

## Research Article

# A Short-Term Load Forecasting Method Based on GRU-CNN Hybrid Neural Network Model

Lizhen Wu , Chun Kong, Xiaohong Hao, and Wei Chen

*College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, 730050, Gansu, China*

Correspondence should be addressed to Lizhen Wu; wulzlut@163.com

Received 19 November 2019; Revised 19 January 2020; Accepted 15 February 2020; Published 21 March 2020

Guest Editor: Chi-Hua Chen

Copyright © 2020 Lizhen Wu 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.

Short-term load forecasting (STLF) plays a very important role in improving the economy and stability of the power system operation. With the smart meters and smart sensors widely deployed in the power system, a large amount of data was generated but not fully utilized, these data are complex and diverse, and most of the STLF methods cannot well handle such a huge, complex, and diverse data. For better accuracy of STLF, a GRU-CNN hybrid neural network model which combines the gated recurrent unit (GRU) and convolutional neural networks (CNN) was proposed; the feature vector of time sequence data is extracted by the GRU module, and the feature vector of other high-dimensional data is extracted by the CNN module. The proposed model was tested in a real-world experiment, and the mean absolute percentage error (MAPE) and the root mean square error (RMSE) of the GRU-CNN model are the lowest among BPNN, GRU, and CNN forecasting methods; the proposed GRU-CNN model can more fully use data and achieve more accurate short-term load forecasting.

## 1. Introduction

Due to the difficulty of large-scale storage of electrical energy and electrical energy changes in power demand, it is required that the system power generation should be dynamically balanced with changes in load [1, 2]. Load forecasting plays an important role in power construction planning and power grid operation, accurate load forecasting can minimize the gap between electricity supply and demand, improving the stability of power systems [3]. A tiny little error increased in the load forecasting may cost millions of dollars lost every year [4], and thus it is essential to build an accurate load forecasting model. According to the predicted time range, the power load forecasts can be divided into long-term, medium-term, short-term, and ultra-short-term forecast [5]. This paper mainly focuses on short-term load forecasting, which predicts future loads from minutes to weeks; accurate STLF can help power system staff to develop reasonable production plans, maintain supply and demand balance, and ensure grid safety while reducing resource waste and electricity costs [6, 7].

With the large-scale application of smart meters and smart sensors in the power system, the degree of informatization continues to increase, a large amount of data are generated, and it provides a reliable source of data for accurate load forecasting. Meanwhile, with the continuous improvement of computer computing performance and the application of distributed parallel computing, powerful computing performance for massive data is provided. Under this background, many load forecasting methods based on massive data were emerged, and these methods are mainly divided into two categories, one is traditional statistical methods [8]: they are most frequently used in the early literature, including linear regression (LR) analysis approach and autoregressive moving average (ARMA) approach [9]. Lee C. K. proposed a lifting scheme and autoregressive integrated moving average (ARIMA) models to achieve STLF in [10]. These methods can achieve short-term load forecasting to a certain extent; however, there are many inherent nonlinear features in the massive data, and traditional statistical methods cannot well learn these nonlinear data [11]; so, it is very challenging for these traditional statistical methods to predict accurately in STLF, and these

traditional statistical methods cannot meet the requirements of load forecasting accuracy in the STLF. The other is machine learning methods; they have been widely and successfully used in prediction and classification problems, including artificial neural network (ANN) [12], support vector machine (SVM) [13], and fuzzy inference system (FIS) [14]. For better abstraction of nonlinear features, machine learning methods are good approaches to solve nonlinear problems, and in [15], Niu DX creates a system for power load forecasting using support vector machine and ant colony optimization. In [16], the authors present short-term load forecasting models developed by using the fuzzy logic and adaptive neuro-fuzzy inference system (ANFIS). In recent years, with the rise of artificial intelligence, ANN methods have been widely used in load forecasting. Back-propagation neural network (BPNN) is the first widely used ANN method for STLF [17]. A combined model, which used the back-propagation neural network (BPNN) with the multilabel algorithm based on K-nearest neighbor (K-NN) and K-means, was proposed for STLF in [18]; however, BPNN is a feedforward neural network, and it cannot well learn time sequence data in the power system [19]. In order to efficiently process the time sequence data, such as holiday, weather, and temperature information in the power system. Recurrent neural network (RNN) [20], a kind of neural network which is specific for processing sequence data, is widely used for STLF [21]. The authors in [22] use local RNN models to deal with the problem of long-term wind speed and power forecasting based on meteorological information. However, due to the excessive depth of time and the simple hidden layer in the traditional RNN structure, when the error back propagation is performed, there will be problems with gradient vanishing. It is impossible for RNN to learn long historical data. In response to the shortcomings of RNN, Hochreiter and Schmidhuber proposed a long short-term memory (LSTM) recurrent neural network in 1997 [23], which overcame the disadvantages of traditional RNNs by combining short-term memory with long-term memory through the gate control. A novel method which integrates LSTM and genetic algorithm (GA) was proposed for STLF [24], and it yielded a small mean absolute percentage error. Gated recurrent unit (GRU) [25] is a special type of recurrent neural network based on optimized LSTM, and the GRU internal unit is similar to the internal unit of the LSTM [26], except that the GRU combines the input gate and the forgetting gate in the LSTM into a single update gate. In [27], a novel system called multi-GRU (gated recurrent unit) prediction system was developed based on GRU models for electricity generation's planning and operation. And Wang proposed a novel approach to forecast short-term photovoltaic power based on GRU networks [28]. However, there is not only sequence data in the power system, but also other kinds of high-dimensional data, such as spatiotemporal matrix and image information in the power system. The GRU model cannot well handle all these kinds of high-dimensional data, the convolution neural network (CNN) [29] is ideal for processing high-dimensional data, which has been widely used in image recognition and the fields of prediction [30]. When there is a strong relationship between

the nearby data point, CNN can capture local trend features and scale-invariant features [31, 32]. In [33], the author proposed an end-to-end automatic image annotation method based on a deep CNN and multilabel data augmentation, and the model performs well in automatic image annotation.

In order to make full use of the various data in the power system and achieve accurate STLF, the GRU-CNN hybrid neural network model was proposed, which combines the GRU model with the CNN model. In the proposed model, the GRU module is used to model dynamic changes in historical load sequence data for better learning potential features in time sequence data. The CNN module is utilized to process spatiotemporal matrixes and map spatiotemporal matrixes into the feature vector. The GRU-CNN model combines the output of GRU and CNN to derive the load prediction result through the activation function. To verify the superiority of the GRU-CNN model in short-term load forecasting, the proposed method was compared with BPNN, GRU, and CNN models in a real-world experiment. The four models were trained and tested, and mean absolute percentage error (MAPE) and root mean square error (RMSE) were used as the evaluation indexes. The results of the experiments demonstrate that the GRU-CNN model achieves the best predicting performance in STLF among the four models.

This paper is organized as follows: In Section 2, the proposed GRU-CNN hybrid neural networks and its modules are introduced. The GRU-CNN model was utilized to forecast the electrical load, and also it was compared with BPNN, GRU, and CNN in a real-world case in Section 3. Finally, the conclusion is drawn in Section 4.

## 2. The GRU-CNN Hybrid Neural Network Model

**2.1. The Establishment of GRU Module.** RNN is a kind of artificial neural network which is suitable for analyzing and processing time sequence data, unlike traditional neural networks, which are based on the weight connection between the layers. RNN applies the hidden layers to preserve information from the previous moment, and the output is influenced by the current states and previous memories. For better understanding of RNN, the unrolled structure of RNN is shown in Figure 1

where  $x^{(t)}$  and  $\hat{y}^{(t)}$  represent the input and output at time  $t$ ,  $a^{(t)}$  represents the output of one single hidden layer at time  $t$ , and  $\omega_{aa}^{(t)}$ ,  $\omega_{ax}^{(t)}$ , and  $\omega_{ay}^{(t)}$  represent the hidden layers weight matrixes, the input weight matrixes, and the output weight matrixes, respectively. Figure 1 can be represented as following formulas:

$$\begin{aligned} a^{(t)} &= g_1(\omega_{aa}^{(t)} a^{(t-1)} + \omega_{ax}^{(t)} x^{(t-1)} + b_a), \\ \hat{y}^{(t)} &= g_2(\omega_{ay}^{(t)} a^{(t)} + b_y), \end{aligned} \quad (1)$$

where  $b_a$  and  $b_y$  represent the bias vectors of one single hidden layer and the output, respectively. and  $g_1$  and  $g_2$  are the nonlinear activation function. RNN performs well when the output is close to its associated inputs; however, when the

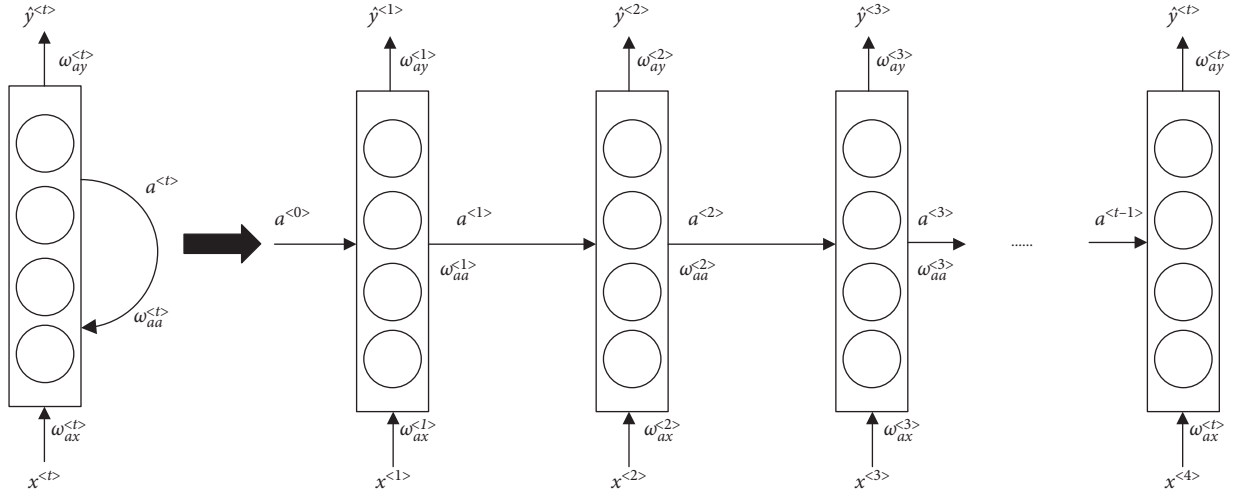


FIGURE 1: Recurrent neural network structure.

time interval is long and the number of weights becomes large, the input will have little effect on the output due to the gradient vanishing problem. In order to solve the gradient vanishing and simple hidden layer structure problems of RNN, a special type of RNN called GRU was proposed.

GRU is a variant of LSTM with a gated recurrent neural network structure, and comparing with LSTM, there are two gates (update gate and reset gate) in GRU and three gates (forgetting gate, input gate, and output gate) in LSTM; meanwhile, GRU has fewer training parameters than LSTM, so GRU converges quicker than LSTM during training [34]. The GRU structure is shown in Figure 2, where  $\sigma$  and  $\tanh$  are the activation functions,  $c^{<t-1>}$  is the input of the current unit, which is also the output of the previous unit,  $c^{<t>}$  is the output of the current unit, which links to the input of the next unit.  $x^{<t>}$  are the inputs of training data,  $\hat{y}^{<t>}$  is the outcome of this unit, generated by the activation function,  $\Gamma_r$  and  $\Gamma_u$  represent the reset gate and the update gate, respectively, and the candidate activation  $\tilde{c}^{<t>}$  is computed similarly to that of the traditional recurrent unit. There are two gates in GRU, one is the update gate, which preserve previous information to the current state; The value of  $\Gamma_u$  ranges from 0 to 1, the closer  $\Gamma_u$  is to zero, the more previous information it retains; the other is the reset gate, which is used to determine whether the current status and previous information are to be combined. The value of  $\Gamma_r$  ranges from -1 to 1, the smaller the value of  $\Gamma_r$ , the more previous information it ignores. According to Figure 2, the formulas of GRU can be shown as

$$\begin{aligned}\Gamma_u &= \sigma(\omega_u [c^{<t-1>}, x^{<t>}] + b_u), \\ \Gamma_r &= \sigma(\omega_r [c^{<t-1>}, x^{<t>}] + b_r), \\ \tilde{c}^{<t>} &= \tanh(\omega_c [\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c), \\ c^{<t>} &= (1 - \Gamma_u) * c^{<t-1>} + \Gamma_u * \tilde{c}^{<t>},\end{aligned}\quad (2)$$

where  $\omega_u$ ,  $\omega_r$ , and  $\omega_c$  represent the training weight matrix of the update gate, the reset gate, and the candidate activation  $\tilde{c}^{<t>}$ , respectively and  $b_u$ ,  $b_r$ , and  $b_c$  are the bias vectors.

**2.2. The Establishment of CNN Module.** CNN is a kind of artificial neural network which can well process high-dimensional data. It is commonly applied in visual image, video recognition, and text categorization. There are many smart sensors and devices in the power system. In order to preserve the spatial information of data recorded by smart sensors and devices in the power system, the spatiotemporal matrix was proposed and the spatiotemporal matrixes data are based on the location of the sensors and time sequence. The spatiotemporal matrix is shown as

$$X = \begin{bmatrix} X_1(1) & X_1(2) & \cdots & X_1(n) \\ X_2(1) & X_2(2) & \cdots & X_2(n) \\ \vdots & \vdots & \ddots & \vdots \\ X_k(1) & X_k(2) & \cdots & X_k(n) \end{bmatrix}, \quad (3)$$

where  $k$  represents the  $k^{th}$  smart sensor,  $n$  represents the  $n^{th}$  time sequence, and  $X_k(n)$  represents the data recorded by the  $k^{th}$  smart sensor at  $n$  time. In order to extract the load feature from the spatiotemporal matrix, CNN was used to process the spatiotemporal matrix. The structure of CNN is shown in Figure 3.

As shown in Figure 3, firstly, many two-dimensional spatiotemporal matrixes are stacked into three-dimensional matrix blocks, and then these blocks were applied with a convolution operation. The purpose of the convolution operation is to get a highly abstract feature, and after the convolution operation, the outputs of convolution operation are applied with pooling operation. Pooling operation does not change the depth of the input matrix, but it can reduce the size of matrixes and the number of nodes, so as to reduce the parameters in the entire neural networks. After repeated convolution and pooling operations, the highly abstract feature was obtained and flattened to an one-dimensional vector, so it can be connected with the fully connected layer. Then, the weights and bias parameters of the fully connected layer can be calculated iteratively. Finally, prediction results are obtained through the output of activation function.

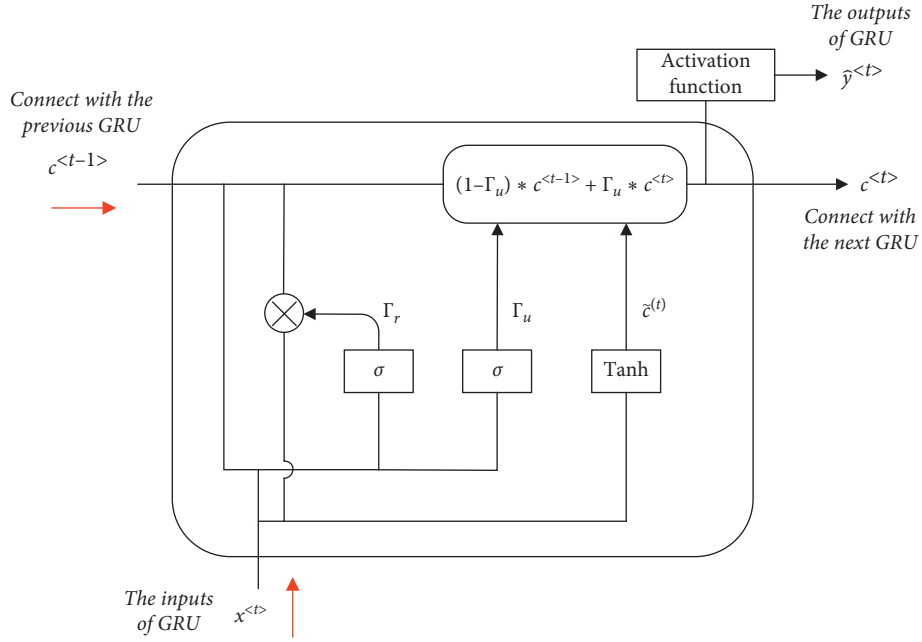


FIGURE 2: Gated recurrent unit structure.

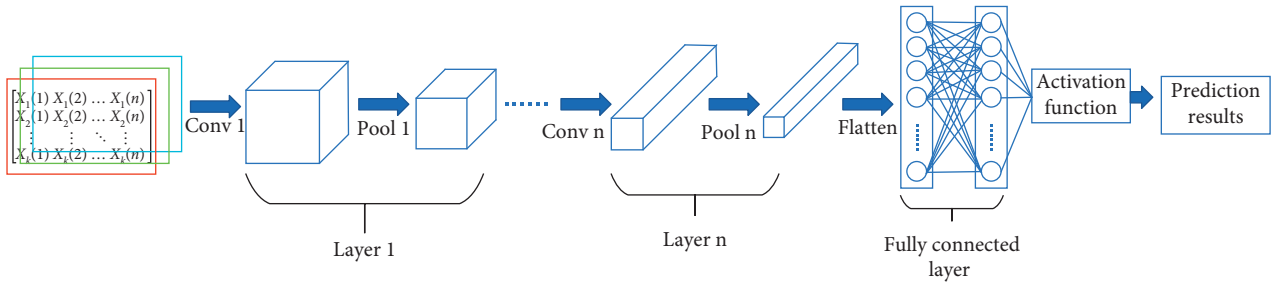


FIGURE 3: Convolution neural network structure.

**2.3. The GRU-CNN Hybrid Neural Networks.** To combine the advantages of the GRU module which can well process time sequence data and the advantages of the CNN module which is ideal for handling high-dimensional data, the GRU-CNN hybrid neural networks was proposed, and the structure of GRU-CNN hybrid neural networks is shown in Figure 4.

The framework of the proposed GRU-CNN hybrid neural networks consists of a GRU module and a CNN module. The inputs are the information of time sequence data and spatiotemporal matrixes collected from the power system; the outputs are the prediction of the future load value. In aspect of the CNN module, it is good at processing two-dimensional data, such as spatiotemporal matrixes and images. The CNN module uses local connection and shared weights to directly extract local features from the spatiotemporal matrixes data and obtain effective representation through the convolution layer and pooling layer. The structure of the CNN module contains two convolution layers and a flatten operation, and each convolution layer contains a convolution operation and a pooling operation. After the second pooling operation, the high-dimensional data are flattened into one-dimensional data, and the

outputs of the CNN module are connected with the fully connected layer. On the other hand, the aim of the GRU module is to capture the long-term dependency and the GRU module can learn useful information in the historical data for a long period through the memory cell, and the useless information will be forgotten by the forget gate. The inputs of GRU module are time sequence data; the GRU module contains many gated recurrent units, and the outputs of all these gated recurrent units are connected with the fully connected layer. Finally, the load predicting results can be obtained by calculating the mean value of all neurons in the fully connected layers. The flow chart of the GRU-CNN method is shown in Figure 5.

### 3. Experiments and Results

**3.1. Datasets Description.** In this experiment, the electric load dataset is provided by a power distribution network in Wuwei, Gansu province. The dataset was collected in August 2018. It contains 44640 samples which were recorded every minute for a total of 31 days. There are 31680 samples that are selected from these datasets as the training set. Also,

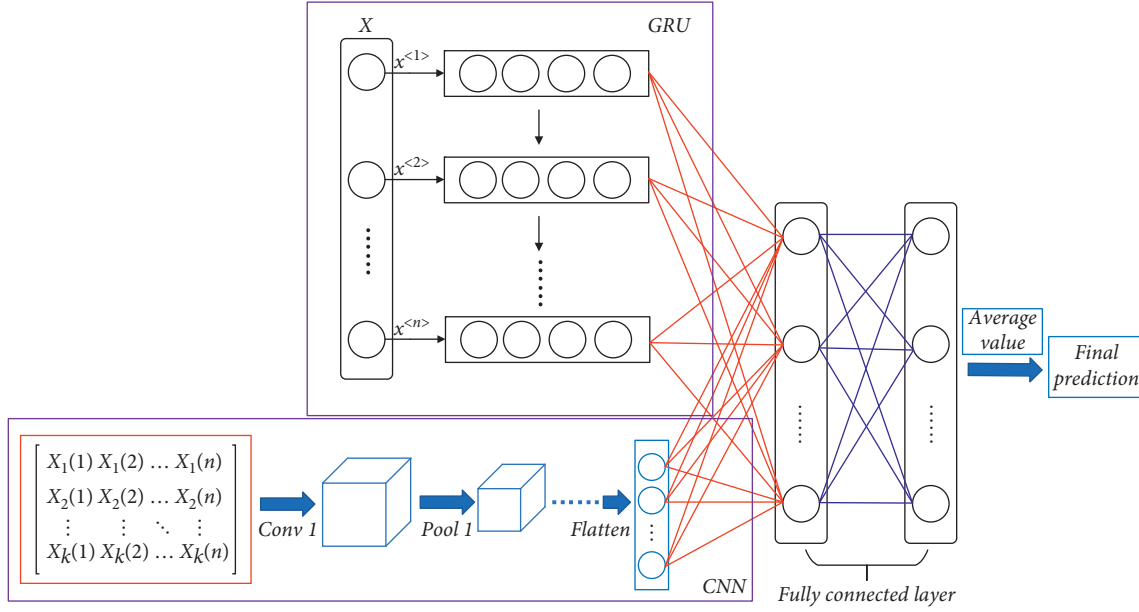


FIGURE 4: GRU-CNN hybrid neural networks.

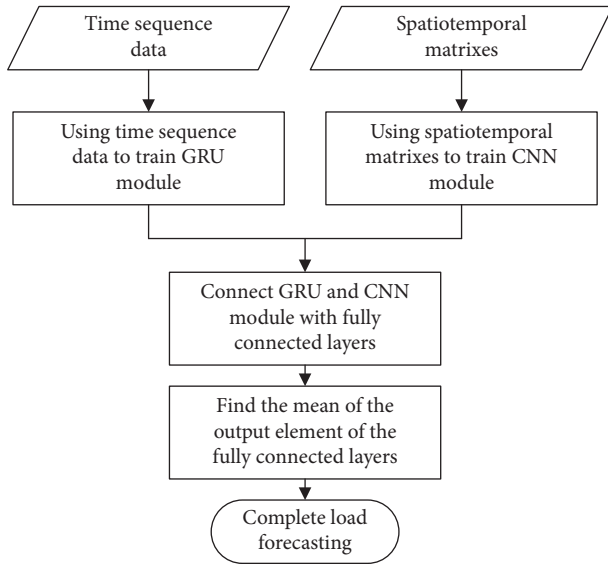


FIGURE 5: The flow chart of the GRU-CNN method.

11520 samples are utilized as the test set, which is divided into 8 tests, and each test contains 1440 samples. And the last 1440 samples are selected as the dev set. Each sample contains time sequence data such as temperature, holidays, and weather in Wuwei, and spatiotemporal matrixes data, which were collected by distributed smart electric meters in the Wuwei distribution network. Time sequence data of the samples were selected as the input of the GRU module, and spatiotemporal matrixes data of the samples were selected as the input of the CNN module.

**3.2. Model Evaluation Indexes.** To evaluate the performance of different predicting models, the absolute percentage error (APE), the mean absolute percentage error (MAPE),

and the root mean square error (RMSE) are introduced, and the formulas of APE, MAPE, and RMSE are shown as follows:

$$\begin{aligned} \text{APE} &= \left| \frac{y(i) - \hat{y}(i)}{y(i)} \right| \times 100\%, \\ \text{MAPE} &= \frac{1}{n} \sum_{i=1}^n \left| \frac{y(i) - \hat{y}(i)}{y(i)} \right| \times 100\%, \\ \text{RMSE} &= \sqrt{\frac{1}{n} \sum_{i=1}^n |y(i) - \hat{y}(i)|^2}, \end{aligned} \quad (4)$$

where  $n$  is the size of the training and test samples,  $y(i)$  and  $\hat{y}(i)$  are the actual value and the predicted value, respectively. The APE represents the ratio between the error and actual values at one predicting point, the MAPE represents the average of APE in all the test sets, and the RMSE is the sample standard deviation of differences between the predicted value and the actual value; the smaller the values of MAPE and RMSE, the better predicting performance the model achieves.

**3.3. Experimental Results and Analyses.** In order to evaluate the performance and stability of the GRU-CNN model, several most commonly used neural networks were chosen as contrast models, including back-propagation neural network (BPNN) model, convolutional neural network (CNN) model, and gated recurrent unit (GRU) model. The parameters of BPNN, GRU, CNN, and the GRU-CNN model were obtained by learning the training datasets, and the test datasets were fed to the four forecast models. A way to assess whether a prediction model is well trained and learned is the loss value of the loss function. Cross entropy



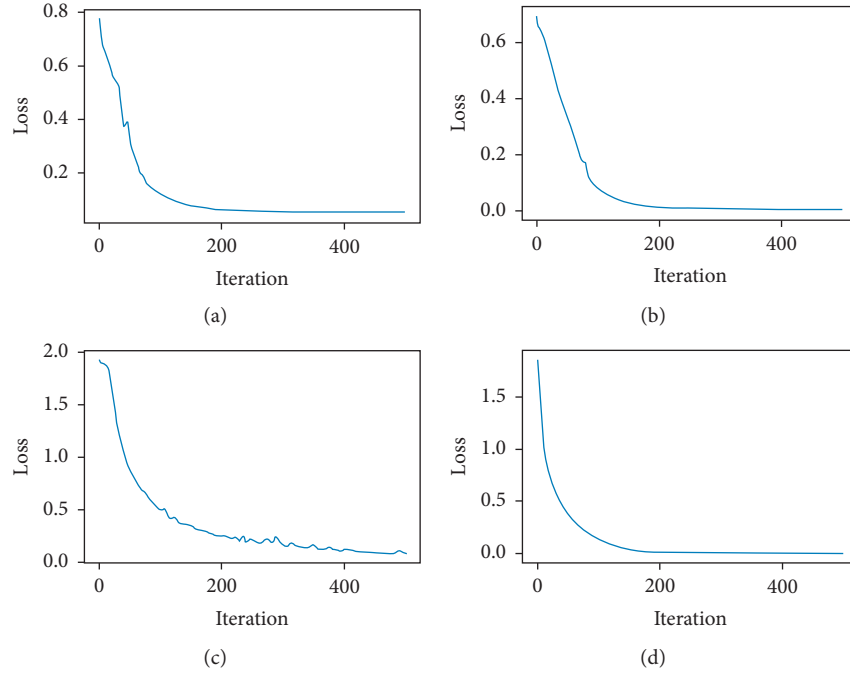


FIGURE 6: Neural networks' losses on the training set for 500 iterations. (a) BPNN. (b) GRU. (c) CNN. (d) GRU-CNN.

function was selected as loss function, which characterizes the distance between two probability distributions. The smaller the cross entropy, the closer the two probability distributions. The formula of cross entropy is shown as follows:

$$\text{Loss} = - \sum_{i=1}^n y_i \log y'_i, \quad (5)$$

where  $n$  is the size of the training samples and  $y_i$  and  $y'_i$  are the actual value and the predicted value, respectively. The closer the loss value is to zero, the better the prediction model fits the training set. The loss value of the four load prediction models is shown in Figure 6.

As seen from Figure 6, the BPNN, GRU, and GRU-CNN model converge fast, and the loss values of them are lower than 0.1 after 200 iterations, but the loss value of the CNN model is about 0.2 at the 200th iteration, and there are slight fluctuations in the CNN model. All these methods achieve a very small loss value after 500 iterations, and this means that all the methods can well learn the training set and achieve good performance at the test set. It can be found that training the training set more than 500 iterations, the loss value decreased extremely slowly. And then if they continued to increase the times of iteration, all these models perform well on the test set, but they cannot predict well on the dev set. Obviously, the models were overfitted, and so the times of iteration were settled at 500 times in this experiment.

There are eight groups of data which were trained and tested in the prediction of 24 hours load forecast for the BPNN, GRU, CNN, and GRU-CNN models, and the MAPE and RMSE of the load predicting results of the four models are shown in Table 1 and Table 2, respectively.

Where, test avg represents the average value of the eight testing results. As seen in Tables 1 and 2, the results of the BPNN model performs a little better than the GRU model, while the CNN model performs much better than the GRU model and the BPNN mode. Because most of the datasets in this paper are spatiotemporal matrixes data, there is a certain loss of information for flattening spatiotemporal matrixes data into one-dimensional time sequence data, and so the BPNN and GRU models cannot fully learn the features from the dataset. However, CNN is really good at handling high-dimensional data, and the spatiotemporal matrixes can be processed fully and rapidly. The proposed GRU-CNN model which is the fusion of the GRU model and the CNN model provides the best forecasting results. The test avg indexes of the GRU-CNN model are the minimum among all the four models. The GRU-CNN model can well learn both the time sequence data and the spatiotemporal matrixes and extract more features from the dataset. The average value of MAPE and RMSE of the GRU-CNN model is 2.8839% and 1203.23, respectively. The proposed model has an improved performance of 1.5842% which is better than the BPNN model, 1.7538% better than the GRU model, and 0.5051% better than the CNN model. For better visualization, the MAPE and RMSE lines of the four models are drawn in Figure 7.

As shown in Figure 7, the MAPE and RMSE lines of BPNN and GRU are very close, but the BPNN model performs better than the GRU model in test 3, test 4, test 5, test 6, and test 8. The lines of the CNN model fluctuate between BPNN lines and GRU-CNN lines, and it proves that the CNN model performs always better than the BPNN model and the GRU model but is inferior to the

TABLE 1: MAPE of the four prediction models.

Test	BPNN (%)	GRU (%)	CNN (%)	GRU-CNN (%)
Test 1	4.1245	4.0132	3.1973	3.1380
Test 2	4.9930	4.8877	3.0290	2.4428
Test 3	4.5197	4.9622	3.1971	2.8844
Test 4	4.1853	4.5516	3.8850	2.8688
Test 5	4.3401	4.7281	3.5980	2.9557
Test 6	4.3920	4.6418	3.5102	3.0947
Test 7	4.7342	4.7136	3.3682	2.8035
Test 8	4.4556	4.6037	3.3271	2.8835
Test avg	4.4681	4.6377	3.3890	2.8839

TABLE 2: RMSE of the four prediction models.

Test	BPNN	GRU	CNN	GRU-CNN
Test 1	1703.18	1658.33	1329.52	1305.62
Test 2	2053.19	2010.75	1261.69	1025.46
Test 3	1862.45	2040.77	1329.44	1203.43
Test 4	1727.68	1875.30	1606.66	1197.13
Test 5	1790.07	1946.43	1491.00	1232.15
Test 6	1810.98	1911.65	1455.62	1288.17
Test 7	1948.89	1940.59	1398.39	1170.82
Test 8	1836.61	1896.30	1381.83	1203.06
Test avg	1841.63	1910.02	1406.77	1203.23

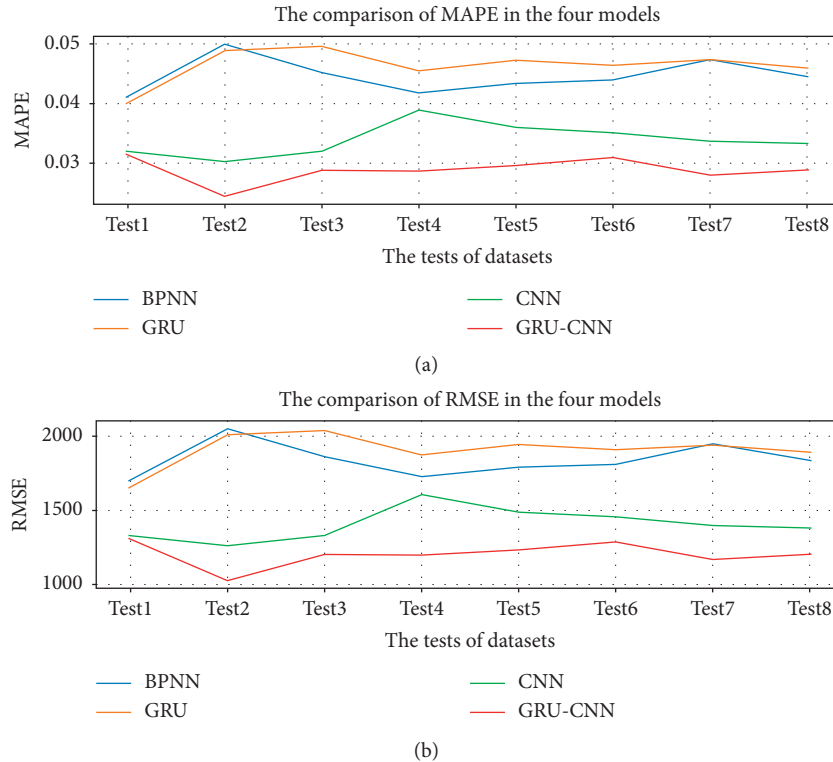


FIGURE 7: The comparison of MAPE (a) and RMSE (b) of the four models.

GRU-CNN model. The curves of the proposed GRU-CNN model are the minimum among all the eight tests, which demonstrates that the proposed GRU-CNN model achieves the best predicting performance among the four models.

In addition, the dev set, as the inputs of all the four models, was sent to the four trained models to predict the actual load. Finally, the predicting results of the four models were obtained and compared with actual load, as shown in Figure 8.

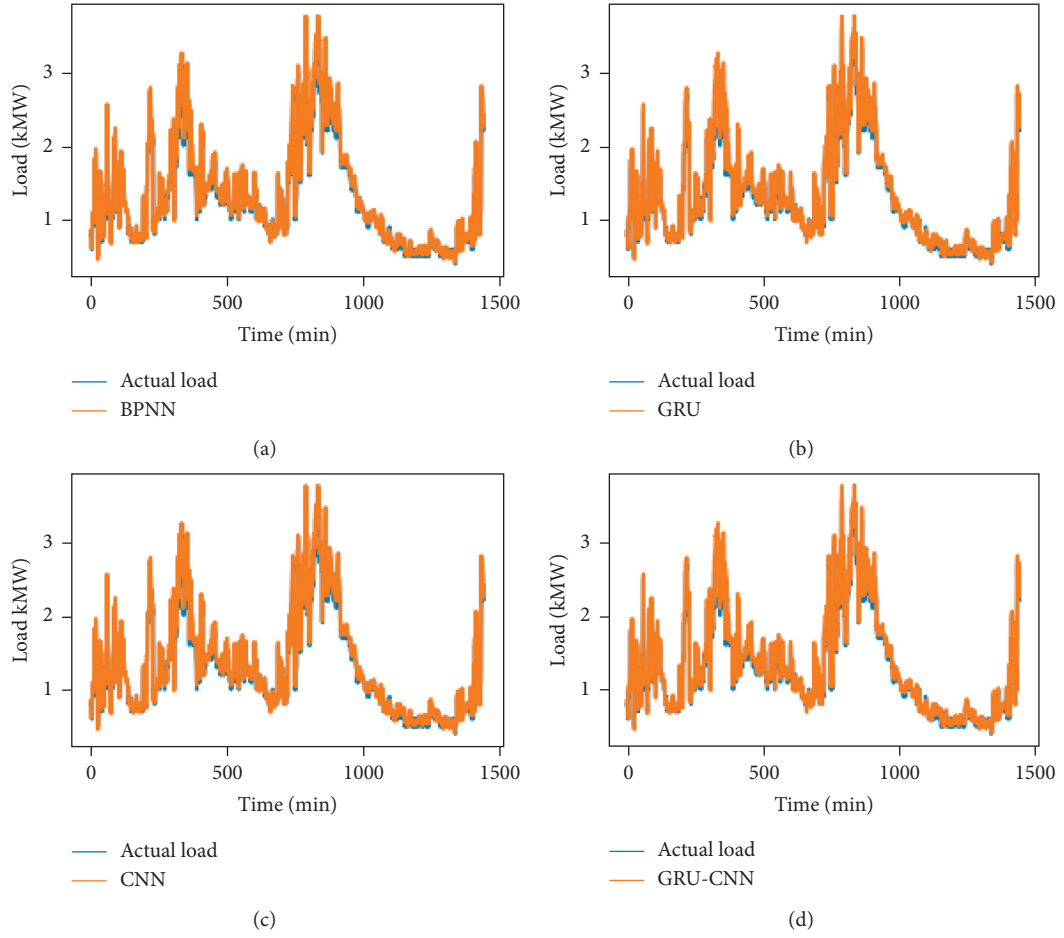


FIGURE 8: The prediction result of the GRU-CNN model. (a) BPNN. (b) GRU. (c) CNN. (d) GRU-CNN.

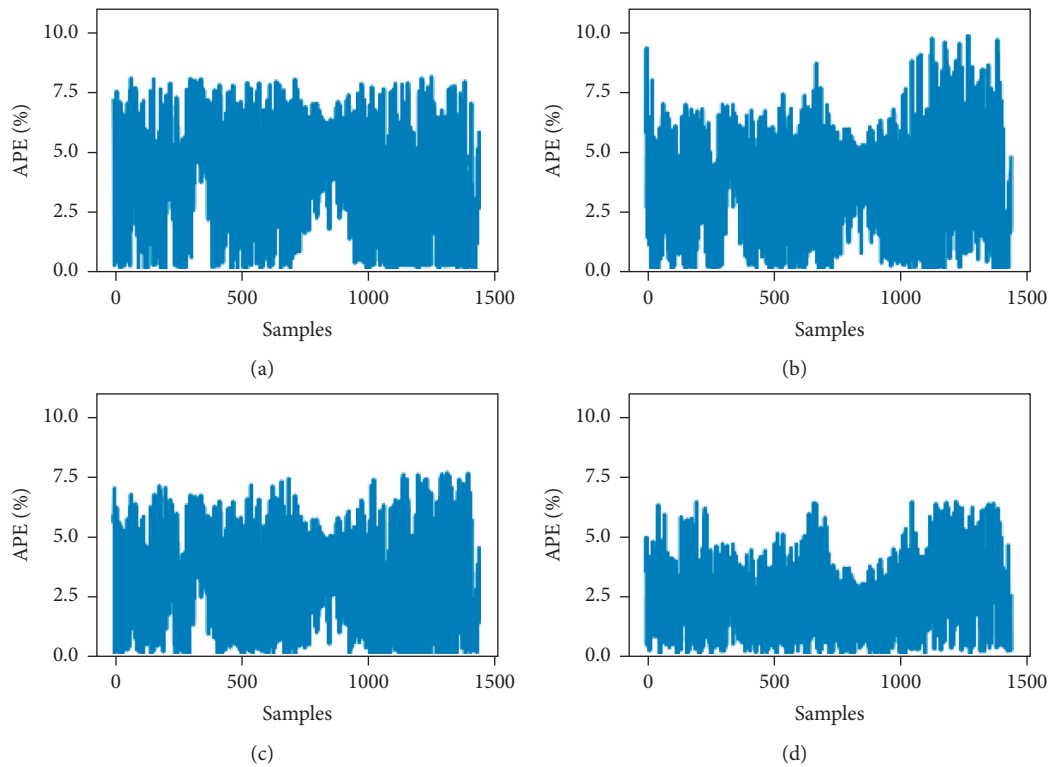


FIGURE 9: The absolute percentage errors of the GRU-CNN model. (a) The APE of BPNN. (b) The APE of GRU. (c) The APE of CNN. (d) The APE of GRU-CNN.



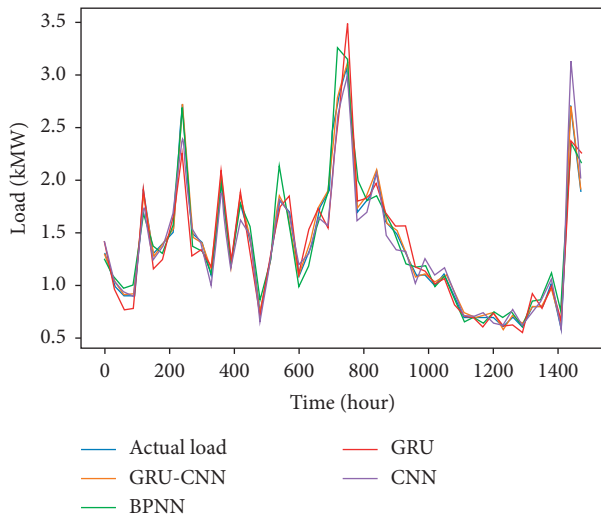


FIGURE 10: The forecasting load of four models and the actual load.

As shown in Figure 8, the prediction curves of all the four models fit well; whether in the peaks or troughs, the blue and orange lines almost overlap. Due to the large ranges of load changing, it is difficult to see the details of the predicting results, so the absolute percentage errors of the four models are drawn, respectively, in Figure 9.

As can be seen from Figure 9(b), the APE of GRU has the largest fluctuation, and it fluctuates from 0% to 10%, and the max APE of the GRU predicting model is 9.8578%, and the MAPE is 4.8781%. It can be seen from Figure 9(d) that the APE of the GRU-CNN model fluctuates the least from 0% to 7% and the max APE of the GRU-CNN predicting model is 6.418%, and the MAPE is 2.3034%. The max APEs of the BPNN model and the GRU model are 8.1306% and 7.605%, respectively, and the MAPEs are 4.0942% and 3.4581%, respectively. From the experiment of the dev set, the proposed GRU-CNN model achieves the most accurate and stable predicting result. And all the load forecasting results of the four models and the actual load are drawn in Figure 10.

In Figure 10, all the four models can predict well on the dev set, and the proposed GRU-CNN predicting method fits best. When predicting peaks, BPNN, GRU, and CNN perform not very well. BPNN, GRU, and CNN model cannot accurately analyze the load fluctuation law, resulting in a decrease in prediction accuracy. In contrast, the GRU-CNN model can effectively learn the load changing trend and accurately analyze the influence of the input characteristics on the load during the time of reaching the peaks to ensure the prediction accuracy.

#### 4. Conclusions

Aiming at improving the accuracy of STLF, the GRU-CNN hybrid neural network was proposed in this paper, which is based on the GRU model and the CNN model. The GRU module of the GRU-CNN model dedicates to process time sequence data, and the CNN module is good at processing spatiotemporal matrixes. The proposed model can predict

electrical load quickly and accurately by extracting features from variable data that affect the power system. In the real-world experiments of the Wuwei area electrical load forecasting, the proposed GRU-CNN model was compared with BPNN, GRU, and CNN models. The results show the proposed GRU-CNN model can well process both time sequence data and spatiotemporal matrixes data and can effectively extract the hidden feature of the datasets. The GRU-CNN model has the lowest value of MAPE and RMSE, and it demonstrates that the proposed GRU-CNN model achieves the best performance among all the four models.

In further study, the proposed model can be improved by collecting and adding more relevant factors that may affect the load change. At the same time, in order to train the neural network more effectively, a new loss function training method is designed to reduce the training time of the model. It can even be considered for random load forecasting, such as electric vehicles.

#### Data Availability

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

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### Acknowledgments

This project was supported by the National Natural Science Foundation of China (no. 51467009); the Basic Research Innovation Group Project of Gansu Province (no. 18JR3RA133).

#### References

- [1] B. Yildiz, J. I. Bilbao, and A. B. Sproul, "A review and analysis of regression and machine learning models on commercial building electricity load forecasting," *Renewable and Sustainable Energy Reviews*, vol. 73, pp. 1104–1122, 2017.
- [2] K. Ni, J. Wang, G. Tang, and D. Wei, "Research and application of a novel hybrid model based on a deep neural network for electricity load forecasting: a case study in Australia," *Energies*, vol. 12, no. 13, p. 2467, 2019.
- [3] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019.
- [4] L. Yang and H. Yang, "Analysis of different neural networks and a new architecture for short-term load forecasting," *Energies*, vol. 12, no. 8, 2019.
- [5] C. Tian, J. Ma, C. Zhang, and P. Zhan, "A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network," *Energies*, vol. 11, no. 12, 2018.
- [6] J. Zhang, Y.-M. Wei, D. Li, Z. Tan, and J. Zhou, "Short term electricity load forecasting using a hybrid model," *Energy*, vol. 158, pp. 774–781, 2018.
- [7] N. Ghadimi, A. Akbarimajd, H. Shayeghi, and O. Abedinia, "Two stage forecast engine with feature selection technique

- and improved meta-heuristic algorithm for electricity load forecasting," *Energy*, vol. 161, pp. 130–142, 2018.
- [8] A. Goia, C. May, and G. Fusai, "Functional clustering and linear regression for peak load forecasting," *International Journal of Forecasting*, vol. 26, no. 4, pp. 700–711, 2010.
  - [9] A. H. Nury, K. Hasan, and M. J. B. Alam, "Comparative study of wavelet-ARIMA and wavelet-ANN models for temperature time series data in northeastern Bangladesh," *Journal of King Saud University-Science*, vol. 29, no. 1, pp. 47–61, 2017.
  - [10] C.-M. Lee and C.-N. Ko, "Short-term load forecasting using lifting scheme and ARIMA models," *Expert Systems With Applications*, vol. 38, no. 5, pp. 5902–5911, 2011.
  - [11] G.-Y. Chen, M. Gan, and G.-L. Chen, "Generalized exponential autoregressive models for nonlinear time series: stationarity, estimation and applications," *Information Sciences*, vol. 438, pp. 46–57, 2018.
  - [12] L. Hernández, C. Baladrón, J. Aguiar et al., "Artificial neural network for short-term load forecasting in distribution systems," *Energies*, vol. 7, no. 3, pp. 1576–1598, 2014.
  - [13] J.-p. Liu and C.-l. Li, "The short-term power load forecasting based on sperm whale algorithm and wavelet least square support vector machine with DWT-IR for feature selection," *Sustainability*, vol. 9, no. 7, 2017.
  - [14] A. Jadidi, R. Menezes, N. de Souza, and A. C. de Castro Lima, "Short-term electric power demand forecasting using NSGA II-anfis model," *Energies*, vol. 12, no. 10, p. 1891, 2019.
  - [15] D. Niu, Y. Wang, and D. D. Wu, "Power load forecasting using support vector machine and ant colony optimization," *Expert Systems With Applications*, vol. 37, no. 3, pp. 2531–2539, 2010.
  - [16] H. H. Çevik and M. Çunkaş, "Short-term load forecasting using fuzzy logic and ANFIS," *Neural Computing and Applications*, vol. 26, no. 6, pp. 1355–1367, 2015.
  - [17] Z. Xiao, S.-J. Ye, B. Zhong, and C.-X. Sun, "BP neural network with rough set for short term load forecasting," *Expert Systems With Applications*, vol. 36, no. 1, pp. 273–279, 2009.
  - [18] X. Sun, Z. Ouyang, and D. Yue, "Short-term load forecasting model based on multi-label and BPNN," *Communications in Computer and Information Science*, Springer, New York, NY, USA, pp. 263–272, 2017.
  - [19] X. Xiong, P. Zhou, and A. Chen, "Asymptotic Normality of the Local Linear Estimation of the Conditional Density for Functional Time Series Data," *Communication In Statistics-Theory And Methods*, vol. 47, no. 14, pp. 3418–3440, 2017.
  - [20] G. Li, H. Wang, S. Zhang, J. Xin, and H. Liu, "Recurrent neural networks based photovoltaic power forecasting approach," *Energies*, vol. 12, no. 13, 2019.
  - [21] D. Yi, S. Bu, and I. Kim, "An enhanced algorithm of RNN using trend in time-series," *Symmetry*, vol. 11, no. 7, p. 912, 2019.
  - [22] T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis, and P. S. Dokopoulos, "Long-term wind speed and power forecasting using local recurrent neural network models," *IEEE Transactions on Energy Conversion*, vol. 21, no. 1, pp. 273–284, 2006.
  - [23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
  - [24] A. S. Santra and J.-L. Lin, "Integrating long short-term memory and genetic algorithm for short-term load forecasting," *Energies*, vol. 12, no. 11, 2019.
  - [25] J. Chung, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *Clinical Orthopaedics and Related Research*, vol. 3555, 2014.
  - [26] H. Cheng, Z. Xie, L. Wu, Z. Yu, and R. Li, "Data prediction model in wireless sensor networks based on bidirectional LSTM," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, Article ID 203(2019), 2019.
  - [27] W. Li, T. Logenthiran, and W. L. Woo, "Multi-GRU prediction system for electricity generation's planning and operation," *IET Generation Transmission & Distribution*, vol. 13, no. 9, pp. 1630–1637, 2019.
  - [28] Y. Wang, W. Liao, and Y. Chang, "Gated recurrent unit network-based short-term photovoltaic forecasting," *Energies*, vol. 11, no. 8, 2018.
  - [29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
  - [30] W. Liu, Y. Song, D. Chen et al., "Deformable object tracking with gated fusion," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3766–3777, 2019.
  - [31] P.-H. Kuo and C.-J. Huang, "A high precision artificial neural networks model for short-term energy load forecasting," *Energies*, vol. 11, no. 1, 2018.
  - [32] X. Ke, L. Shi, W. Guo, and D. Chen, "Multi-dimensional traffic congestion detection based on fusion of visual features and convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2157–2170, 2019.
  - [33] X. Ke, J. Zou, and Y. Niu, "End-to-End automatic image annotation based on deep CNN and multi-label data augmentation," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2093–2106, 2019.
  - [34] X. Gao, X. Li, B. Zhao, W. Ji, X. Jing, and Y. He, "Short-term electricity load forecasting model based on EMD-GRU with feature selection," *Energies*, vol. 12, no. 6, 2019.