M. Cai, M. Pipattanasomporn, and S. Rahman, "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," *Applied Energy*, vol. 236, pp. 1078–1088, 2019.
- [21] F. Mehmood Butt, L. Hussain, L. Hussain, A. Mahmood, and K. Javed Lone, "Artificial Intelligence based accurately load

- forecasting system to forecast short and medium-term load demands,” *Mathematical Biosciences and Engineering*, vol. 18, no. 1, pp. 400–425, 2021.
- [22] L. Barolli, F. Xh Afa, and N. Javaid, “Short term load forecasting based on deep learning for smart grid applications,” *Advances in Intelligent Systems and Computing*, vol. 10, no. 25, pp. 276–288, 2019.
- [23] S. Smyl, N. G. Hua, and R. J. Hyndman, “Machine learning methods for GEFCom2017 probabilistic load forecasting,” *International Journal of Forecasting*, vol. 35, no. 4, pp. 1424–1431, 2019.
- [24] G. Fu, “Deep belief network based ensemble approach for cooling load forecasting of air-conditioning system,” *Energy*, vol. 148, pp. 269–282, 2018.
- [25] J. Wang, X. Chen, F. Zhang, F. Chen, and Y. Xin, “Building load forecasting using deep neural network with efficient feature fusion,” *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 1, pp. 160–169, 2021.
- [26] S. Bouktif, A. Fiaz, A. Ouni, and M. Serhani, “Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: comparison with machine learning approaches †,” *Energies*, vol. 11, no. 7, pp. 1636–1645, 2018.
- [27] Z. Kong, C. Zhang, H. Lv, F. Xiong, and Z. Fu, “Multimodal feature extraction and fusion deep neural networks for short-term load forecasting,” *IEEE Access*, vol. 8, no. 3, pp. 185373–185383, 2020.
- [28] W. Kim, Y. Han, K. J. Kim, and K.-W. Song, “Electricity load forecasting using advanced feature selection and optimal deep learning model for the variable refrigerant flow systems,” *Energy Report*, vol. 6, no. 2, pp. 2604–2618, 2020.