Journal of Autonomous Intelligence (2020) Volume 3 Issue 2.
doi:10.32629/jai.v3i2.285

Original Article

Prediction of Building Energy Consumption Based on IPSO-CLSTM Neural Network

Liu Xudong*[1]*, Li Shuo*[1]*, Fan Qingwu*[1]

Information Department Beijing University of Technology Beijing, China

ABSTRACT

Accurate prediction of building load is essential for energy saving and environmental protection. Exploring the impact of building characteristics on heating and cooling load can improve energy efficiency from the design stage of the building. In this paper, a prediction model of building heating and cooling loads is proposed, which based on Improved Particle Swarm Optimization (IPSO) algorithm and Convolution Long Short-Term Memory (CLSTM) neural network model. Firstly, the characteristic variables are extracted and evaluated by Spearman’s correlation coefficient method; Then the prediction model based on the CLSTM neural network is constructed to predict building heating and cooling load. The IPSO algorithm is adopted to solve the problem that manual work cannot precisely adjust parameters. In this method, the optimization ability of the PSO algorithm is improved by changing the updating rule of inertia weight and learning factors. Finally, the parameters of the neural network are taken as IPSO optimization object to improve the prediction accuracy. In the experimental stage of this paper, a variety of algorithm models are compared, and the results show that IPSO-CLSTM can get the best results in the prediction of heating and cooling load.

Keywords: *Heating Load; Cooling Load; Protection; Particle Swarm Optimization; Long-Short Term Memory*

ARTICLE INFO
Received: Mar 23, 2021
Accepted: Jun 22, 2021
Available online: Jun 28, 2021

***CORRESPONDING AUTHOR**
Fan Qingwu, Information Department Beijing University of Technology Beijing, China; fqw@bjut.edu.cn;

CITATION
Liu Xudong, Li Shuo, Fan Qingwu. Prediction of Building Energy Consumption Based on IPSO-CLSTM Neural Network. Journal of Autonomous Intelligence 2020; 3(2): 11-22. doi: 10.32629/jai.v3i2.285

1. Introduction

With the rapid development of economic society, people are paying increasing attention to energy waste and environmental protection. The “China Building Energy Consumption Report” released in 2019 pointed out that building energy consumption accounts for 21.11% of the national energy consumption. Several studies have shown that the consumption of Heating Ventilation and Air Conditioning (HAVC) in the total energy consumption of buildings takes the largest proportion[1,2].

The prediction of Heating Load (HL) and Cooling Load (CL) plays an important role in the planning and management of the energy system. Recently, many researchers have focused on predicting model methods. Li *et al.* estimated the hourly cooling load of the building by using the Support Vector Machine (SVM) and three common neural networks which are BP, Radial Basis Function (RBF), and General Regression (GR). The results showed that SVM and GR performed better[3]. Catalina *et al.* used the multiple regression method to estimate the heat demand of buildings. They used the south equivalent surface, the building global heat loss coefficient, and the difference between outdoor and indoor temperatures as input variables to predict the heat demand. The experiment proved that this method had good accuracy[4]. Al-Shammari *et al.* developed a firefly algorithm with grid search to optimize the SVM to predict district heating load, which proved superior to genetic programming (GP), Artificial Neural Network (ANN), and SVM algorithms[5]. Chou *et al*. combined the Support Vector Regression (SVR) with the ANN model to predict the HL and CL of 17 buildings. Compared with other methods, ANN-SVR showed better accuracy[6]. Tsanas *et al.* and Wang *et al.* used Random Forest (RF) to estimate building energy consumption and proved that its performance was superior to Partial Least-Square (PLS)[7], Regression Tree (RT), and SVR[8]. Besides, Wang *et al*. used time factors and weather factors to predict the building cooling load, and the results showed that LSTM neural network had better performance in short-term prediction, while eXtreme Gradient Boosting (XGBoost) had better performance in long-term prediction[9]. Taking the heating and cooling load of a 5-story office building in Tianjin as the study object and outdoor meteorological factors as input, Zhao *et al.* established a wavelet-PLS-SVM prediction model, which is proved to apply to this kind of problem[10]. Guo *et al*. used correlation analysis and the LSSSO method to optimize the feature set. Then they compared the four methods, Media Loss Rate (MLR), BP neural network, SVR, and Extreme Learning Machine (ELM), to respectively predict the energy demand of office buildings. The results proved that ELM has better performance[11]. Yun *et al.* utilized an AutoRegressive exogenous (ARX) model with indexes to predict the hourly heating load of buildings[12]. Wan *et al*. used the quadratic regression method to research the impact of climate change on the HL and CL of office buildings[13]. Some researchers combined meta-heuristic algorithms with neural networks to improve the quality of predictions. Le Thi Le *et al*. combined ANN and meta-heuristic algorithms to predict the heating load of buildings, and the results proved that the meta-heuristic algorithms could optimize ANN parameters to a large extent[14]. Bui *et al*. took advantage of the Genetic Algorithm (GA) and Imperialist Competitive Algorithm (ICA) to optimize the ANN parameters, and the results proved that ICA-ANN had a better prediction effect[15]. Zhou *et al.* used PSO and Artificial Bee Colony Algorithm (ABC) to optimize Multilayer Perceptron (MLP), and the results showed that the optimization performance of PSO was better[16]. Huang *et al.* used the improved Ant Colony Optimization (ACO) to optimize the Wavelet Neural Network (WNN). Compared with the WNN model, the model proposed in this paper has a better effect[17]. Roy S S *et al*. discussed the application of different machine learning technologies in the prediction of residential building heating load and cooling load. They used MARS to evaluate the importance of each parameter in the prediction, and these important parameters were fed to ELM to establish a hybrid model, achieving a relatively good prediction effect[18].

The cooling and heating capacity of air conditioners is predicted based on different parameters, such as building characteristics, climatic factors, and working conditions. Exploring the influence of different influencing factors on cooling and heating load can realize energy saving at different levels. Due to the numerous factors affecting building load, it is difficult to determine its accurate mathematical model, so the current research mainly uses machine learning or neural network methods to model the model.

Weather factors and working parameters are the main factors to be considered in the prediction of building HL and CL. The time series model is used to predict the load demand in the next cycle, to adjust the current equipment working state, and achieve the effect of energy-saving. There are few studies on the influence of building data on energy consumption. In this paper, building data without timestamps are used to regressively predict the HL and CL of buildings. Estimating energy efficiency can guide the design and improvement phase of a building, greatly reducing energy waste.

In this paper, the data set given in the article[15] is used to carry out experiments to predict the HL and CL of the building. Based on this data, some researchers have combined heuristic algorithms with the neural network[7,13,16], but the prediction accuracy still needs to be improved. Therefore, the PSO algorithm and CLSTM neural network model are combined to improve the prediction accuracy of building HL and CL in this paper.

LSTM neural network has shown great performance in dealing with time series problems, but it is rarely used in dealing with regression problems. LSTM memory units can remember long and short-term information. In this paper, we take advantage of this feature to solve the problem of building load regression prediction. At the same time, the convolution layer is used to extract the feature information, and the neural network prediction model of CLSTM is established. Some parameters of the neural network need to be set artificially. With different parameters set, the prediction performance of the trained model is also different. Therefore, it is particularly important to select appropriate model parameters. At present, the selection of network model hyperparameters often depends on the experience of researchers and the results of multiple experiments, which consumes a lot of manpower and computing resources. In this paper, we use the IPSO algorithm to optimize the parameters of the CLSTM neural network, establish the IPSO-CLSTM model.

2. IPSO-LSTM Neural Network Prediction Model

2.1 Overall structure of IPSO-LSTM

The overall structure of the model in this paper is shown in **Figure 1**, which mainly contains three parts: the Convolution layer, the IPSO optimization module, and LSTM neural network prediction model. The Convolution layer module mainly extracts the feature information from the data and then inputs it into the LSTM model. The LSTM neural network model mainly predicts the heating load and the cooling load, the input is $x_1, x_2, \cdots x_n$, and the output is the HL and CL. The main goal of building this model is to achieve accurate prediction of heating load and cooling load. The IPSO algorithm mainly optimizes the parameters of the neural network model, and the output is the number of hidden layer neurons, the number of iterations, and the batch size.

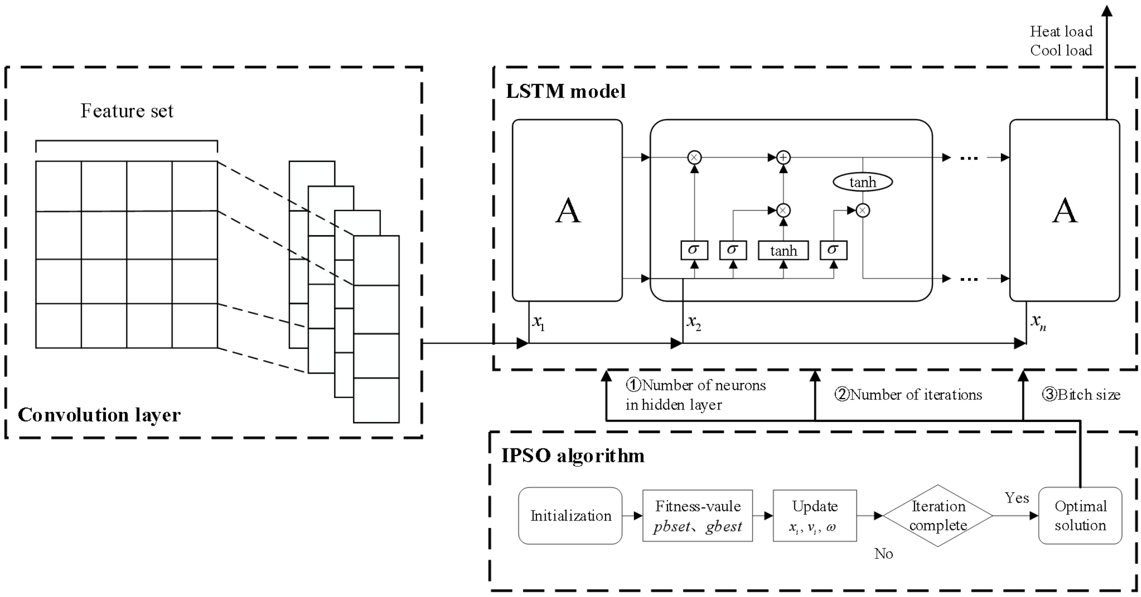


Figure 1. IPSO-CLSTM model structure.

2.2 Convolution layer

The convolution layer is composed of several convolution units, and the parameters of each convolution unit are optimized by the backpropagation algorithm. Each unit is a filter with a width of ω and a height of m, and the same number of m and features. Then the output of the i filter is:

$$h_i = tanh\left(W_i * X + b_i\right) \tag{1}$$

Where the output h_i is the vector, $*$ is the convolution operation. W_i and b_i are the weight matrix and bias, respectively. $tanh$ is the activation function, and it's defined as the following formula:

$$tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2}$$

2.3 LSTM neural network model

LSTM is a special kind of Recurrent Neural Network (RNN). By elaborately designing the gate structure, it avoids the problem of gradient disappearance and explosion caused by traditional RNN, and can effectively learn the long-term dependence relationship. LSTM adds a structure called a memory unit to the neurons in the hidden layer of RNN to remember past information. It has an input gate, forgetting gate, and output gate, which can control the use of historical information.

The neuronal structure of the neural network for LSTM is shown in **Figure 2**. The neuronal structure of the LSTM is composed of three gate structures, which play a role in turn when neurons process information. First, the forgetting gate determines the useless information to be abandoned in the neuron structure, then the input gate determines the useful information to be retained in the neuron structure, and finally, the output gate determines the output result.

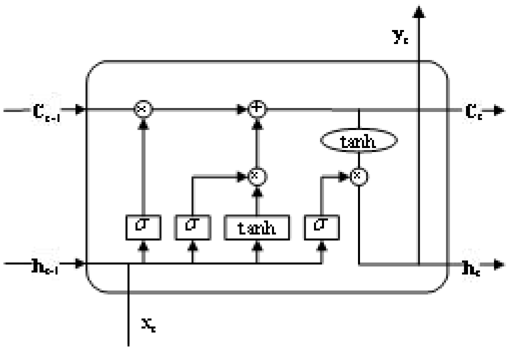


Figure 2. The neural network structure of LSTM.

In the neural network structure of LSTM the forgetting gate, input gate, and output gate can be expressed as the following formula:

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

$$f_t=\sigma\left(W_f\cdot[h_{t-1},x_t]+b_f\right) \tag{4}$$

$$o_t=\sigma\left(W_o\cdot[h_{t-1},x_t]+b_o\right) \tag{5}$$

Then, according to the input x_t at the current moment and the state value h_{t-1} at the previous moment, the candidate state value of the current neuron is calculated:

$$\hat{C}_t=tanh\left(W_c\cdot[h_{t-1},x_t]+b_c\right) \tag{6}$$

The proportion of the state value C_{t-1} at the previous moment and the candidate state value at the current moment in the new state value $\hat{C}_t$ is determined by forgetting the gate f_t and input gate i_t.

$$C_t=f_t\cdot C_{t-1}+i_t\cdot\hat{c}_t \tag{7}$$

Finally, calculate the output value y_t at the current moment:

$$y_t=h_t=o_t\cdot tanh(C_t) \tag{8}$$

In this paper, the model in the experiment is built under the Python Keras framework. The loss function uses MSE, and the training process is optimized by the Adam algorithm.

2.4 PSO algorithm and improvement

PSO algorithm is a swarm intelligence optimization algorithm that simulates the social behavior of animals such as birds and fish. The particle has only two properties: velocity and position. Each particle represents a possible solution to the problem, and its characteristic information is described by the position, velocity, and fitness value. The fitness value is calculated by the fitness function.

The PSO is initialized as a group of random particles and then finds the optimal solution through continuous updating and iteration. In each iteration, the particle updates itself by tracking two “extreme values”($pbest$, $gbest$). After finding these two optimal values, the particle updates its velocity and position using the following formula.

$$v_i(t+1)=\omega\times v_i(t)+c_1\times rand()\times\left(pbest_i-x_i(t)\right)+c_2\times rand()\times\left(gbest_i-x_i(t)\right) \tag{9}$$

$$x_i(t+1)=x_i(t)+v_i(t+1) \tag{10}$$

Where v_i is the velocity of the particle; $rand()$ is a random number between (0,1); c_1 and c_2 are the learning factors; ω is the inertia weight. In this paper, the updating rules of inertial weight and learning factors are changed to improve particle swarm optimization.

2.4.1 Improvement of inertial weight

The inertia weight has a great influence on the convergence of the PSO algorithm. When the value is large, the global capability is strong, and the local optimization capability is weak. When the value is small, the global optimization ability is weak, and the local optimization ability is strong. In this paper, ω is updated using a linear decrement strategy, which enables ω to change as the number of iteration. Typical linear decline strategies are as follows:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} \times t \tag{11}$$

Where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum values of ω respectively; t and $t_{\max}$ are the current iteration number and maximum iteration number respectively.

This method enables PSO to follow the iteration number to control the global search ability and local search ability, which improves the algorithm performance. However, if the global optimum value cannot be searched at the beginning of the iteration, the local searching ability will be enhanced with the decrease of the value, and the local extremum will be easily trapped.

Because of the large range of neural network parameters, a typical linear decreasing strategy is easy to fall into the local extremum. To overcome this limitation, a linear differential diminishing strategy is adopted in this paper. The calculation formula is as follows:

$$\frac{d\omega}{dt} = -\frac{2\left(\omega_{\max} - \omega_{\min}\right)}{t_{\max}^2} \times t \tag{12}$$

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}^2} \times t^2 \tag{13}$$

In the early stage of the algorithm, the decreasing trend of ω is slow, and the global search ability is very strong, which is conducive to finding a suitable solution in a large range. In the later stage of the algorithm, the decreasing trend of ω is accelerated. Once a suitable solution is found in the early stage, the convergence speed of the algorithm can be accelerated.

2.4.2 Improvement of learning factors

It can be seen from formula (1) that c_1 reflects the trend degree of particles approaching their personal historical best position and c_2 reflects the trend degree of particles approaching their global historical best position. Usually, $c_1 = c_2 = 2$, but we hope that the global search capability is strong in the early stage of iteration and the local search capability is strong in the later stage. Therefore, c_1 should decrease with the progress of the algorithm and c_2 should increase with the progress of the algorithm. The specific expression is as follows:

$$c_1\left(t\right) = c_{\max} - \frac{t\left(c_{\max} - c_{\min}\right)}{t_{\max}} \tag{14}$$

$$c_2\left(t\right) = c_{\min} + \frac{t\left(c_{\max} - c_{\min}\right)}{t_{\max}} \tag{15}$$

In the above formulas, $c_{\max}$ and $c_{\min}$ are the maximum and minimum values of c respectively; t and $t_{\max}$ are the current iteration times and the maximum iteration times respectively.

The improvement in the above two directions can make the algorithm search the solution space in a wide range in the initial phase of iteration and gather together the optimal solution quickly in the later phase to improve the performance of all aspects of the algorithm.

2.5 IPSO-CLSTM model

To accurately predict the building load, the Convolution and LSTM neural network are combined in this paper to construct the prediction model CLSTM for the building heating load and cooling load. The value of some hyperparameters in the neural networks controls the model network structure. To make the model network structure match the data characteristics, we combine the

IPSO algorithm with the neural network model to build the IPSO-CLSTM prediction model.

The model first takes the number of hidden layer neurons, iteration times, and batch size as the optimization objects of the IPSO algorithm, besides to randomly initializes the particle position information according to the value range of each parameter. Then the CLSTM neural network model is established by using the parameters corresponding to the particle position, and the model is trained and predicted by using the data. The mean square error of the model is taken as the particle fitness value. The fitness function is defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{16}$$

In the formula: n is the number of test data; y_i is the true value of test data i ; $\hat{y}_i$ is the predicted value of test data i .

According to the fitness value of each particle, the individual extremum and the global extremum are obtained. Equations (9) and (10) are used to update the particle velocity and position respectively to achieve the optimization goal of minimum MSE. Finally, the neural network model is constructed with the optimal particle position information to complete the establishment of IPSO-CLSTM.

The flow of the model algorithm is as follows:

Step 1: Divide experimental data into training data and test data, and conduct standardized processing;

Step 2: The number of hidden layer neurons, iteration times, and batch size in the CLSTM neural network model are taken as optimization objects, and the IPSO algorithm is initialized;

Step 3: Determine Equation (16) as the fitness function. The CLSTM neural network model is constructed by initializing the parameters corresponding to particle information, and the fitness value is obtained through training and prediction;

Step 4: Calculate the fitness value of each particle and compare it. Then record individual extremum and global extremum;

Step 5: According to Equation (13)-(15), the inertia weight is updated. Then, Equations (9) and (10) are applied to constantly update the velocity and position of particles;

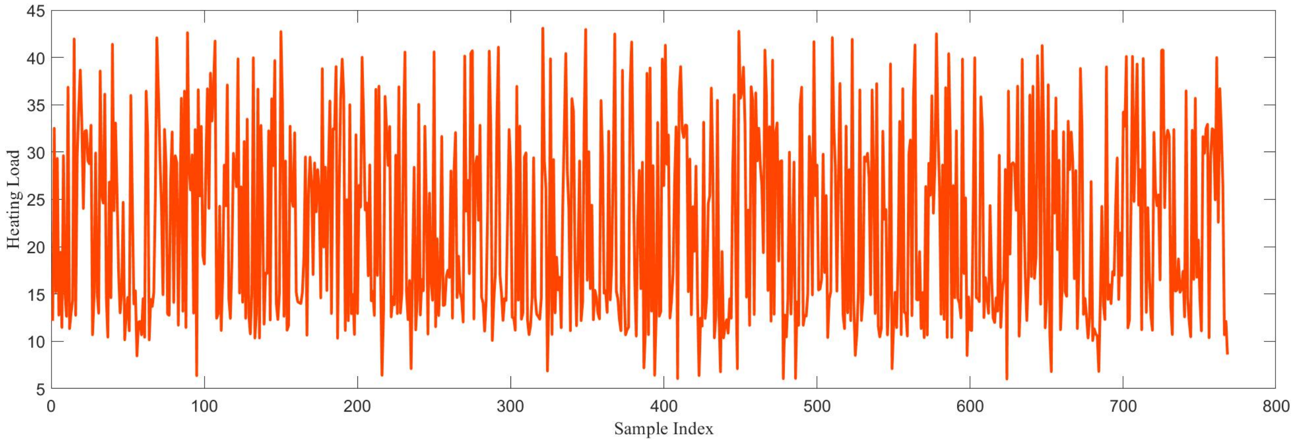
Step 6: After meeting the maximum number of iterations of the IPSO algorithm, the CLSTM neural network model is constructed by taking the optimal value of the hyperparameters for training and prediction.

3. Experiments and Discussion

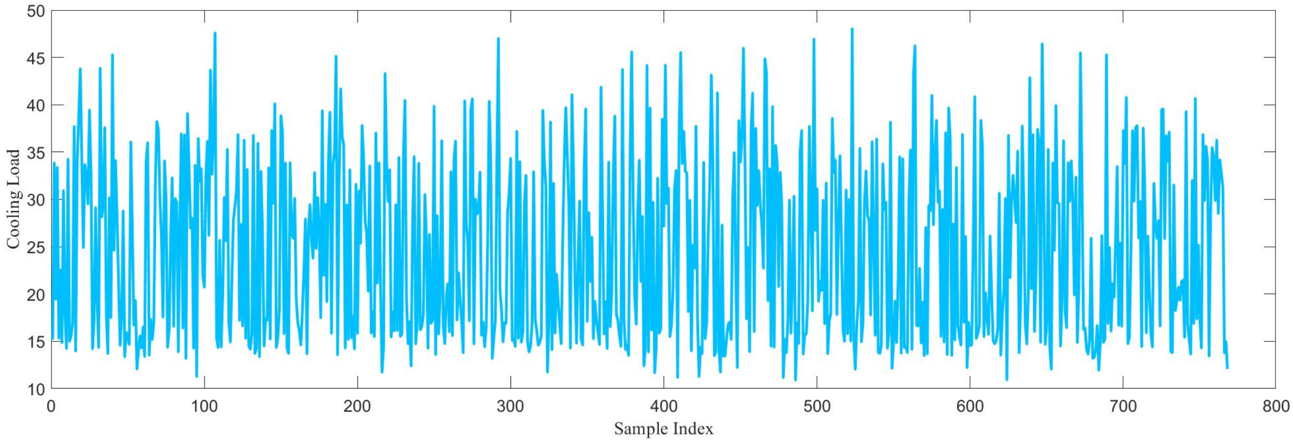
3.1 Experimental Data

The experiment applied the data set selected from the UCI machine learning storage library. The data set is also established in the paper[7]. The data was generated by simulating 12 different building shapes for energy analysis in Ecotect software. There are 8 input variables in the data set, Relative Compactness (RC), Surface Area (SA), Wall Area (WA), Roof Area (RA), Overall Height (OH), Orientation (OR), Glazing Area (GA), and Glazing Area Distribution (GAD). The response variables are HL and CL in the data set. Although there is no guarantee that the simulated data will perfectly reflect the actual data in the actual project, the simulated data results can provide a good indication of the possible percentage changes and potential trends in the actual data, thus enabling the building energy comparison.

Since LSTM can learn sequence features, we randomly scrambled experimental data to eliminate the influence of sequence. Heating and cooling loads after disruption are shown in **Figure 3**.



(a) Heating Load



(b) Cooling Load

Figure 3. Disorganized data presentation.

Because it is the data simulated by the software, all variables have no missing values or outliers. Therefore, there is no need to do too much data preprocessing work. In this paper, to eliminate the influence caused by different dimensions, and improve the speed and accuracy of the solution, we standardized the data.

$$y_i = \frac{x_i - \bar{x}}{s} \tag{17}$$

Where y_i is the normalized value; x_i is the original value; $\bar{x}$ and s represent the mean and variance of the original data, respectively. After standardization, the data mean is 0, the variance is 1, and dimensionless.

The error evaluation indexes of the experiment adopt Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-square, namely:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{18}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{19}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{20}$$

$$R\text{-}square = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}_i\right)^2} \tag{21}$$

Where y_i is the true value; $\hat{y}_i$ is the predicted value.

In this paper, the dataset is split into training and test data with a probability of 0.8 and 0.2 respectively. Then the prediction results of IPSO-CLSTM, IPSO-LSTM, MLR, SVR, and ELM models are compared.

3.2 Selection of feature variables

Before the formal experiment, the data needs to be reduced in dimensionality. Since the Principal Components Analysis (PCA) algorithm cannot handle nonlinear problems and will be accompanied by information loss, traditional methods are used in this paper to reduce the dimensionality of the data.

In this paper, the correlation coefficient and significance degree between different variables and loads are calculated to optimize the feature set of the prediction model. This method can not only reduce the dimension of data but also reduce the workload in the subsequent data collection process. Since the data is non-normally distributed, the Spearman rank correlation coefficient is used to measure the correlation between each input variable and the two output variables. The p-value is used to assess whether the relationship is statistically significant and to check for significance at the 0.01 level. The results are shown in the following table:

Table 1. Correlation of various variables and load

Input variable	**HL(Y1)**		**CL(Y2)**	
	correlation coefficient	***p-value***	***correlation coefficient***	***p-value***
RC(X1)	0.6221	<0.01	0.6510	<0.01
SA(X2)	-0.6221	<0.01	-0.6510	<0.01
WA(X3)	0.4715	<0.01	0.4160	<0.01
RA(X4)	-0.8040	<0.01	-0.8032	<0.01
OH(X5)	0.8613	<0.01	0.8649	<0.01
OR(X6)	-0.0042	>0.05	0.0176	>0.05
GA(X7)	0.3229	<0.01	0.2889	<0.01
GAD(X8)	0.0683	>0.05	0.0465	>0.05

According to the results in the above table, the values of orientation and glass area distribution and HL and CL are both greater than 0.05, so they show an insignificant relationship. The glass area is weakly correlated for HL and CL because the correlation coefficients between the glass area and two output variables are both about 0.3. According to the experimental results, 3 feature sets are constructed by using the complete set, removing OR and GAD, removing OR, GAD and GA, as shown in the table below:

Table 2. Feature sets information

Feature sets for the load prediction models			
Name	***Variables***	***Number***	***Output***
FS1	RC, SA, WA, RA, OH, OR, GA, GAD	8	HL, CL
FS2	RC, SA, WA, RA, OH, GA	6	HL, CL
FS3	RC, SA, WA, RA, OH	5	HL, CL

Then, MLR, SVR, and ELM are used in this paper, and RMSE and MAPE are used to evaluate the influence of each feature set on the prediction accuracy of the model. The results are shown in the table below:

Table 3. Model performance comparison under different feature sets

Model		FS1		FS2		FS3	
		RMSE	***MAPE***	***RMSE***	***MAPE***	***RMSE***	***MAPE***
MLR	HL	3.146	10.85%	2.889	10.61%	4.376	16.44%
	CL	3.538	13.33%	3.121	9.05%	3.899	11.95%
SVR	HL	2.163	7.22%	1.724	6.12%	3.876	14.79%
	CL	2.433	6.73%	2.292	6.12%	3.376	10.45%
ELM	HL	0.989	3.31%	0.503	1.58%	3.538	13.33%
	CL	3.097	7.00%	1.749	3.69%	3.124	9.38%

By horizontal comparison of the performance of different models under different feature sets, it is not difficult to find that compared with FS1, FS2 has better performance after removing OR and GAD. Therefore, insignificant variables have no positive influence on the model. The performance of FS3 is worse after removing OR, GAD, and GA variables. By comparing FS2 and FS3, it can be seen that the correlation is weak, but it still played a role in model training and prediction.

3.2 Selection of feature variables

According to the experimental results in the previous section, the subsequent experiments mainly focus on the feature set FS2 in this paper. Because there is a big difference between HL and CL, we separate the two for research in this paper. There are altogether 768 data in the data set, among which 613 data are taken as the training set and 155 data are taken as the test data. The number of particle swarm is set as 30, the maximum number of iterations is 50. The learning factor is 2.5 at the maximum and 0.5 at the minimum. The inertia weight is 0.9 at the maximum and 0.4 at the minimum.

For heating load prediction, the number of hidden layer neurons, the number of iterations, the batch size set value range respectively [32, 128], [50, 100], and [20, 100], the final optimization parameters are [64, 512, 80]. The experimental results are shown in **Figure 2**, the IPSO-CLSTM model, and other models contrast results as shown in **Table 4**. For cooling load predicting, the parameter value range respectively [64, 256], [150, 800] and [80, 600], and the final optimization parameters are [128, 512, 512]. The experimental results are shown in **Figure 3**, the IPSO-CLSTM model compared with other models of the results as shown in **Table 5**.

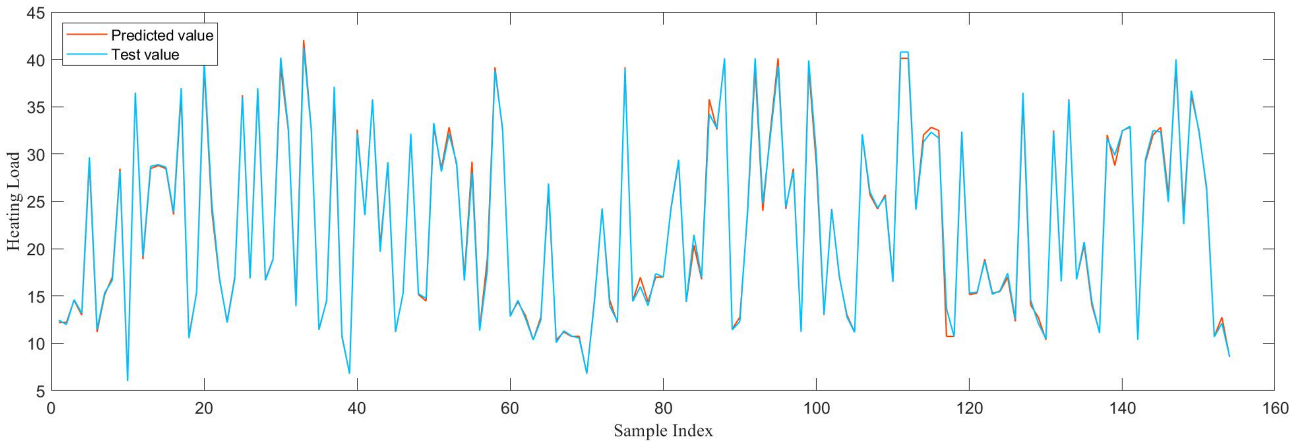


Figure 4. IPSO-CLSTM heating load prediction results.

Table 4. Comparison of evaluation indexes of various models for heating load forecast

Model	RMSE	MAE	MAPE	R-square
MLR	3.184	2.253	10.87%	0.896
SVR	1.724	1.207	6.12%	0.970
ELM	1.068	0.659	3.47%	0.988
LSTM	0.595	0.453	2.43%	0.996
IPSO-LSTM	0.518	0.375	1.89%	0.997
IPSO-CLSTM	**0.506**	**0.343**	**1.60%**	**0.997**

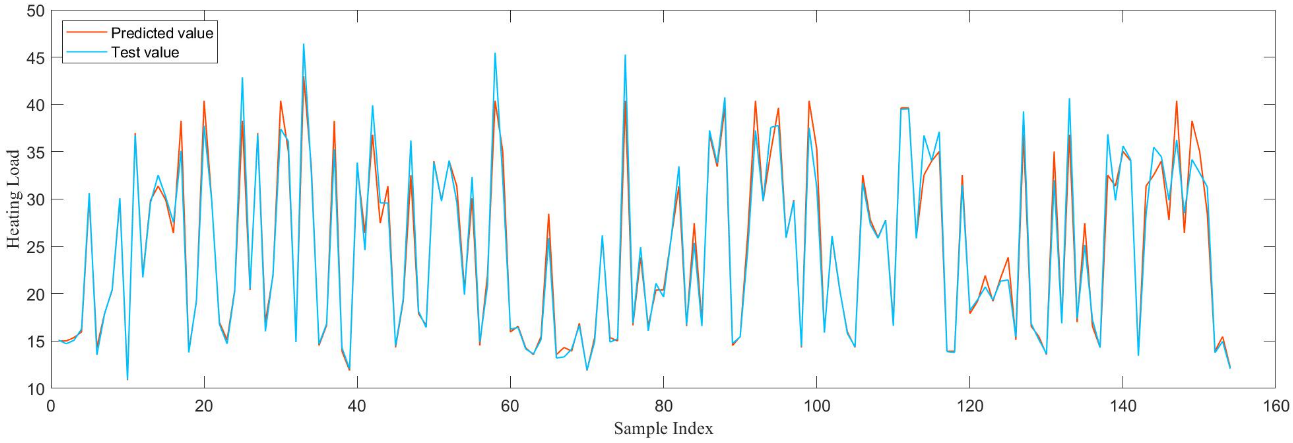


Figure 5. IPSO-LSTM cooling load prediction results.

Table 5. Comparison of evaluation indexes of various models for cooling load forecast

Model	RMSE	MAE	MAPE	R-square
MLR	3.121	2.277	9.05%	0.888
SVR	2.292	1.546	6.12%	0.940
ELM	1.792	1.211	4.21%	0.963
LSTM	1.789	1.170	4.08%	0.963
IPSO-LSTM	1.757	1.166	3.98%	0.965
IPSO-CLSTM	**1.629**	**1.020**	**3.46%**	**0.970**

From the above figures and tables, it can be seen that for building heating load and cooling load prediction, the prediction performance of the IPSO-CLSTM model proposed in this paper is better than other models, and the R-square index is closer to 1 than other models. For HL prediction, compared with ELM, IPSO-CLSTM reduced RMSE by 14.9%, MAE by 24.3%, and MAPE by 34.2%, and model accuracy is significantly improved. For CL prediction, compared with ELM, IPSO-CLSTM reduced RMSE by 8.9%, MAE by 14.0%, MAPE by 15.2%, and model accuracy is improved. Each index of the IPSO-CLSTM model is better than that of the LSTM neural network model, but the difference between them is not obvious, mainly because these two prediction models have the same unit structure. The most prominent advantage of the IPSO-CLSTM model is that manual parameter adjustment is not required during the construction process, and the prediction results are better than the common LSTM neural network model.

The experimental results of this paper are compared with the experimental results of the best method in the paper [14][15][16]. The results are shown in the table below:

Table 6. Performance comparison of methods between papers

Paper	Method	HL			CL		
		RMSE	*MAE*	R^2	*RMSE*	*MAE*	R^2
[13]	GA-ANN	1.625	0.798	0.98	-	-	-
[14]	ICA-ANN	2.782	2.009	0.912	2.799	2.105	0.929
[15]	PSO-MLP	2.569	1.863	0.937	3.122	2.136	0.900
This paper	IPSO-CLSTM	0.506	0.343	0.997	1.629	1.020	0.970

It can be seen from the table that the IPSO-CLSTM algorithm proposed in this paper is obviously superior to the methods in the other three papers, showing better advantages in the prediction of building HL and CL.

4. Conclusion

Aiming at the problem of HL and CL regression prediction based on building data, the IPSO-CLSTM model is proposed in this paper. The model improves the optimization ability of the PSO algorithm by improving the updating rule of inertia weight. This method mainly optimizes the CLSTM network structure through the IPSO algorithm to reduce the influence of human factors. In this paper, data sets from the UCI machine learning repository are selected for experiments, and the results show that IPSO-CLSTM has higher prediction accuracy than other algorithms. Compared with the ELM model, the RMSE of IPSO-CLSTM decreased by 14.9% for heating load and 8.9% for the cooling load.

It can be seen from the experimental results that the prediction accuracy of various models for the cooling load is not good. The following work is mainly carried out from two aspects: one is to replace a more suitable model to predict cooling load; the other is to look for building factor variables more closely related to the cooling load.

In this paper, it mainly makes the following innovations and contributions:

(1) Use only building data to explore its influence on cooling and heating loads.

(2) The LSTM is applied to the multiple regression prediction of non-temporal relationship to explore its fitting effect on the expected value of multidimensional data.

(3) By improving the inertial weight and learning factor of PSO, the global optimization and local fast convergence of the algorithm are enhanced.

(4) Compared with other methods, the proposed IPSO-CLSTM method has a great improvement in accuracy; Compared with LSTM, the proposed method reduces the intervention of human factors. Although it is higher in time complexity and computational cost than LSTM, this cost is acceptable for the improvement of accuracy.

References

1. Rim D, Schiavon S, Nazaroff WW. Energy and cost associated with ventilating office buildings in a tropical climate. PLoS One 2015; 10(3).
2. Yao R, Li B, Steemers K. Energy policy and standard for built environment in China. Renewable Energy 2005; 30(13): 1973-1988.
3. Li Q, Meng Q, Cai J, *et al.* Predicting hourly cooling load in the building: A comparison of support vector machine and different artificial neural networks. Energy Conversion and Management 2009; 50(1): 90-96.
4. Catalina T, Iordache V, Caracaleanu B. Multiple regression model for fast prediction of the heating energy demand. Energy and Buildings 2013; 57: 302-312.
5. Al-Shammari, *et al.* Prediction of heat load in district heating systems by Support Vector Machine with Firefly searching algorithm. Energy 2013; 95: 266-273.
6. Chou JS, Bui DK. Modeling heating and cooling loads by artificial intelligence for energy-efficient building design. Energy and Buildings 2014; 82: 437-446.
7. Tsanas A, Xifara A. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. Energy and Buildings 2012; 49: 560-567.
8. Wang Z, Wang Y, Zeng R, *et al.* Random forest based hourly building energy prediction. Energy and Buildings 2018; 171: 11-25.
9. Wang Z, Hong T, Piette MA. Building thermal load prediction through shallow machine learning and deep learning. Applied Energy 2020; 263: 114683.
10. Zhao J, Liu X. A hybrid method of dynamic cooling and heating load forecasting for

office buildings based on artificial intelligence and regression analysis. Energy and Buildings 2018; 174: 293-308.
11. Guo Y, Wang J, Chen H, *et al.* Machine learning-based thermal response time ahead energy demand prediction for building heating systems. Applied Energy 2018; 221: 16-27.
12. Yun K, Luck R, Mago PJ, *et al.* Building hourly thermal load prediction using an indexed ARX model. Energy and Buildings 2012; 54: 225-233.
13. Wan KK, Li DH, Liu D, *et al.* Future trends of building heating and cooling loads and energy consumption in different climates. Building and Environment 2011; 46(1): 223-234.
14. Le LT, Nguyen H, Dou J, *et al.* A comparative study of ‘PSO-ANN, GA-ANN, ICA-ANN, and ABC-ANN in Estimating the Heating Load of Buildings’ energy efficiency for smart city planning. Applied Sciences 2019; 9(13): 26-30.
15. Bui DT, Moayedi H, Anastasios D, *et al.* Predicting heating and cooling loads in energy-efficient buildings using two hybrid intelligent models. Applied Sciences 2019; 9(17): 35-43.
16. Zhou G, Moayedi H, Bahiraei M, *et al.* Employing artificial bee colony and particle swarm techniques for optimizing a neural network in prediction of heating and cooling loads of residential buildings. Journal of Cleaner Production 2020; 254:120082.
17. Huang Yuting, Li Chao. Accurate heating, ventilation and air conditioning system load prediction for residential buildings using improved ant colony optimization and wavelet neural network. Journal of Building Engineering 2019; 35.
18. Sanjiban Sekhar Roy, Roy R , Balas VE. Estimating heating load in buildings using multivariate adaptive regression splines, extreme learning machine, a hybrid model of MARS and ELM. Renewable and Sustainable Energy Reviews 2018; 82: 4256-4268.