mathematics

MDPI

Article

Power Load Forecast Based on CS-LSTM Neural Network

Lijia Han [1], Xiaohong Wang [1,*], Yin Yu [2] and Duan Wang [3]

[1] Institute of Information and Computation, School of Mathematics and Physics, North China Electric Power University, Beijing 102206, China; hljmath@ncepu.edu.cn
[2] Nanjing Laisi Information Technology Co., Ltd., Nanjing 210014, China; guwole@163.com
[3] Department of Mathematics, Nuclear Industry College, Beijing 102413, China; wangduan@niunep.com
* Correspondence: wangxiaohong2024@126.com

Abstract: Load forecast is the foundation of power system operation and planning. The forecast results can guide the power system economic dispatch and security analysis. In order to improve the accuracy of load forecast, this paper proposes a forecasting model based on the combination of the cuckoo search (CS) algorithm and the long short-term memory (LSTM) neural network. Load data are specific data with time series characteristics and periodicity, and the LSTM algorithm can control the information added or discarded through the forgetting gate, so as to realize the function of forgetting or memorizing. Therefore, the use of the LSTM algorithm for load forecast is more effective. The CS algorithm can perform global search better and does not easily fall into local optima. The CS-LSTM forecasting model, where CS algorithm is used to optimize the hyper-parameters of the LSTM model, has a better forecasting effect and is more feasible. Simulation results show that the CS-LSTM model has higher forecasting accuracy than the standard LSTM model, the PSO-LSTM model, and the GA-LSTM model.

Keywords: load forecast; long short-term memory neural network; cuckoo search

MSC: 68T07; 68W40

Citation: Han, L.; Wang, X.; Yu, Y.; Wang, D. Power Load Forecast Based on CS-LSTM Neural Network. *Mathematics* **2024**, *12*, 1402. https://doi.org/10.3390/math12091402

Academic Editor: Pedro A. Castillo Valdivieso

Received: 18 March 2024
Revised: 18 April 2024
Accepted: 30 April 2024
Published: 3 May 2024

1. Introduction

Power load forecasting plays a very important role in the power system. Accurate load prediction helps to rationally arrange the start and stop of generator sets, maintain the safety and stability of the power grid, and improve social and economic benefits [1].

Load forecast has mainly experienced three development stages [2–5]: the manual forecasting stage, statistical forecasting stage, and the machine forecasting stage. The power load series is a time series, which has obvious periodicity, reflected in the daily periodicity, weekly periodicity and seasonal periodicity, and the power load between holidays and working days also has different characteristics.

Over the past few decades, numerous methods have been applied to compute an accurate load forecasting. Traditional methods include inter sequence methods [6], autoregressive methods [7], Kalman filters [8], and so on. Intelligent algorithms are mainly machine learning algorithms represented by Support Vector Machines [9] and neural networks [10]. In [11,12], the authors addressed the systematic design of a multistage artificial-neural-network-based short-term load forecaster (ANNSTLF). In [13], the authors presented an expert system-based algorithm as an alternative. In [14], the authors presented an SVR-based electric load forecasting model that applied a novel algorithm, namely Chaotic Ant Swarm optimization (CAS), to improve the forecasting performance by searching for a suitable combination of parameters. With the emergence of deep learning, many scholars have turned their attention to Deep Confidence Networks, Convolutional Neural Networks, Recurrent Neural Networks, etc. A Recurrent Neural Network (RNN) is a kind of neural network commonly used to process time series data, but it is prone to

gradient explosion when it remembers distant information. Therefore, the direct use of RNNs to predict power load data cannot meet the requirements of high accuracy in power load prediction. In [15], an improved gray model was used to weigh load data, select appropriate initial conditions, and self-adaptively optimize model parameters. Compared with the traditional gray model, this model had a higher accuracy. In [16], an improved GRU model was used to forecast the power load. The GRU model is a variant of the RNN model. Experimental results showed that this method has a high fitting degree, fast convergence rate, and a good forecasting effect. A long short-term memory neural network (LSTM) is a kind of improved RNN, which can control the loss or increase in information through the forgetting gate, so as to realize the function of forgetting or memory, and can solve the problem of "forgetfulness" in the process of RNN training. Reference [17] used the LSTM to forecast the power load, and the results showed that the proposed method can effectively improve the accuracy of load prediction.

However, a single prediction method often fails to meet the requirements of high precision. At present, some hybrid machine learning methods are beginning to be widely used, and the prediction accuracy of the hybrid method is higher than that of the single method. Reference [18] used a combination of Principal Component Analysis (PCA) and Support Vector Machine (SVW) to improve the accuracy of power load prediction. The results showed that compared with the standard SVM algorithm, the PCA-SVM model can reduce the dimension of the sample set and improve the accuracy of load forecast. In [19], the combined prediction model of SD selection, EMD and LSTM was used to forecast short-term power loads. The forecasting accuracy of the hybrid EMD-LSTM algorithm was higher than that of the LSTM, SD-LSTM, EMD-LSTM, ARIMA, BPNN, and SVR models. In [20], an improved PSO-BP neural network was used to predict the short-term power load. Experimental results showed that the improved PSO-BP algorithm has faster convergence speed and higher forecasting accuracy than the conventional BP algorithm. A model based on the combination of the Prophet addition model and LSTM, the experimental results showed that the proposed method has higher forecasting accuracy than the traditional load forecasting method and the standard Prophet and LSTM load forecasting method [21]. In addition, studies on short-term load forecasting can be found in [22–24] and so on.

Many studies on power load forecasting use some optimization methods to optimize the hyperparameters of the standard model. At present, heuristic search algorithms are often used to solve optimization problems, including Particle Swarm Optimization (PSO), the Simulated Annealing Algorithm (SAA), and the genetic algorithm (GA) and so on, but they often fall into local optimal solutions, and the quality of the solutions cannot be guaranteed. In [25], the authors presented a study of short-term power load forecasting based on improved Particle Swarm Optimization and a Recurrent Neural Network. In view of the shortcomings of a cyclic neural networks in power load forecasting, which are prone to local minima and weak in terms of global search ability, the idea of introducing Simulated Annealing Algorithm probability mutation into the Particle Swarm Optimization Algorithm search process is proposed. Similarly, in [26], a method based on the Support Vector Machine and genetic algorithm was proposed for the power system load forecasting. The cuckoo search (CS) algorithm, as a new heuristic search algorithm [27] is essentially a bionic algorithm, which is based on the parasitic brood rearing behavior of cuckoos, and is enhanced by Levy flight. The cuckoo search algorithm can carry out a global search better, so the solution does not easily to fall into local optimization, which makes the model prediction accuracy higher. In [28], a BP neural network optimized by the cuckoo algorithm was used to predict the health state of lithium batteries, and the results showed that the CS-BP algorithm resulted in fewer errors. Ref. [29] used the improved CS algorithm to optimize power system scheduling, and the results showed that the improved algorithm has a higher calculation accuracy. Ref. [30] used the CS algorithm to study odor sources, and the results showed that the CS algorithm can locate odor sources more accurately.

According to the time series characteristics of power load data, this paper innovatively combines the CS algorithm with LSTM to propose a power load forecast model based

on CS-LSTM and verifies the rationality of the prediction model through examples. The experimental results show that using the CS algorithm to optimize the hyperparameters of the LSTM model, the established CS-LSTM prediction model has better prediction accuracy and more feasibility than the standard LSTM, PSO-LSTM, and GA-LSTM prediction models.

2. Method and Principle

2.1. LSTM Neural Network

When the standard RNN deals with long-term dependence, the problem of gradient hour or gradient explosion will occur. In order to solve this problem, Hochreiter et al. [31] proposed an improved RNN structure, which is the LSTM neural network. Figure 1 shows the basic structure of the LSTM memory cell.

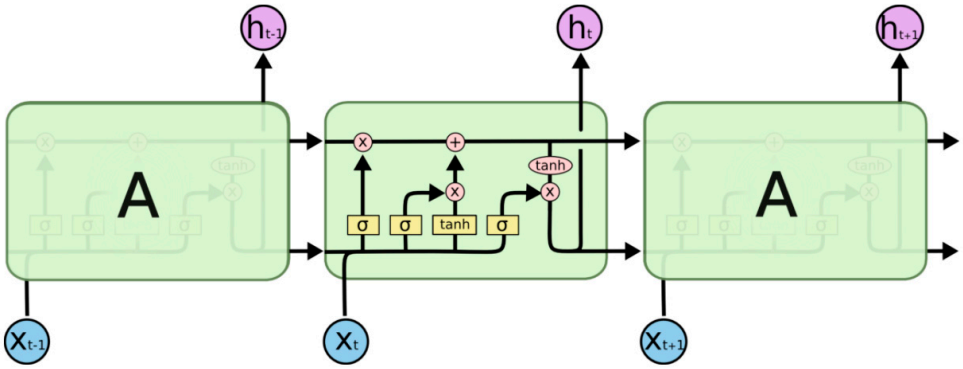

Figure 1. LSTM memory cell structure.

In Figure 1, x_{t-1}, x_t, x_{t+1} represent the input values, h_{t-1}, h_t, h_{t+1} represent the output values, and σ, tanh represent the incentive functions.

Different from RNN, the LSTM model adds the input gate and forgetting gate to neurons. The forgetting gate is used to solve the problem of gradient disappearance or gradient explosion. It is mainly used to confirm which input information needs to be updated and which information needs to be retained [31].

The LSTM operation process is as follows:

$$\begin{gathered} f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f), \\ i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i). \\ \widetilde{C}_t = \tanh(W_C \times [h_{t-1}, x_t] + b_C), \\ C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t. \\ o_t = \sigma(W_o[h_{t-1}, x_t] + b_0), \\ h_t = o_t \tanh(C_t). \end{gathered} \tag{1}$$

In the above formula, W_f, W_i, W_C, W_o are the weight matrix of each structure of the LSTM neural network; respectively, b_f, b_i, b_C, b_o are the offset vectors corresponding to each structure; $C_t, \widetilde{C}_t, C_{t-1}$ indicate the cell state.

2.2. Cuckoo Search

The cuckoo search (CS) algorithm, as a new heuristic search algorithm [26], is essentially a bionic algorithm. Its principle is based on the parasitic brood rearing behavior of cuckoos, which is enhanced by Lévy flight.

Traditional optimization methods either have a short computation time but low accuracy, or high accuracy but high complexity. For more complex optimization problems, heuristic search algorithms can help researchers achieve ideal results in a relatively short period of time.

In response to the fact that traditional heuristic search algorithms often fall into local optima, this chapter uses the cuckoo search (CS) algorithm to optimize and tune the model parameters.

2.2.1. Definition of Cuckoo Search

The CS algorithm mainly has global search ability and local search ability, which are mainly controlled by switching random walk (*pa*) probability. *pa* is expressed as the proportion of the local search time to the total search time, for example, when $pa = 1/2$, the search time of the local search and the search time of the global search are equally divided. Therefore, when the appropriate *pa* value is set, the distribution of local search and global search in this search space is more reasonable, and the global optimal solution can be found more easily, which is more efficient than other heuristic search algorithms.

The specific implementation process of the cuckoo search algorithm based on the Levy flight is shown in the following Algorithm 1:

Algorithm 1: CS Algorithm Based on Levy Flight

Objective function $f(x), x = (x_1, \ldots, x_d)^T$.
An initial population that produces n hostsx_i.
While (t < MaxGeneration) or (stop criterion) do
  Take a cuckoo at random.
  A solution is generated by Levy flight.
  Evaluate the quality or objective function value of the solution f_i.
  Randomly select one nest from n nests (Set to j).
  If $f_i < f_j$ do
    Replace j with solution i.
  End if.
  Some (*pa*) bad nests are abandoned.
  A new nest solution Formula (6-3) is generated.
  Preserve the best solution (or the nest of high-quality solution).
  Arrange the solution to find out the best at present.
  Update $t \leftarrow t + 1$.
End while.
Post-processing and visualization.

2.2.2. Principle of Cuckoo Search

The CS algorithm is based on Levy flight to carry out global random walk, and its formula is shown in Equation (2):

$$X_{g+1,j} = X_{g,j} + \alpha \otimes levy(\lambda). \tag{2}$$

In Formula 2, $g = 1, 2, \ldots, M$ represents the current iterations, and M represents the maximum number of iterations. $\alpha > 0$ is the step size scaling factor, $\otimes$ represents the dot multiplication operation; $levy(\lambda)$ represents Levy flight, which is a random path [24]. The theory shows that when the μ and σ^2 of Levy flight are infinite, the CS algorithm can significantly improve the search efficiency compared to other algorithms based on a standard Gauss process. Further combined with local search ability and global convergence, the CS algorithm can be more efficient.

The formula for Levy flight is shown in Equation (3):

$$levy(\lambda) = t^{-\lambda}. \tag{3}$$

In Formula (3), t represents the number of iterations.

The Levy flight is actually a classification process, which involves multiple steps to determine the distance from a stable distribution to a random walk. Statistically, the simplified Mantogna algorithm can be used in this process to generate random numbers with symmetric Levy distribution, mainly to generate random numbers based on symmetric

Levy stable distribution [22]. Using this algorithm, the specific implementation steps of replacing the old vector with the new vector are shown in the following Formula (4):

$$x_{i,new} = W \cdot X_i + stepsize \cdot rand. \tag{4}$$

In Formula (4), W represents the inertia weight, and its value can be set according to Formula (5):

$$W = W_{\max} - \left(\frac{W_{\max} - W_{\min}}{iter_{\max}} \right) / iter. \tag{5}$$

In addition, *stepsize* is as shown in Equation (6):

$$stepsize = \gamma \cdot step \cdot (X_i - X_{best}). \tag{6}$$

The difference factor $(X_i - X_{best})$ shows that if the current solution is close to the optimal solution, *stepsize* remains unchanged. Here, the step size is multiplied by γ, and the setting of γ is related to the activity of the point, for example, $\gamma = 0.01$ indicates the $step/100$ and represents the step length [23]; the setting of step size and multiplication is related to the activity of the point. Among them, the step factor for finding a new nest is *step*, and its specific equation is as follows:

$$step = (\sigma(\beta) \cdot rand)^{1/\beta}. \tag{7}$$

In Formula (7), *rand* represents the uniformly distributed random number in the interval [0,1]; β is a constant and satisfies $1 \leq \beta \leq 3$.

However, appropriate settings must be set to avoid new solutions that may jump out of the scope of the solution due to aggressive Levy flying [22,25].

The standard deviation $\alpha(\beta)$ can be calculated by Equation (8):

$$\alpha(\beta) = \left(\frac{\Gamma(1+\beta) \cdot \sin\left(\pi \cdot \frac{\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}}. \tag{8}$$

In Equation (8), the range of β is [0.3,1.99]; Γ represents the gamma function. It has been shown that when the absolute value of the step factor is much larger than 0, the distribution obtained by Equation (8) is the same as the row of the Lévy distribution.

The updating process of the optimal solution is shown in Equation (9):

$$X_{best} \leftarrow f(X_{best}) < f(X_i). \tag{9}$$

3. Experimental Setup

3.1. Data Preprocessing

In order to reduce the effect of seasonality on the power load data, in this paper, we select data from 1 January to 1 March 2010 in Australia, with half-hour time intervals, for a total of 4320 power load data values. The dataset is divided into training and test sets. The data from the first 83 days are used as the training set, and the data from the last 7 days are used as the test set to fit the models and build the prediction models for BP, LSTM and CS-LSTM, PSO-LSTM and GA-LSTM, respectively.

Considering that the main factors affecting the power load include temperature, humidity, season and holidays, etc., in addition to the power load data of 48 moments per day, five factors are added: maximum temperature, minimum temperature, average temperature, average humidity and type of week per day, where the relative week is represented by a three-digit binary from Monday to Sunday. Since the data used are from Australia, considering the differences in holidays between Australia and China, only the week is changed to binary representation to distinguish workdays and weekends.

Meanwhile, the power load data is normalized to a range between [0,1]. The specific calculation process is as follows:

$$x_{i,st} = \frac{x_i - x_{i,\min}}{x_{i,\max} - x_{i,\min}} (i = 1, 2, \ldots, n). \tag{10}$$

where $x_{i,st}$ denotes the standardized load data, x_i denotes the power load data of the day i, $x_{i,\min}$, $x_{i,\max}$ denote, respectively, the minimum and maximum power load of the day i, and n indicates the total number of days.

3.2. Evaluation Measures

In order to evaluate the performance of the prediction method in this paper, the following error indexes are used, and the relevant calculation formulas are as follows:

(1) Average absolute percentage error (MAPE):

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right|. \tag{11}$$

(2) Mean square error (MAE):

$$\text{MAE} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}. \tag{12}$$

(3) Root mean square error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}. \tag{13}$$

In the above Formulas (11)–(13), $\hat{y}_i$ and y_i represent, respectively, the predicted value and the true value of the prediction time, and n represents the total of forecast samples. MAE and RMSE measure the absolute size of the deviation from the predicted value, while MAPE measures the relative size of the deviation (that is, percentage).

3.3. Experimental Design

We evaluate the effectiveness of CS-LSTM over real-world datasets from the credit platform by comparing it with other models. The models selected for comparison include single and optimized models. Among them, single models include the BP model and the LSTM model. Optimization models include the CS-LSTM model, the GA-LSTM model and the PSO-LSTM model. Through the establishment of these comparison models and the evaluation measures of the models, the effectiveness of the CS-LSTM model proposed in this paper is verified.

4. Experimental Results

4.1. Power Load Forecasting Based on CS-LSTM

The dataset of 90 samples was divided into 83 training sets and 7 test sets. We used the 53 features of the previous day and the first five features of the predicted day as inputs, and the load of the predicted 48 time points of the day as outputs. We substitute the processed data into the CS-LSTM model and used the CS algorithm based on Levy flight to solve the optimal LSTM hyperparameters; the parameters are set as shown in Table 1:

The meanings of each hyperparameter in the table above are as follows:

The number of training epochs for all samples is 10, the samples collected in one iteration is 16 and the learning rate is 0.01. There are two layers of hidden layers and there are 100 nodes in each layer. The variables input each time are five weather conditions indicators of the previous day, 48 standardized power load data and five weather conditions indicators of the same day. The output is the value of 48 power loads predicted for the day.

Table 1. The LSTM neural network model hyperparameter settings.

Hyperparameter	Value
num_epochs	10
batch_size	16
alpha	0.01
hidden_nodes0	100
hidden_nodes	100
input_features	58
output_class	48

The first point in the future is taken as a "historical value" using prediction. After predicting the model with the LSTM, in order to make the model more accurate, the CS algorithm is used to adjust the hyperparameters of the LSTM model. Among them, the parameters of the CS algorithm are set in Table 2:

Table 2. The CS algorithm parameter settings.

Parameter	Set Value
β	1.5
pa	0.25
λ	1
Number of nests	40
Max_iter	10

The meanings of the parameters in the above table are as follows:

After debugging the code in Python many times, it is found that the bird's nest is 40, the maximum number of iterations is 10, and the discovery rate pa is 0.25; generally, β and λ are 1.5 and 1, and the CS model has the fastest convergence speed [24,32].

The upper and lower bounds of the LSTM neural network hyperparameter settings are in Table 3:

Table 3. The LSTM neural network hyperparameter settings.

Hyperparameter	Upper Bound	Lower Bound
LSTM Number of iteration	500	100
Learning rate	0.01	0.001
Number of nodes in the first hidden layer	200	100
Number of nodes in the second hidden layer	200	100

The CS algorithm is used to optimize the hyperparameters of LSTM neural network. After ten iterations, the results are shown in Table 4:

Table 4. The super-parameter value of the optimal solution in each iteration.

Number of Iterations of CS Algorithm	The LSTM Iterative Algebra	Learning Rate	Number of Nodes in the First Layer	Number of Nodes in the Second Layer
2	252	0.0059	166	89
3	224	0.0069	187	120
4	224	0.0069	187	120
5	152	0.0069	187	79
6	260	0.0086	87	141
7	260	0.0086	87	141
8	260	0.0086	87	141
9	260	0.0086	87	141
10	260	0.0063	87	155

The changes in the hyperparameters during each iteration are in Figure 2:

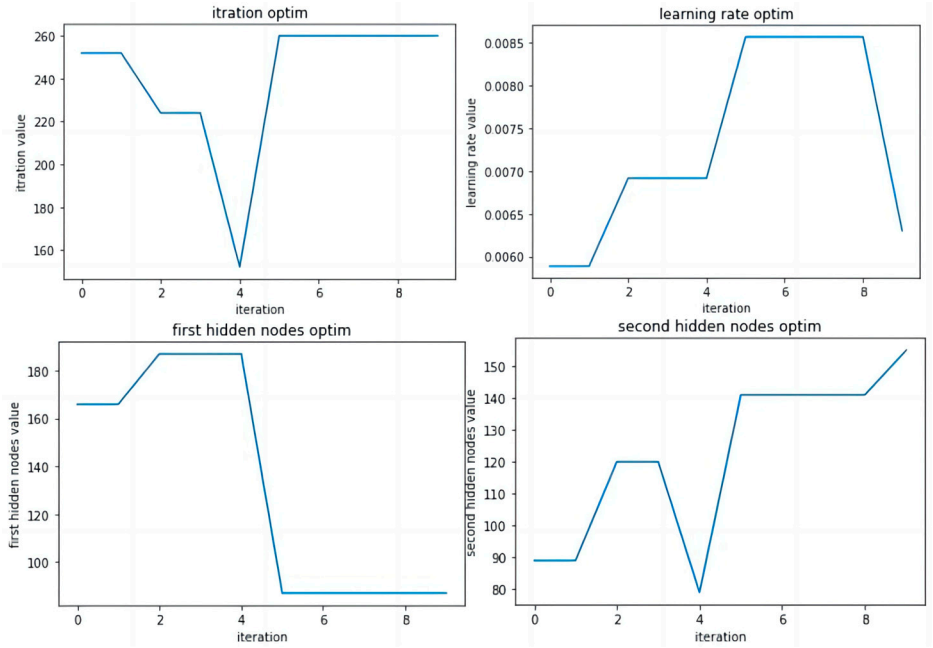

Figure 2. Each hyperparameter variation curve.

Taking the best individual training, and outputting the prediction results, the results are in Figure 3:

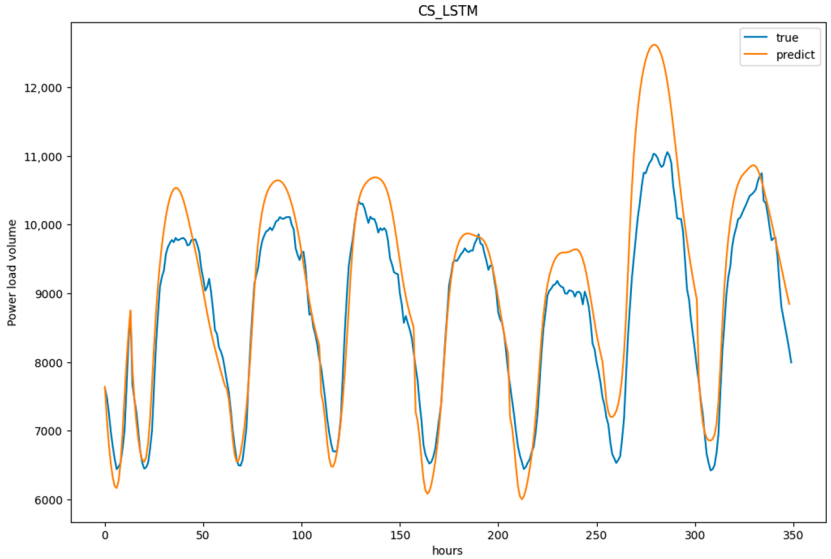

Figure 3. Prediction result of the CS-LSTM model.

The number of iterations, learning rate and number of hidden layer nodes of the LSTM are optimized. After setting the corresponding parameters, the upper and lower bounds are [100,500], [0.01,0.001] and [100,200]. Python is used to train and fit the model, and the loss curve is shown in Figure 4:

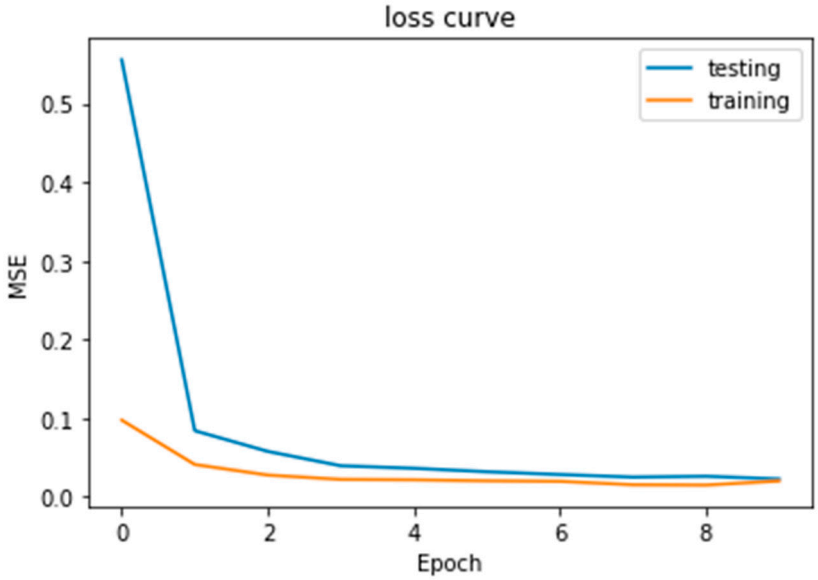

Figure 4. Loss curve.

It can be seen that with the increase in the number of iterations, the errors of the training set and the test set have been decreasing and gradually converging, indicating that the model has a good fitting effect.

In order to evaluate the model from the perspective of load forecasting and evaluation of the power system, Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), Root Mean Square Error (RMSE) are used as the evaluation indexes. The lower the value, the better the model fit and the better the accuracy.

The MAPE, MSE, and MSE values of the model are shown in the Table 5 below:

Table 5. Comparison of evaluation indicators of prediction models.

Evaluation Index	CS-LSTM Model
MAPE	1.82%
RMSE	232.22
MAE	170.99

In this section, the cuckoo algorithm is used to optimize the hyperparameter in the LSTM model, which makes the prediction results more accurate. The final prediction results of the model are better than the unoptimized model, which also confirms the feasibility of this model. In the next section, we will compare the unoptimized models to further confirm the feasibility of the CS-LSTM model.

4.2. Compared with Single Models (the BP, the LSTM)

The BP neural network is the most basic neural network; its output results are propagated forward, and the error is carried out by back propagation. The BP neural network is supervised learning, which is usually used in function approximation, pattern recognition, classification, data compression and data mining.

We use the 53 features of the previous day and the first 5 features of the predicted day as inputs and the load of the predicted 48 moments of the day as outputs.

The final fitting result is shown in Figure 5:

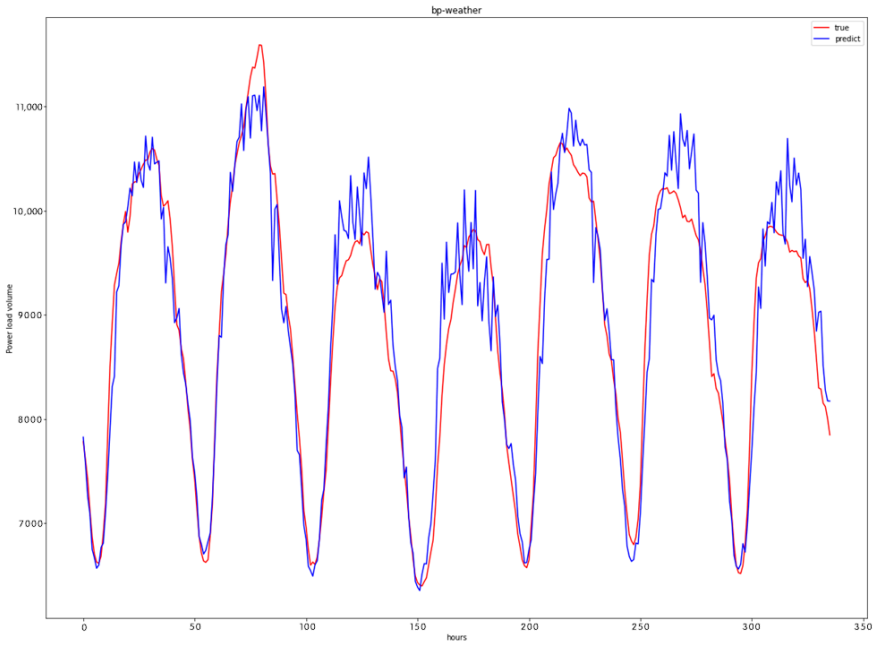

Figure 5. Fitting results of BP neural network model (including weather factors).

According to the discussion of the LSTM model in the second chapter, the data are substituted into the LSTM neural network model for prediction and analysis. TensorFlow in Python is used to predict the model.

After 10 iterations, as can be seen from the figure, both training loss and testing loss have been converged and the difference between them is very small. The prediction shows that the model fits well. Figure 6 below shows a comparison between the predicted results and the real values:

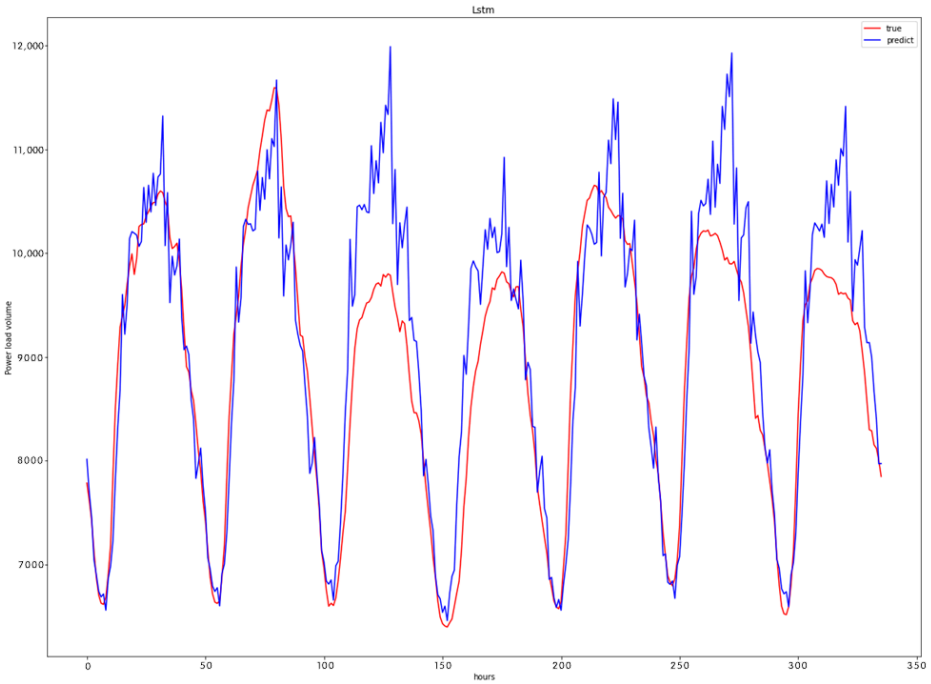

Figure 6. Fitting results of LSTM neural network model.

The red curve is the real value curve and the blue curve is the predicted value curve. The fitting results obtained by the three models are shown in Figure 7:

Figure 7. The results comparison chart.

In Figure 7 above, the blue curve is the original data, the green curve is the fitting data predicted by the BP model, the yellow curve is the fitting data predicted by the LSTM model, and the red curve is the fitting data predicted by the CS-LSTM model. It can be seen that the data fitted by the CS-LSTM model are more consistent with the original data and have higher accuracy.

The MAPE, RMSE, and MAE values of the three models are shown in the table below:

In order to more intuitively see the differences between the above models, we created an error histogram, as shown in Figure 8 below. To keep the dimension constant, we multiplied MAPE by 10^4.

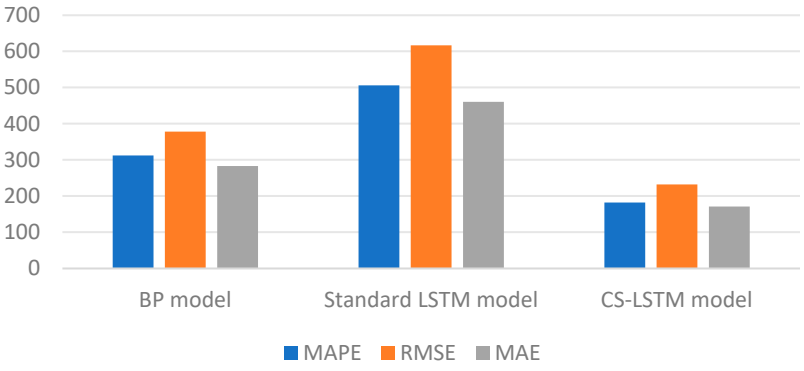

Figure 8. The error histogram of single models.

As can be seen from Table 6 above, the MAPE predicted by the standard LSTM neural network model is 5.059%. Compared to the MAPE of the LSTM neural network, the performance is not as good as that of the BP neural network, but it is not comparable since the BP neural network has 10 hidden layers. In order to improve the performance of the LSTM neural network, the algorithm of optimizing parameters is used to optimize the model in the next chapter.

Table 6. Comparison of evaluation indicators of prediction models.

Evaluation Index	BP Model	Standard LSTM Model	CS-LSTM Model
MAPE	3.12%	5.059%	1.82%
RMSE	378.09	616.48	232.22
MAE	282.76	460.12	170.99

According to the conventional error MAPE of the power system, it can be seen that the MAPE value of the CS-LSTM model is smaller, which also shows the superiority of this model.

4.3. Compared with Optimization Models (the GA-LSTM, the PSO-LSTM)

The genetic algorithm is a targeted search optimization algorithm that integrates the concepts of selection, crossover, and mutation into the process of solving optimization problems by simulating the process of natural evolution. Genetic algorithms are not only adaptive, but also have global optimization capabilities. When applied to the optimization of LSTM neural networks, it can significantly improve the training efficiency of the neural network and reduce the risk of falling into local optimization [26]. The PSO algorithm is a swarm intelligence evolution technique that simulates the predatory behavior of birds. The PSO algorithm locates the optimal particles in the solution space by adjusting the position of the particles, and then searches for the best solution. It is an efficient optimization algorithm that exhibits excellent optimization capabilities.

In this section, we use heuristic search algorithms to solve optimization problems, including Particle Swarm Optimization (PSO), the genetic algorithm (GA), and so on. The GA-LSTM and PSO-LSTM models are established to power load forecasting, and then the optimal model is obtained.

As is the case for the BP model, the data are substituted into the GA-LSTM neural network model for prediction and analysis. The final fitting result is shown in Figure 9:

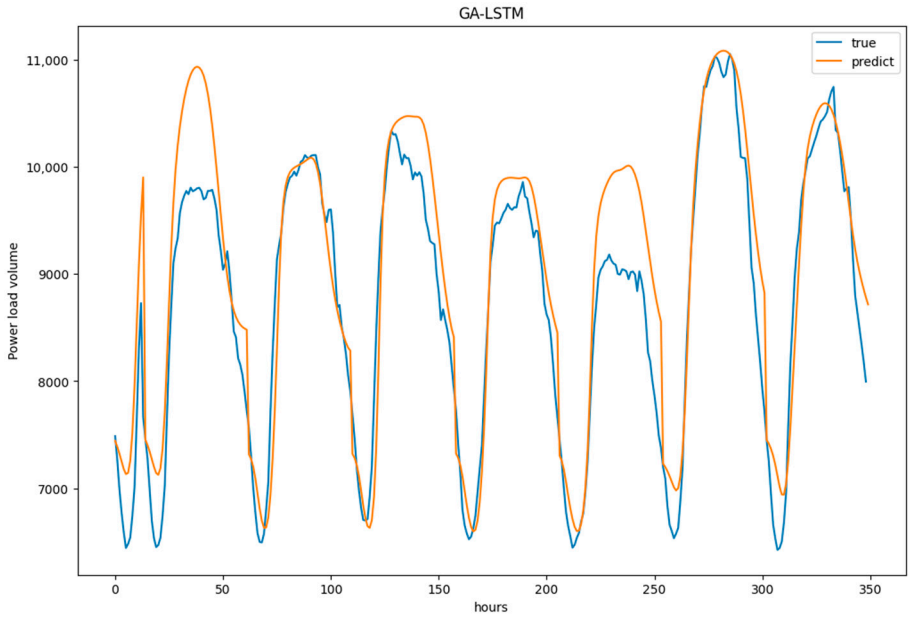

Figure 9. Fitting results of the GA-LSTM neural network model.

The blue curve is the real value curve and the yellow curve is the predicted value curve.

The data are substituted into the PSO-LSTM neural network model for prediction and analysis. TensorFlow in Python is used to predict the model.

The prediction shows that the model fits well. Figure 10 below shows a comparison between the predicted results and the real values:

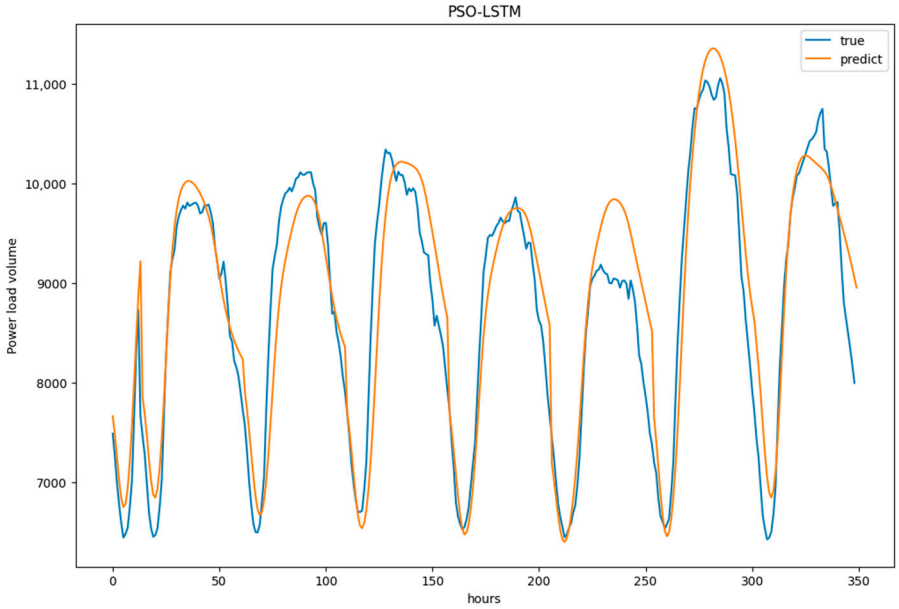

Figure 10. Fitting results of the PSO-LSTM neural network model.

The blue curve is the real value curve, and the yellow curve is the predicted value curve. The fitting results obtained by the three models are shown in Figure 11:

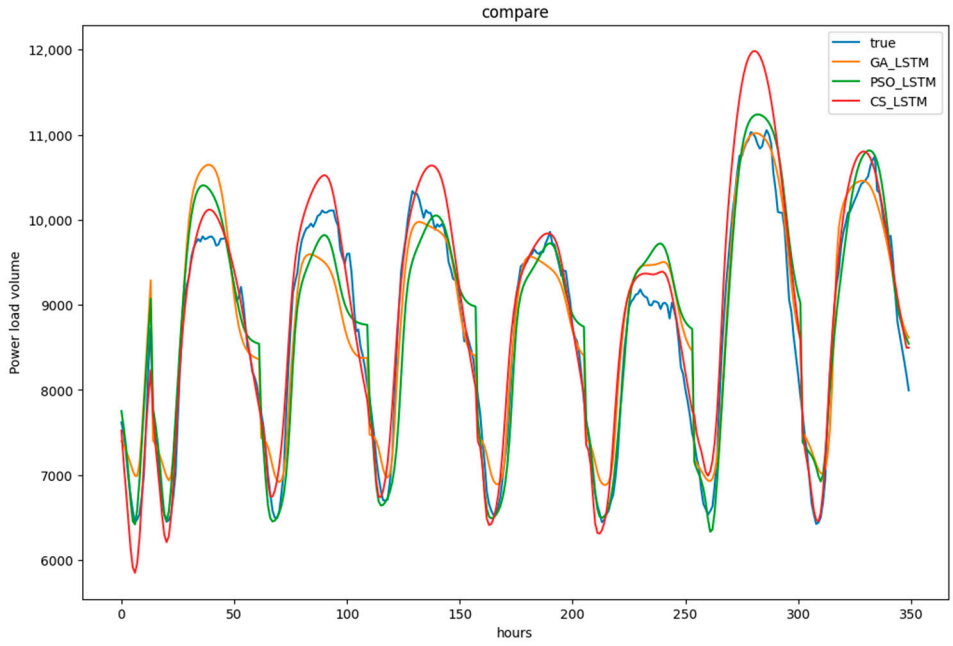

Figure 11. The results comparison chart.

In the Figure 10 above, the blue curve is the original data, the green curve is the fitting data predicted by the PSO-LSTM model, the yellow curve is the fitting data predicted by the GA-LSTM model, and the red curve is the fitting data predicted by the CS-LSTM model. It can be seen that the data fitted by the CS-LSTM model are more consistent with the original data and has higher accuracy.

The MAPE, RMSE and MAE values of the three models are shown in the Table 7 below:

Table 7. Comparison of evaluation indicators of prediction models.

Evaluation Index	The GA-LSTM Model	The PSO-LSTM Model	The CS-LSTM Model
MAPE	0.051	0.053	0.047
RMSE	0.042	0.042	0.037
MAE	0.222	1.095	0.156

In order to more intuitively show the differences between the above models, an error histogram is shown in Figure 12 below:

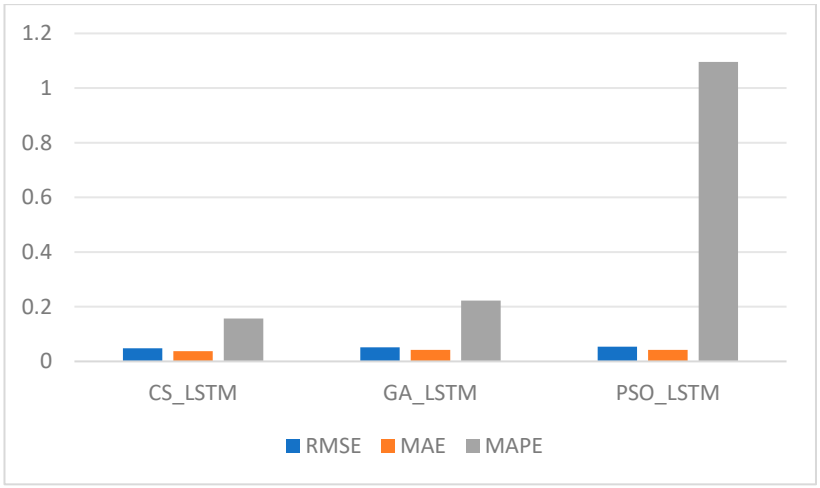

Figure 12. The error histogram of model optimization comparison.

The data in Figure 12 can more intuitively show the prediction accuracy of the three methods. The most accurate prediction is that of the CS-LSTM. It shows that the difference between the predicted value and the true value of the LSTM model optimized by the CS is the smallest. The MAPE value, RMSE value and MSE value of the CS-LSTM are, respectively, 0.047, 0.037 and 0.156. Compared with the GA-LSTM neural network method, the MAPE value, RMSE value and MSE value decreased by 0.004, 0.005, and 0.066, indicating that the predicted value of the CS-LSTM is closer to the real value and could improve the accuracy of load prediction. Compared with the PSO-LSTM method, the MAPE value decreased by 0.005, the RMSE value decreased by 0.005, and the MSE value decreased by 0.939. It shows that the CS algorithm can improve the LSTM model's ability to optimize hyperparameters, and the prediction performance of the model is obviously improved.

5. Conclusions

In this paper, a power load prediction model combining a long- and short-term memory neural network based on the cuckoo search algorithm is used. The LSTM is able to store long-term information, and the cuckoo search algorithm is used to optimize the hyperparameters of the LSTM, which makes the prediction accuracy of the model higher. The CS algorithm can effectively find the global optimal solution to optimize the hyperparameters of the LSTM network, which can avoid the problem that the parameters of the LSTM model are not optimal, caused by human parameter selection, and reduce the influence of human factors on the model accuracy. The higher the accuracy of power load forecasting, the more it can meet the needs of daily life and production and promote the stable operation of the power grid.

Author Contributions: Conceptualization, L.H.; methodology, L.H. and X.W.; software, Y.Y.; validation, Y.Y., X.W. and L.H.; formal analysis, L.H.; investigation, L.H.; resources, L.H. and D.W.; data curation, Y.Y.; writing—original draft preparation, X.W.; writing—review and editing, X.W.; visualization, X.W.; supervision, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: The authors are supported in part by the National Science Foundation of China. Funding number is 11971166.

Data Availability Statement: Data are available upon request to the corresponding author.

Conflicts of Interest: Author Yin Yu was employed by the company Nanjing Laisi Information Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Nti, I.K.; Teimeh, M.; Nyarko-Boateng, O.; Adekoya, A.F. Electricity load forecasting: A systematic review. *J. Electr. Syst. Inf. Technol.* **2020**, *7*, 1–19. [CrossRef]
2. Jiang, P.; Nie, Y. A Hybrid Double Forecasting System of Short Term Power Load Based on Swarm Intelligence and Nonlinear Integration Mechanism. *Appl. Sci.* **2020**, *10*, 1550. [CrossRef]
3. Qiang, G.; Yilong, L. Short-term power load forecasting based on EEMD-SE and RBF optimized by genetic algorithm. *Appl. Electron. Tech.* **2019**, *45*, 51–54.
4. Nobis, F.; Fent, F.; Betz, J. Kernel Point Convoluion LSTM Networks for Radar Point Cloud Segmentation. *Appl. Sci.* **2021**, *11*, 2599. [CrossRef]
5. Liu, J. Application research of artificial neural Network in power load forecasting. *Technol. Econ.* **2019**, *3*, 27–35.
6. Li, D.; Tan, Z.; Lin, S.; Zheng, X.; Wang, T. Short-term Load Forecasting for Microgrid Based on Method of Chaotic Time Series. *Proc. CSU-EPSA* **2015**, *27*, 14–18.
7. Mahmud, M.A. Isolated, Area. Load Forecasting using Linear Regression Analysis: Practical Approach. *Energy Power Eng. (S1949-243X)* **2011**, *3*, 547–550. [CrossRef]
8. Jo, S.H.; Son, S.; Lee, S.; Park, J.W. Kalman-filter-based multilevel analysis to estimate electric load composition. *IEEE Trans. Ind. Electron.* **2011**, *59*, 4263–4271. [CrossRef]
9. Wu, Q.; Gao, J.; Hou, G.S.; Han, B.; Wang, K.Y. Short-term Load Forecasting Support Vector Machine Algorithm Based on Multi-source Heterogeneous Fusion of Load Factors. *Electr. Autom.* **2016**, *40*, 67–72.
10. Zhang, Z.; Zhao, J.; Lu, X. Application of BP Neural Network Based on Genetic Algorithmin Power Load Forecasting. *Comput. Eng.* **2017**, *43*, 277–282.
11. Methaprayoon, K.; Lee, W.J.; Rasmiddatta, W.; Liao, J.R.; Ross, R.J. Multistage Artificial Neural Network Short-Term Load Forecasting Engine with Front-End weather Forecast. *Leee Trans. Ind. Appl.* **2007**, *43*, 1410–1416. [CrossRef]
12. Parlos, A.G.; Ouf, E.; Muthusami, I.; Patton, A.D.; Atiya, A.F. Development of an intelligent long-term electric load forecasting system. In Proceedings of the International Conference on Intelligent System Application to Power Systems, Orlando, FL, USA, 28 January–2 February 1996; pp. 288–292.
13. Rahman, S.; Bhatnagar, R. An expert system based algorithm for short term load forecast. *Power Syst. IEEE Trans.* **1988**, *3*, 392–399. [CrossRef]
14. Hong, W.C. Application of chaotic ant swarm optimization in electric load forecasting. *Energy Policy* **2010**, *38*, 5830–5839. [CrossRef]
15. Kong, X.; Han, S.; Hao, C.; Li, L.; Bai, D. Research on power data forecasting method based on improved grey model. *Electron. Des. Eng.* **2021**, *29*, 139–143.
16. Gao, X.; Li, X.; Zhao, B.; Ji, W.; Jing, X.; He, Y. Short-term electricity load forecasting model based on EMD-GRU with feature selection. *Energies* **2019**, *12*, 1140. [CrossRef]
17. Pang, C.; Zhang, B.; Yu, J. Short-term Power Load Prediction Based on LSTM Recurrent Neural Network. *Electr. Power Eng. Technol.* **2021**, *40*, 175–194.
18. Li, Y.; Huang, Y.; Jiang, G. Short-term Load Prediction of Power System based on PCA-SVM. *J. Electr. Power Syst. Autom.* **2007**, *5*, 66–70.
19. Zheng, H.; Yuan, J.; Chen, L. Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. *Energies* **2017**, *10*, 1168. [CrossRef]
20. Qiao, W. Research on Short-term Power Load Forecasting Based on Improved PSO-BP Neural Network. *Relay* **2007**, *17*, 17–21.
21. Peng, P.; Liu, M. Short-term Load Forecasting Method Based on Prophet-LSTM Combination Model. *J. Electr. Power Syst. Autom.* **2021**, *33*, 15–20.
22. Salgotra, R.; Singh, U.; Saha, S.; Gandomi, A.H. Self adaptive cuckoo search: Analysis and experimentation. *Swarm Evol. Comput.* **2021**, *60*, 100751. [CrossRef]

23. Li, Y.; Jia, Y.; Li, L.; Hao, J.; Zhang, X. Short-term power load forecasting based on stochastic forest algorithm. *Power Syst. Prot. Control* **2020**, *48*, 117–124.
24. Zhang, Y.; Wang, R.; Wu, Q. Dynamic adaptive cuckoo search algorithm. *Control Decis. Mak.* **2014**, *29*, 617–622.
25. Cheng, X.; Huang, Z. Short-term Load Forecasting Model based on Improved PSO optimization of RNN. *Electron. Meas. Technol.* **2019**, *4*, 94–98.
26. Abbas, S.R.; Arif, M. Electric load forecasting using support vector machines optimized by genetic algorithm. In Proceedings of the IEEE International Multitopic Conference, Islamabad, Pakistan, 23–24 December 2006; pp. 395–399.
27. Yang, X.S.; Deb, S. Cuckoo Search via Levy Flights. In Proceedings of the World Congress on Nature and Biologically Inspired Computing, Coimbatore, India, 9–11 December 2009; pp. 210–214.
28. Wei, X.; She, S.; Rong, W.; Liu, A. Health Status Prediction of Lithium Battery Based on BP Neural Network Optimized by Cuckoo Algorithm. *Comput. Meas. Control* **2021**, *29*, 65–75.
29. Tan, Z. Optimization Scheduling of Power System Based on Improved Cuckoo Algorithm. *Electr. Mater.* **2021**, 28–30.
30. Fang, Y. Research on odor source identification method based on Cuckoo algorithm. *Min. Res. Dev.* **2021**, *41*, 183–189.
31. Yu, H.; Zhuang, X.; Liu, H.; Yu, Q.; Luo, S. Application of Power Load Forecasting in Urban Distribution Network Planning Based on 3D Real Scene Platform. *J. Phys. Conf. Ser.* **2020**, *1549*, 52–121.
32. Xie, J.; Gao, Y.; Yu, H. Hybrid particle swarm optimization algorithm based on Gaussian mutation and Levy flight strategy. *J. Baoji Univ. Arts Sci. (Nat. Sci. Ed.)* **2021**, *41*, 5–10.