J. Meas. Sci. Instrum., 2024, 15(2): 204-215. http://jmsi.publish.founderss.cn
DOI: 10.62756/jmsi.1674-8042.2024021 jmsi@nuc.edu.cn

Short-term PV power forecasting based on combined SOM-FCM and KELM method

LIU Qibo, LI Jun*

School of Automation and Electrical Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

***Corresponding author:** LI Jun (lijun691201@mail.lzjtu.cn)

Received: January 15, 2023 **Revised:** March 10, 2023 **Accepted:** April 21, 2023

Abstract: A hybrid forecasting model was proposed to improve the accuracy of short-term photovoltaic (PV) power generation forecasting, which combined the clustering of trained self-organizing map (SOM) network and optimized kernel extreme learning machine (KELM) method. First, a pure SOM was employed to complete the initial partitions of the training data set. Then clustering was executed on the trained SOM network by fuzzy C-means (FCM). Meanwhile, the davies-bouldin index (DBI) was hired to determine the optimal size of clusters. Finally, in each data partition, the regional KELM model was built with the KELM optimized by differential evolution, or the regional linear regression (MR) model was built with the multiple MR using the least square method to complete the coefficient evaluation. In addition, varying local multiple regression model was also proposed based on SOM. The proposed model based on SOM-FCM and KELM was employed to one-hour-ahead PV power forecasting instances of three different solar power plants provided by the GEFCom2014. Compared with other control models, the mean absolute error (MAE) of plant 1 was reduced by 61.41%, that of plant 2 by 60.19%, and that of plant 3 by 58.92%. The root means square errors (RMSE) of plant 1 was reduced by 52.06%, that of plant 2 by 54.56%, and that of plant 3 by 51.43% on average. The forecasting accuracy was significantly improved with the proposed model.

Key words: photovoltaic power generation; power forecasting; self-organizing map; regional modeling methods; optimized kernel extreme learning machine (KELM) method

0 Introduction

In terms of replacing traditional energy sources, solar energy has become one of the most popular solutions with the advantages of abundant resources, pollution-free, free use, and no need for transportation[1,2]. However, due to the variability of sunshine and weather, photovoltaic (PV) power generation has intermittent, volatility and randomness, which is difficult to control and measure, further endangering the safety and stability of PV grid-connected. Accurate forecasting is generally considered to be an effective solution to this problem[3].

Ordinarily, persistence models and statistical models are commonly used technologies in PV power forecasting. With the rapid development of machine learning and deep learning, artificial intelligence (AI) technology is widely used, and neural networks (NNs) are regularly used AI technologies in short-term PV power forecasting system. The state-of-art forecasts based on NNs include various hybrid forecasting techniques. For instance, a hybrid deep learning model that combines packet decomposition with long short-term memory network[4], a novel heuristic competitive group optimization algorithm which is proposed to train the linear or nonlinear parameters of the radial basis function (RBF) NN[5], and an attention-based long-term and short-term temporal NN[6], etc. It is worth mentioned that based on the advantages of simple structure and super learning ability, single-hidden layer feedforward neural networks (SLFNs) are widely used for wind and PV power forecasting. Compared with the ELM method, KELM is easy to implement. Since the kernel function is used to replace the unknown nonlinear feature mapping of the hidden layer, under the premise of ensuring the learning speed of the ELM, the generalization ability of the KELM method is superior. For forecasting tasks, the KELM method has been widely adopted. Different KELM methods have been used to forecast short-term wind and PV power generations[7-9].

In addition, self-organizing map (SOM) is a special NN. To determine the output of the SOM, neurons in the output layer are regularly arranged to compete with each other to

find the best matching unit. SOM has been widely available in fields of pattern recognition, biological modeling data mining, etc.[10]. Notably, SOM and its extended models are increasingly employed to time series forecasting, including local autoregressive model[11], the vector-quantized temporal associative memory (VQTAM) technology[12], and a novel extension of double vector quantization[13].

Although SOM is widely employed in time series prediction, it has great potential as a tool for regression analysis. On the other hand, in most cases of PV power forecast, a single global model may not be able to capture the dynamics of system with high curvatures, and cannot smooth the details. In addition, in the case with a large volume of data, the performance of a single global model will be greatly compromised. In order to avoid the shortcomings of a single global model, the potential of SOM as a regression analysis tool is explored, combined with the success of SOM in the course of forecasting tasks.

Different local modelling methods were first proposed in this paper, which combined SOM with multiple linear regression (MR) or KELM based on the unified local modeling framework to improve the accuracy of single MR or KELM. However, two problems exist in local modeling methods. First, the optimal number of local models does not have any empirical knowledge and cannot be determined in advance. Second, superabundant local models are prone to overfitting, resulting in poor generalization ability.

A novel forecasting model was proposed that combined the two-level algorithm (the clustering of trained SOM network) [14,15] and optimized KELM by differential evolutionary algorithm[16]. Specifically, a pure SOM was first employed to divide the input space into multiple prototypes[17], and the prototypes were then clustered using fuzzy C-means (FCM) [18]. Meanwhile, the davies-bouldin index (DBI) [19] was used to evaluate the clustering under different sizes and determine the optimal one. A given cluster defined a region in the input space formed by merging the prototypes belonging to that cluster. In the final step, for each partition, only data mapped to this region was combined with the optimized KELM to construct RKELM model. The proposed method was applied to the forecasts of one-hour-ahead PV power generation, and the data was derived from GEFCom2014[20].

1 SOM-based local regression model

1.1 SOM network

The SOM generates a low-dimensional discrete mapping space $\boldsymbol{Y}$ by learning the data from a high-dimensional continuous input space $\boldsymbol{X}$[21]. The typical network structure of SOM is shown in Fig.1.

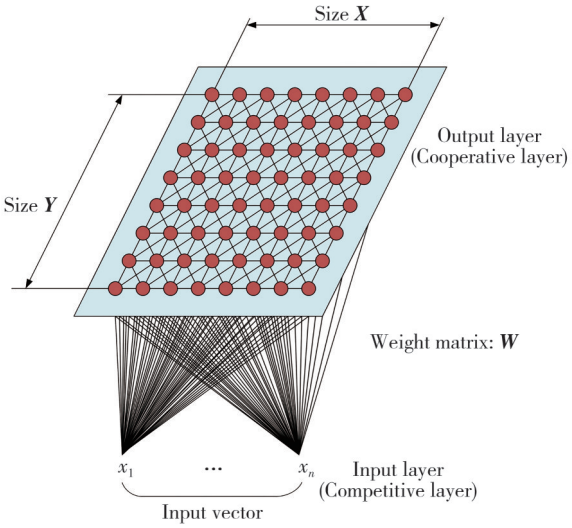


Fig. 1 Typical structure of SOM network

The SOM algorithm is divided into two stages: the competitive stage and the cooperative stage. In the first case, the "winner" neuron is selected. Let $\boldsymbol{W}=\{\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_P\}$, $\boldsymbol{w}_i\in\mathbf{R}^d$ be the weight matrixes of P neurons which are arranged in fixed topological forms, e.g., a hexagonal two-dimensional array. For each input vector $\boldsymbol{x}=[x_1,x_2,\cdots,x_d]\in\mathbf{R}^d\subset\boldsymbol{X}$, a winning neuron $i^*\in\boldsymbol{Y}$ is selected with the smallest Euclidean distance, that is

$$i^*=\arg\min_i\|\boldsymbol{w}_i-\boldsymbol{x}\|,\ i=1,2,\cdots,P. \tag{1}$$

In the second stage, the weights of the winning neuron and its neighboring neurons are updated. The update rules are

$$\boldsymbol{w}_i(t+1)=\boldsymbol{w}_i(t)+\alpha(t)\Lambda\big(i,i^*,\sigma(t)\big)\big[\boldsymbol{x}(t)-\boldsymbol{w}_i(t)\big], \tag{2}$$

where $0<\alpha(t)<1$ is the learning rate; $\Lambda\big(i,i^*,\sigma(t)\big)$ represents the neighborhood function. The update rule of the neighborhood function is performed by the Gaussian function.

$$\Lambda\big(i,i^*,\sigma(t)\big)=\exp\left(-\frac{\|\boldsymbol{r}_i(t)-\boldsymbol{r}_i(t)\|^2}{2\sigma^2(t)}\right), \tag{3}$$

where $\boldsymbol{r}_i(t)$ and $\boldsymbol{r}_{i^*}(t)$ are the coordinates of the neurons i and i^* in the output array, respectively. $\sigma(t)>0$ is the radius of the neighborhood function at step t. The values of $\alpha(t)$ and $\sigma(t)$ should decrease with time to guarantee the convergence of weight vector. In this paper, the two variables are updated using the usual exponential decay methods.

$$\alpha(t)=\alpha_0\left(\frac{\alpha_T}{\alpha_0}\right)^{(t/T)},\ \sigma(t)=\sigma_0\left(\frac{\sigma_T}{\sigma_0}\right)^{(t/T)}, \tag{4}$$

where α_0 is the initial learning rate and α_T is the final learning rate, similar, σ_0 and σ_T are the initial neighborhood function radius and final neighborhood function radius, respectively.

1.2 SOM-MR model

To achieve SOM-MR, the SOM network is first trained to match all inputs to the "winner" neuron and complete the data segmentation. Let $\boldsymbol{S}_j=\left\{\boldsymbol{x}_1^j,\boldsymbol{x}_2^j,\cdots,\boldsymbol{x}_{n_j}^j\right\}$ be a subset of $\boldsymbol{S}=\{\boldsymbol{S}_1,\boldsymbol{S}_2,\cdots,\boldsymbol{S}_P\}$, $\boldsymbol{S}_j$ represent a set of n_j data vectors mapped to neuron j, and $\boldsymbol{x}_i^j=\left[x_{i,1}^j,x_{i,2}^j,\cdots,x_{i,d}^j\right]$ represent the ith training data vector in $\boldsymbol{S}_j$.

The coefficient vector $\boldsymbol{\eta}_j=\left[\eta_{j,0},\eta_{j,1},\cdots,\eta_{j,d}\right]^{\mathrm{T}}$ of the jth local MR is estimated using the regularized least squares algorithm.

$$\boldsymbol{\eta}_j=\left(\boldsymbol{R}_j^{\mathrm{T}}\boldsymbol{R}_j+\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{R}_j^{\mathrm{T}}\boldsymbol{y}_j, \tag{5}$$

where the regularization coefficient $\lambda(0\leqslant\lambda\leqslant1)$ is a small constant; $\boldsymbol{I}\in\mathbf{R}^{(d+1)(d+1)}$ is an identity vector; $\boldsymbol{R}_j$ and $\boldsymbol{y}_j$ are defined as

$$\boldsymbol{y}_j=\begin{bmatrix}y_1^j\\y_2^j\\\vdots\\y_{n_j}^j\end{bmatrix},\boldsymbol{R}_j=\begin{bmatrix}1&x_{1,1}^j&\cdots&x_{1,d}^j\\1&x_{2,1}^j&\cdots&x_{2,d}^j\\\vdots&\vdots&\vdots&\vdots\\1&x_{n_j,1}^j&\cdots&x_{n_j,d}^j\end{bmatrix}. \tag{6}$$

The parameters are frozen as training is completed. In the test phase, given the input vector $\boldsymbol{x}(t)$, the predicted output $\hat{y}(t)$ is

$$\hat{y}(t)=\boldsymbol{x}(t)\boldsymbol{\eta}_j(t), \tag{7}$$

where $\boldsymbol{\eta}_{j^*}(t)$ is the coefficient vector of the winning neuron.

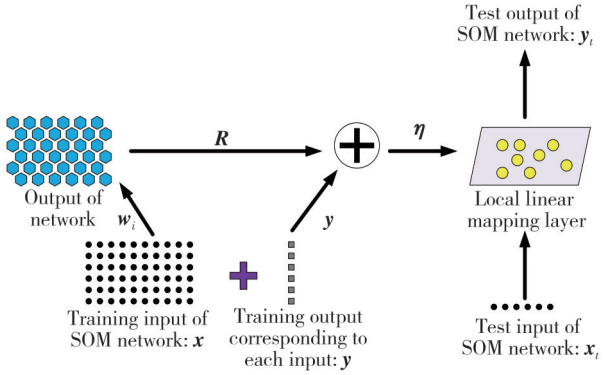


Fig. 2 Dynamic modeling structure of SOM-MR

In contrast, when only one neuron remains in the output layer of the SOM network, SOM-MR degenerates into a single MR Model.

2 SOM-based regional modeling methods

SOM is conceptually distinguished from clustering, and the implementations of SOM are different from any clustering algorithms. SOM attempts to extract and visually display the topological structure of high-dimensional input data, while clustering simply divides the input data into multiple groups. The two-level algorithm used in this paper was to cluster the prototypes after SOM training, rather than clustering the original data directly[14]. Combining the evaluable two-level algorithm and optimized KELM, RKELM and RMR methods were proposed.

2.1 Two-level algorithm

Both contiguity-constrained and agglomerative algorithms can be applied to cluster the trained SOM network. Based on the advantages that the clustering algorithm does not depend on previous clusters and implicit assumptions can be made in the form of clusters, FCM is adopted in the two-level algorithm.

First, for each mapping unit of SOM, a Voronoi cell V_i is defined as

$$V_i=\left\{\boldsymbol{x}\left|\|\boldsymbol{x}-\boldsymbol{w}_i\|\leqslant\|\boldsymbol{x}-\boldsymbol{w}_j\|,\forall j=1,2,\cdots,P,j\neq i\right.\right\}, \tag{8}$$

where $\boldsymbol{x}$ represents the input vector mapped into V_i.

An evaluable FCM was employed to cluster the trained SOM network.

Given a sample set $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$, FCM divides the set D into a number of clusters C by minimizing the clustering loss function J_f defined by the membership.

$$J_f=\sum_{j=1}^{C}\sum_{i=1}^{N}\left[\mu(\boldsymbol{x}_i)\right]^{\mathrm{b}}\|\boldsymbol{x}_i-\boldsymbol{m}_j\|^2, \tag{9}$$

where C is the number of clusters; $\boldsymbol{m}_j(j=1,2,\cdots,C)$ represents the jth cluster center, which possesses the same feature dimensions as $\boldsymbol{x}$; $\mu_j(\boldsymbol{x}_i)$ represents the membership function of the ith sample to the jth cluster; b is a constant, which can be used to adjust the fuzzy degree of the clustering results.

In FCM, the sum of membership degrees of a sample to all clusters is 1, which can be expressed as

$$\sum_{j=1}^{C}\mu_j(\boldsymbol{x}_i)=1,i=1,2,\cdots,N. \tag{10}$$

Let the partial derivatives J_f sub $\boldsymbol{m}_j$ and $\mu_j(\boldsymbol{x}_i)$ be 0, respectively, and the solutions are

$$\boldsymbol{m}_j=\frac{\sum_{i=1}^{N}\left[\mu_j(\boldsymbol{x}_i)\right]^{\mathrm{b}}\boldsymbol{x}_i}{\sum_{i=1}^{N}\left[\mu_j(\boldsymbol{x}_i)\right]^{\mathrm{b}}},i=1,2,\cdots,N;j=1,2,\cdots,C, \tag{11}$$

$$\mu_j(\boldsymbol{x}_i)=\frac{\left(1/\|\boldsymbol{x}_i-\boldsymbol{m}_j\|^2\right)^{\frac{1}{\mathrm{b}-1}}}{\sum_{k=1}^{C}\left(1/\|\boldsymbol{x}_i-\boldsymbol{m}_k\|^2\right)^{\frac{1}{\mathrm{b}-1}}}. \tag{12}$$

As opposed to selection of C, the maximum size of clusters in FCM is predefined as $C_{\max}$, and then the clustering performance under each C, from 2 to $C_{\max}$, are calculated with DBI. The smaller the DBI, the better the performance of clustering. The selection of the optimal C_{opt} determined by DBI is defined as

$$C_{\text{opt}} = \arg\min_{C} DBI(\boldsymbol{W}, \boldsymbol{M}_C), C = 2, 3, \cdots, C_{\max}, \quad (13)$$

where $\boldsymbol{M}_C = \{\boldsymbol{m}_j\}_{j=1}^{C}$, $\boldsymbol{m}_j \in \mathbf{R}^d$ represents a set of C clusters in FCM; $\boldsymbol{W} = \{\boldsymbol{w}_i\}_{i=1}^{P}$, $\boldsymbol{w}_i \in \mathbf{R}^d$ refers to the collection of P prototypes in SOM. Fig. 3 shows the flowchart of the two-level algorithm.

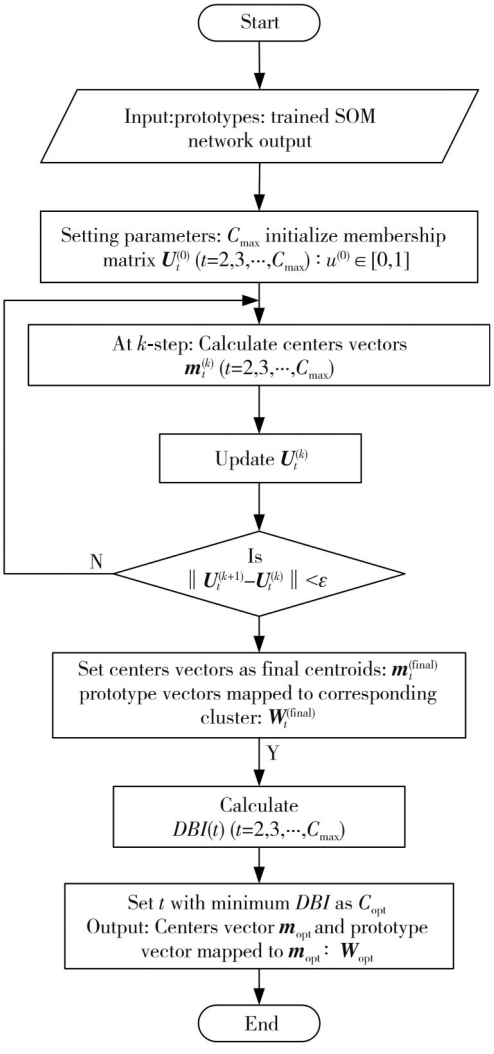


Fig. 3 Flowchart of two-level algorithm

2.2 KELM method

The KELM and ELM methods have several different characteristics. KELM uses the kernel function to represent the unknown nonlinear feature mapping of the hidden layer, and the number of hidden layer nodes does not need to be given in advance. Both ELM and KELM can be solved by the regularized least squares algorithm. Under the premise of ensuring the learning speed, the generalization ability of KELM is superior[22].

Specify N input-output samples, $(\boldsymbol{x}_i, t_i) \in \mathbf{R}^n \times \mathbf{R}^1$, $i = 1, 2, \cdots, N$. If there are L hidden layer nodes, the single-hidden layer feedforward NN(SLFN) can be embodied as

$$\sum_{i=1}^{L} \beta_i \vartheta(\boldsymbol{\omega}_i \boldsymbol{x}_j + b_i) = O_j,\ j = 1, 2, \cdots, N, \quad (14)$$

where $\vartheta(\boldsymbol{x})$ is the activation function; $\boldsymbol{\omega}_i = [\omega_{i,1}, \omega_{i,2}, \cdots, \omega_{i,n}]^{\mathrm{T}}$ is the input weight vector; β_i denotes the output weight vector between the ith hidden layer node and the output layer node; b_i is the bias of the ith hidden layer nodes and $\boldsymbol{\omega}_i \boldsymbol{x}_j$ represents the scalar product of $\boldsymbol{\omega}_i$ and $\boldsymbol{x}_j$; O_i is the actual output of SLFN.

According to the ELM theory, it is assumed that there must be β_i, $\boldsymbol{\omega}_i$, and b_i that make the equation true.

$$\sum_{i=1}^{L} \beta_i \vartheta(\boldsymbol{\omega}_i \boldsymbol{x}_j + b_i) = t_j,\ j = 1, 2, \cdots, N. \quad (15)$$

The matrix form is

$$H\beta = T, \quad (16)$$

where $\boldsymbol{H}$ is the output matrix of the hidden layer and $\boldsymbol{\beta}$ is the output weight matrix; the desired output is $\boldsymbol{T}$. The explicit expressions of $\boldsymbol{H}$, $\boldsymbol{\beta}$, and $\boldsymbol{T}$ are

$$\boldsymbol{H} = (\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \cdots, \boldsymbol{\omega}_L; b_1, b_2, \cdots, b_L; \boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L) = \begin{bmatrix} \vartheta(\boldsymbol{\omega}_1 \boldsymbol{x}_1 + b_1) & \cdots & \vartheta(\boldsymbol{\omega}_L \boldsymbol{x}_1 + b_L) \\ \vdots & \vdots & \vdots \\ \vartheta(\boldsymbol{\omega}_1 \boldsymbol{x}_N + b_1) & \cdots & \vartheta(\boldsymbol{\omega}_L \boldsymbol{x}_N + b_L) \end{bmatrix}_{N \times L}, \quad (17)$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^{\mathrm{T}} \\ \vdots \\ \beta_L^{\mathrm{T}} \end{bmatrix}_{L \times 1}, \quad \boldsymbol{T} = \begin{bmatrix} t_1^{\mathrm{T}} \\ \vdots \\ t_N^{\mathrm{T}} \end{bmatrix}_{N \times 1}. \quad (18)$$

The traditional solutions based on gradient descent required all parameters to be adjusted during the iterations. Nevertheless, in the context of ELM, it could be transformed into solving the problem of linear system $\boldsymbol{H\beta} = \boldsymbol{T}$. Thus, the least squares solution of the network output weight is

$$\hat{\beta} = H^{\mathrm{T}}(HH^{\mathrm{T}} + \eta_k I)^{-1} T, \quad (19)$$

where η_k is the regularization coefficient; $\boldsymbol{I}$ is an identity matrix. Therefore, the output function of ELM can be defined as

$$\hat{y}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta}. \quad (20)$$

In ELM, if the feature mapping function $\boldsymbol{h}(\boldsymbol{x})$ is unknown, it is an effective way to introduce a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ to replace the unknow kernel matrix

$\boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{h}(\boldsymbol{x}_j)$, which is also the main idea of the KELM method. The kernel function in KELM is defined as

$$Q_{\mathrm{ELM}}(i,j)=\boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{h}(\boldsymbol{x}_j)=K(\boldsymbol{x}_i,\boldsymbol{x}_j), \tag{21}$$

and the predicted output of KELM is

$$\hat{y}(\boldsymbol{x})=\begin{bmatrix}K(\boldsymbol{x},\boldsymbol{x}_1)\\ \vdots\\ K(\boldsymbol{x},\boldsymbol{x}_N)\end{bmatrix}\left(Q_{\mathrm{ELM}}+\eta_k\boldsymbol{I}\right)^{-1}T. \tag{22}$$

In the cases of kernel-based methods, the selection of kernel function is not specified, and more kernel functions can be secured using the linear combination or direct product with varying kernel functions. It is worth noting that under the general smoothing assumptions, the Gaussian radial basis function (RBF) tends to perform better, thus the general form of Gaussian RBF is adopted.

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{\gamma^2}\right), \tag{23}$$

where γ represents the radius of the Gaussian RBF. The adjustments of γ and η_k in KELM are easy to implement, which owns an effective impact on improving the performance. Additionally, to further improve the forecast performance, differential evolution algorithm was employed to optimize the parameters of ELM and KELM[22].

2.3 Regional models

The implementations of SOM-based regional MR and KELM (RMR and RKELM) models were discussed step by step.

Step 1: Setting the hyper-parameters. In SOM training, the minimum porotypes were generally set as $P=5\sqrt{N}$, where N was the size of training data set. And the two-dimensional hexagonal topology was constructed. The maximum number of clusters in FCM was set to $C_{\max}=\sqrt{P}$.

Step 2: Training of the SOM network. As mentioned in Section 3.1, a pure SOM completed the initial partitions of training data set. Then, for each map unit, Eq. (8) was used to obtain the corresponding Voronoi cells.

Step 3: Clustering of the trained SOM network. An evaluable FCM was adopted to cluster the trained SOM network. As mentioned before, once the optimal cluster size C_{opt} was determined, the optimal regional partitions were equivalent to the clustering of trained SOM network. The rth cluster of SOM prototypes was composed of all neuron vectors $\boldsymbol{w}_i$, which were mapped onto the cluster $\boldsymbol{C}_r$, $r=1,2,\cdots,C_{\mathrm{opt}}$. The set of SOM prototypes associated with the rth cluster in FCM is defined as

$$\boldsymbol{C}_r=\left\{\boldsymbol{w}_i\in\mathbf{R}^d\middle|\,\|\boldsymbol{w}_i-\boldsymbol{m}_r\|<\|\boldsymbol{w}_i-\boldsymbol{m}_j\|,\ \forall j=1,2,\cdots,C_{\mathrm{opt}},j\neq r\right\}. \tag{24}$$

Step 4: Data partitions. After the above steps, all inputs were still stored in Voronoi cells. To further divide the data into determined C_{opt} data partitions, let $\boldsymbol{F}_i$ be the data set mapped to the ith region, and N_i represent the number of samples in $\boldsymbol{F}_i$. $\boldsymbol{F}_i$ is composed of $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{N_i}$ and the SOM prototypes to which these inputs belong to are mapped to the ith region.

Step 5: Constructing the KELM model in different regions. Finally, the training data set was divided into C_{opt} subsets. In each partition, an optimized RKELM model was constructed using the data subset mapped to this region.

In the modeling process, more precisely, in step 5, the SOM's regional partitioning was combined with the optimized MR to obtain the SOM-based regional MR model, i.e., the RMR model.

3 Experiment

The forecasts were carried out on 12 data subsets (divided by month) of three solar power plants. Data from each plant in 2012 and 2013 was collected as a training dataset, and data from 2014 was used for predictive analysis. Single global models, MR, ELM, and KELM were used as the benchmark methods to evaluate the effectiveness of the proposed RMR and RKELM models. In addition, to perfect the comparisons, the follow-up parts also provided the forecasts of the local KELM model, denoted as SOM-KELM, coupled with DKELM which combined single FCM and KELM.

3.1 Data processing

The experimental data were derived from the PV power forecasting part of GEFCom2014, which was the real-world data of three solar power plants located in a certain region of Australia. The data offered by the institution consisted of two parts. The first part recorded the daily PV power generation (from 00:00 to 23:00) of each solar power plant from 2012 to 2014, with a time resolution of 1 h. The second part was the numerical records of 12 independent meteorological variables provided by the European center for medium-range weather forecasts (ECMWF)[20]. The descriptions of the

12 variables are shown in Table 1.

Table 1 Descriptions of 12 independent meteorological factors

Label	Description
VAR78	Total column liquid water (TCLW)/(kg•m^{-2})
VAR79	Total column ice water (TCIW)/(kg•m^{-2})
VAR134	Surface pressure(SP)/Pa
VAR157	Relative humidity at 100 kPa(r)/%
VAR164	Total cloud cover (TCC)/%
VAR165	10-metre U wind component (10U)/(m•s^{-1})
VAR166	10-metre V wind component (10V)/(m•s^{-1})
VAR167	2-metre temperature (2T)/K
VAR169	Surface solar rad down (SSRD)/(J•m^{-2})
VAR175	Surface thermal rad down (STRD)/(J•m^{-2})
VAR178	Top net solar rad (TSR)/(J•m^{-2})
VAR228	Total precipitation (TP)/m

Since different variables with varying value ranges and metrics, all selected inputs were mapped to the interval of [0, 1]. There is a very strong correlation between photovoltaic power generation and solar irradiance, and this correlation is universal regardless of the geographical location and weather conditions of photovoltaic power plants[1]. Therefore, it was clear to select VAR169 and VAR178 as the inputs of the proposed RKELM. On this basis, the Filter-based ReliefF method and the wrapper-based sequential forward selection (SFS) method were used to select the remaining features[23], and the results of the two methods were comprehensively considered to determine the final inputs. Tables 2 and 3 show the selected meteorological factors in solar power plant #1 using SFS and ReliefF methods. VAR157, 166, 167, 169, 175, 178, and 228 were finally determined as the input of the model. And the results of power plants #2 and #3 were the same as plant #1.

Table 2 Feature selections using SFS method

Stage	Hit feature	Score	Selected feature
1	VAR175	0.980 8	VAR169/VAR175/VAR178
2	VAR228	0.979 4	VAR169/VAR175/VAR178/VAR228

Table 3 Feature selections using ReliefF method

Rank	Hit feature	Weight
1	VAR175	0.041 4
2	VAR166	0.003 3
3	VAR157	0.003 1
4	VAR167	0.001 1
5	VAR165	−0.004 0
6	VAR79	−0.004 1
7	VAR78	−0.004 7
8	VAR134	−0.010 1
9	VAR228	−0.010 5
10	VAR164	−0.012 0

3.2 Performance evaluation metrics

In order to evaluate the forecasting performance of proposed models, the mean absolute error (MAE), the root means square error (RMSE), and the coefficient of determination (R^2) were mainly adopted.

$$MAE=\frac{1}{N}\sum_{i=1}^{N}\left|y(i)-\hat{y}(\boldsymbol{x}_i)\right|, \tag{25}$$

$$RMSE=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[y(i)-\hat{y}(\boldsymbol{x}_i)\right]^2}, \tag{26}$$

$$R^2=1-\frac{\sum_{i=1}^{N}\left[y(i)-\hat{y}(\boldsymbol{x}_i)\right]^2}{\sum_{i=1}^{N}\left[y(i)-\bar{y}\right]^2}, \tag{27}$$

where N is the size of test sets; $y(i)$ represents the actual output and $\hat{y}(\boldsymbol{x}_i)$ represents the predicted PV power generation; $\bar{y}$ represents the average of the actual PV power generations. $R^2(0<R^2<1)$ reflects the fitting degree of the forecasting model. The closer R^2 is to 1, the higher the accuracy of the model.

3.3 Experimental results and analysis

For each solar power plant, the prediction was made one hour in advance. First, the data was divided into different subsets by month, then multiple prediction models were built, and finally the results were integrated.

Table 4 shows the settings of hyper-parameters in SOM for local (SOM-MR and SOM-KELM) and regional models (RMR and RKELM), in which N represents the size of training data set. The neuron weights in competitive layer were randomly initialized using samples. In DKELM, the maximum cluster was set as $C_{\max}=\sqrt{N}$.

Table 4 Settings of SOM hyper-parameter for local and regional models

Parameter	Local model	Regional model
Prototype size	$5\sqrt{N}$	$5\sqrt{N}$
Initial and final learning rate(α_0)	0.1, 0.01	0.8, 0.01
Initial and final radius(σ_0)	$5\sqrt{N}/4$, 0.01	$5\sqrt{N}/2$, 0.01
Clustering of trained SOM		An evaluable FCM
Maximum clusters		$5\sqrt{N}$

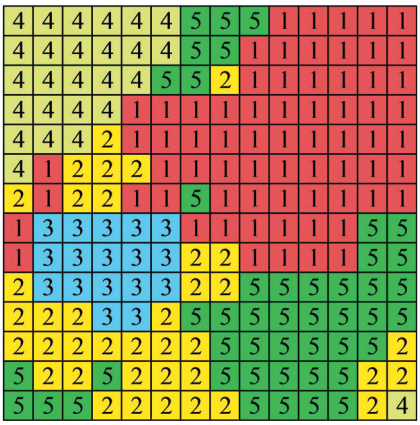
The weights and bias of input to hidden layer in ELM were randomly initialized, and Sigmoid was selected as the activation function of the hidden layer neurons. As mentioned in Section 3.2, the kernel function of KELM choosed the general form of Gaussian kernel function, and the parameters γ, η_k and L were optimized by differential evolution algorithm. Table 5 shows the optimized parameters of ELM and KELM in plant #1.

The training data set in plant#1 (April) was taken as an example, the data size was 1 440, and the number of

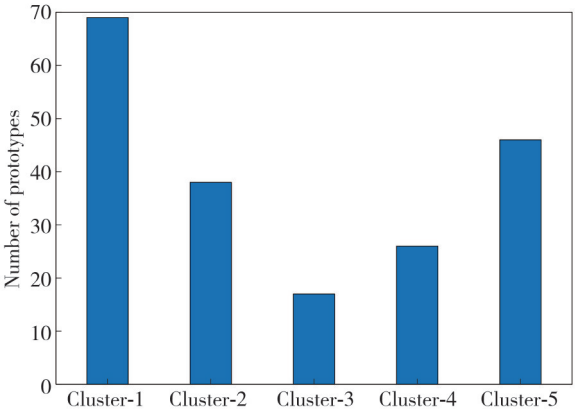
prototypes in SOM is set to $P=196$, the maximum number of clusters was $C_{max}=14$. Fig. 4 shows the DBI results under different clustering sizes. It can be learned that DBI reaches a minimum when the number of clusters is 5. Therefore, the optimal cluster size in plant#1 (April) was set to $C_{opt}=5$. Fig. 4 additionally shows the clustering results using single FCM. It was concluded that the effectiveness of two-level algorithm was significantly superior than single FCM.

Table 5 Optimized parameters of ELM and KELM in plant #1

Month	L	η_k	γ
January	38	82.429 0	0.610 1
February	34	84.832 6	1.554 4
March	29	77.512 3	1.045 6
April	37	70.817 3	0.538 1
May	38	5.888 4	0.169 3
June	38	47.591 1	0.724 0
July	36	64.891 8	1.311 3
August	38	25.773 1	0.831 8
September	36	33.355 6	1.240 0
October	38	78.757 4	1.296 4
November	33	52.596 8	0.335 0
December	34	62.683 7	0.655 3



(a) Clustering results of SOM prototypes



(b) Number of prototypes in each cluster

Fig. 5 Clustering of trained-SOM network

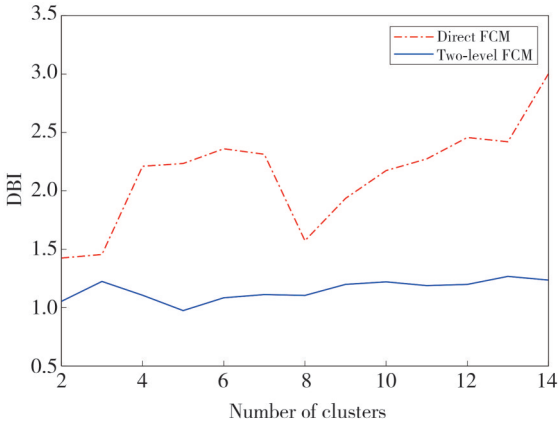


Fig. 4 Results of DBI under different number of clusters

Fig. 5 displays the clustering of trained SOM network. The grid (prototypes in SOM) size was 196. In Fig. 5 (a), the number in each small square indicates the cluster to which the prototype belongs, and the adjacent prototypes arranged in the output layer are clustered into the same category, which echoes the neighborhood update strategy of SOM network and verifies the effectiveness of the proposed two-level algorithm. Fig. 5 (b) shows the number of prototypes in each cluster.

First, to evaluate the performance of the proposed models in different plants, the forecasts of each plant in April, 2014 was taken as an example.

And the error metrics of all forecasts using different methods in plants #1, #2, and #3 are shown in Tables 6—8.

Table 6 Error metrics of different methods in plant #1

Method	MAE	RMSE	R^2
MR	0.115 6	0.160 9	0.530 7
ELM	0.090 9	0.153 4	0.573 5
KELM	0.076 4	0.113 9	0.764 6
DKELM	0.054 8	0.090 3	0.852 0
SOM-MR	0.072 4	0.127 5	0.705 1
SOM-KELM	0.038 1	0.081 1	0.880 7
RMR	0.044 7	0.072 0	0.905 9
RKELM	0.030 5	0.060 6	0.933 3

Table 7 Error metrics of different methods in plant #2

Method	MAE	RMSE	R^2
MR	0.106 7	0.144 7	0.639 0
ELM	0.096 4	0.138 1	0.670 8
KELM	0.065 9	0.108 9	0.795 6
DKELM	0.059 7	0.103 7	0.814 5
SOM-MR	0.074 1	0.129 6	0.710 0
SOM-KELM	0.039 5	0.079 3	0.891 4
RMR	0.044 3	0.073 6	0.906 7
RKELM	0.030 8	0.056 0	0.945 8

It can be seen that in any plant, the proposed RMR and RKELM have reached the highest accuracy. Compared with a single MR in every plant, the MAE of proposed regional RMR model was reduced by 61.33%, 58.48%, and 54.93%, respectively. The RMSE was

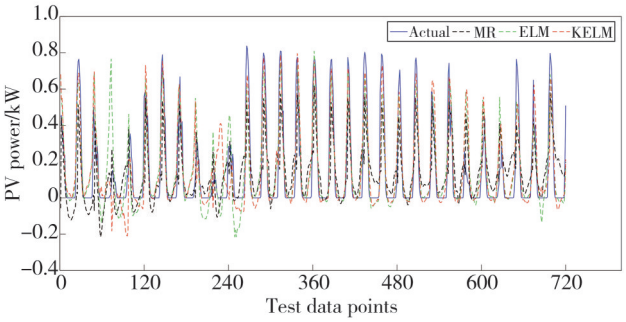
reduced by 55.25%, 49.14%, and 41.81%, respectively. And the accuracy of R^2 was improved by 70.70%, 41.89%, and 45.21%, respectively. Similarly, compared with local SOM-MR model, the MAE and RMSE of proposed RMR model were reduced with varying degrees in all plants, and R^2 has been improved. Significantly, the proposed regional RKELM model had the highest accuracy in any plant. The average results of MAE, RMSE, and R^2 generated by RKELM were 0.032 0, 0.060 1, and 0.936 5, respectively. The accuracy of RKELM was slightly higher than RMR, mainly because the learning ability of NNs were superior in processing nonlinear systems, especially for the meteorological factors with strong uncertainty and nonlinearity. In addition, the SOM-KELM and DKELM models based on pure SOM and direct clustering have achieved satisfactory forecasting performance, but the accuracy was inferior to the proposed RMR and RKELM. Although the forecasting performance of proposed SOM-MR and SOM-KELM models based on the local modeling was not as good as the two regional modeling methods mentioned above, the accuracy was significantly improved compared with a single MR or KELM.

Table 8 Error metrics of different methods in plant #3

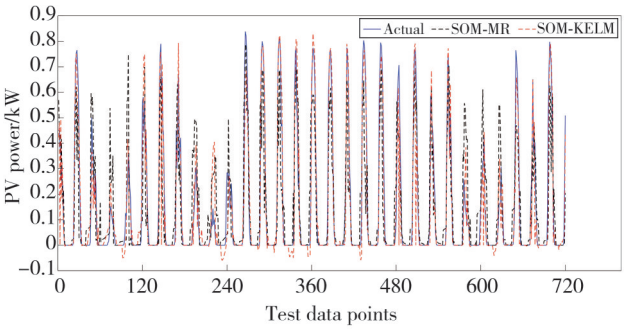
Method	MAE	RMSE	R^2
MR	0.114 7	0.154 5	0.593 0
ELM	0.098 4	0.144 8	0.639 8
KELM	0.070 5	0.113 6	0.778 3
DKELM	0.064 6	0.110 0	0.792 2
SOM-MR	0.079 1	0.125 3	0.730 5
SOM-KELM	0.051 7	0.089 9	0.861 1
RMR	0.042 5	0.073 2	0.907 9
RKELM	0.034 6	0.063 6	0.930 5

In addition, the *k*-nearest neighbor and kernel density estimator method was proposed to perform the forecasts in different plants (the experimental data were the same as data used in this paper), and the results of RMSE for each plant were 0.088 1, 0.077 6, and 0.075 7, respectively[24]. The error metric of RMSE conducted by RMR and RKELM were 0.072 0, 0.073 6, 0.089 9, and 0.060 6, 0.056 0, 0.063 6. It was clearly that the proposed methods obtained a smaller RMSE and higher accuracy.

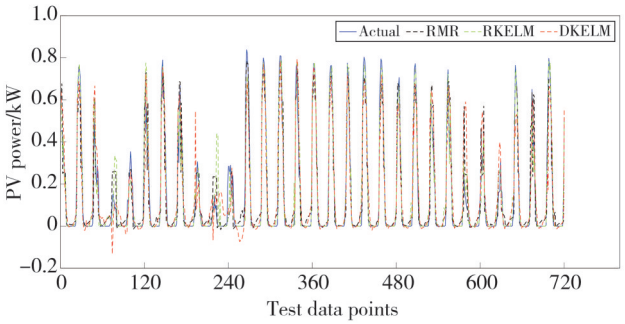
Fig. 6 shows the comparisons between the predicted and actual outputs in solar power plant #1, April, 2014. Both proposed RMR and RKELM methods can fit the actual power curve, and the predicted output of RKELM has a small deviation from the measured power at a few peak power points. And it was observed that other comparison methods could not accurately approximate the actual output, especially at zero-power-point and peak-power-point.



(a) Actual PV power and predicted PV power using MR, ELM, and KELM



(b) Actual PV power and predicted PV power using SOM-MR and SOM-KELM



(c) Actual PV power and predicted PV power using KMR, RKELM and DKELM

Fig. 6 Comparison curves between predicted and actual PV power generations in plant#1

Most of contrast methods showed a weak fitting, mainly manifested in the excessive deviation in a certain area. Such results indicated that RMR and RKELM methods could quickly capture the existing power changes and give the response, which potentially verified that the two-level approach was effective in clustering training data set.

Fig.7 shows the forecasting errors of different methods in plant #1, April, 2014. It can be seen that the RKELM method has the smallest error, and only a few points have a small deviation. The results demonstrated the superiority of the proposed RMR and RKELM models.

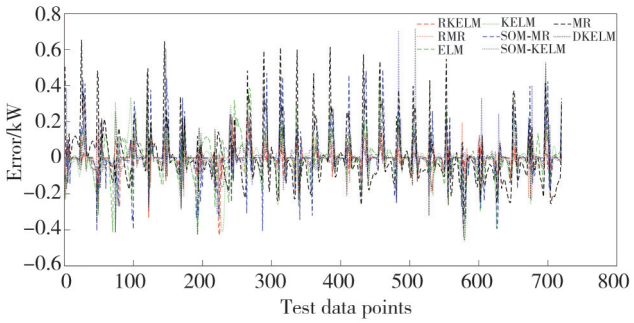
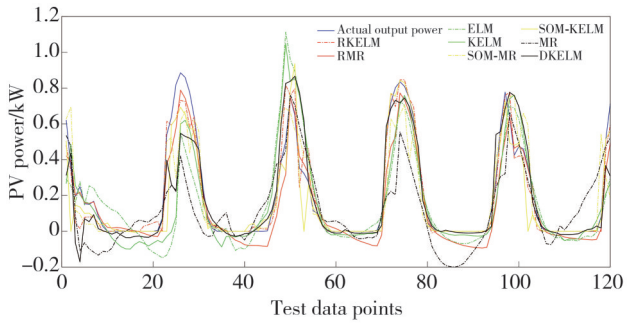


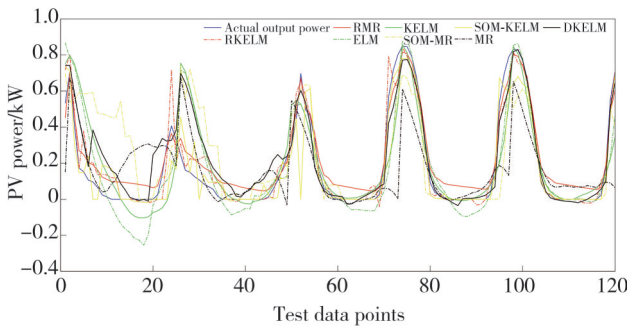
Fig. 7 Errors between predicted and actual PV power generations in plant#1

In order to investigate the influence of seasonal factors on the forecasting of PV power generation, the solar power plant#1 was taken as an example, and the forecasts of different methods in different seasons were compared. The data of solar power plant#1 from 2012 and 2013 were used as training data set. They were October in spring, January in summer, April in autumn, and July in winter. A random sample of five days in corresponding month of solar power plant#1 from 2014 was employed as test dataset.

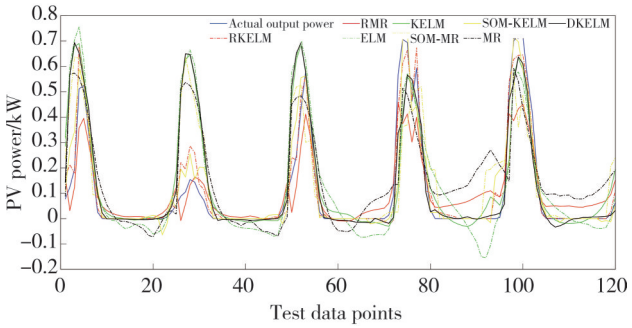
Table 9 shows the error metrics of all forecasts in different seasons. In order to show the indicators more clearly, Fig. 8 displays the statistical results of MAE, RMSE, and R^2 for all methods in all seasons. Overall, the forecasts in autumn and winter were better than that in spring and summer for most models. Possibly because the weather changes in autumn and winter are gentle and fewer large fluctuations exist.



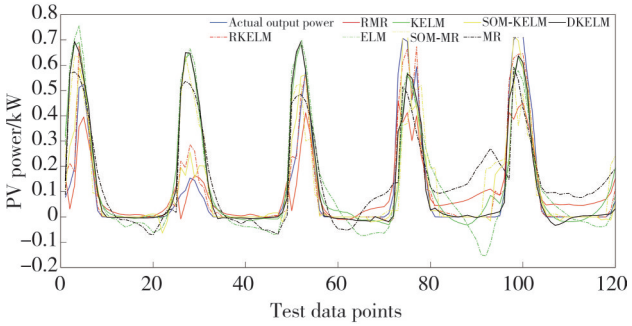
(a) Spring



(b) Summer



(c) Autumn



(d) Winter

Fig. 8 Comparisons between predicted and actual PV power generations of different methods in different seasons

In addition, according to Fig.8 and Table 9, RKELM method has the highest precision among all forecasting methods, followed by RMR. In four seasons, the MAE and RMSE of proposed RKELM were the smallest, and the R^2 exceeded 0.9, which was the closest to 1. The performance of other methods was relatively poor, and various indexes had different degrees of deviation. The forecasts in different seasons have verified the effectiveness of proposed RMR and RKELM models.

The comparisons between the predicted and actual PV power generations in different seasons are shown in Fig.8. It can be seen that different types of photovoltaic power generation are different in different seasons. Sometimes the actual power curve changes rapidly, and the sudden peak drop is more. While the actual power curve changes gently and the sudden peak plunge is less. In either case, the proposed RKELM and RMR can quickly respond to power changes and accurately approximate the outputs, the forecasting accuracy is the highest.

The forecasting errors of different methods in different seasons are shown in Fig. 9. In four seasons, the errors generated by RKELM and RMR fluctuated around zero, and the change ranges were the narrowest among all methods.

The above-mentioned forecasting experiments verified the superiority of the proposed RMR and RKELM methods. Further, RKELM was recommended for accurate short-term PV power forecasting. In addition, the

forecasting results also showed that compared with single clustering method (for example, FCM or single SOM), the evaluable two-level algorithm that combined SOM and FCM network was more effective in data clustering.

Table 9 Error metrics of MAE, RMSE, and R^2 for each method in different seasons

Season	Indexes	Error value							
		MR	ELM	KELM	SOM-MR	SOM-KELM	DKELM	RMR	RKELM
Spring (October)	MAE	0.144 0	0.124 6	0.102 1	0.071 5	0.046 2	0.074 9	0.062 0	0.040 0
	RMSE	0.203 7	0.196 7	0.172 3	0.134 2	0.096 8	0.136 1	0.087 6	0.070 2
	R^2	0.452 1	0.489 1	0.608 3	0.762 4	0.876 3	0.755 5	0.898 6	0.934 9
Summer (January)	MAE	0.141 9	0.103 4	0.107 1	0.137 8	0.054 3	0.082 6	0.064 0	0.047 4
	RMSE	0.212 6	0.157 2	0.168 2	0.216 0	0.101 2	0.128 0	0.070 2	0.075 6
	R^2	0.350 8	0.645 2	0.593 7	0.329 8	0.852 9	0.764 7	0.929 2	0.917 9
Autumn (April)	MAE	0.087 9	0.071 8	0.106 7	0.085 6	0.043 8	0.062 3	0.036 8	0.032 9
	RMSE	0.126 1	0.099 7	0.145 2	0.142 1	0.081 7	0.110 2	0.062 5	0.043 0
	R^2	0.549 8	0.718 8	0.403 9	0.429 0	0.811 3	0.656 4	0.889 5	0.947 6
Winter (July)	MAE	0.111 3	0.099 3	0.077 7	0.082 4	0.033 6	0.064 1	0.059 6	0.024 3
	RMSE	0.152 4	0.156 1	0.143 0	0.151 0	0.067 0	0.136 0	0.087 9	0.044 1
	R^2	0.399 4	0.369 7	0.471 1	0.414 0	0.883 9	0.521 6	0.800 2	0.949 7

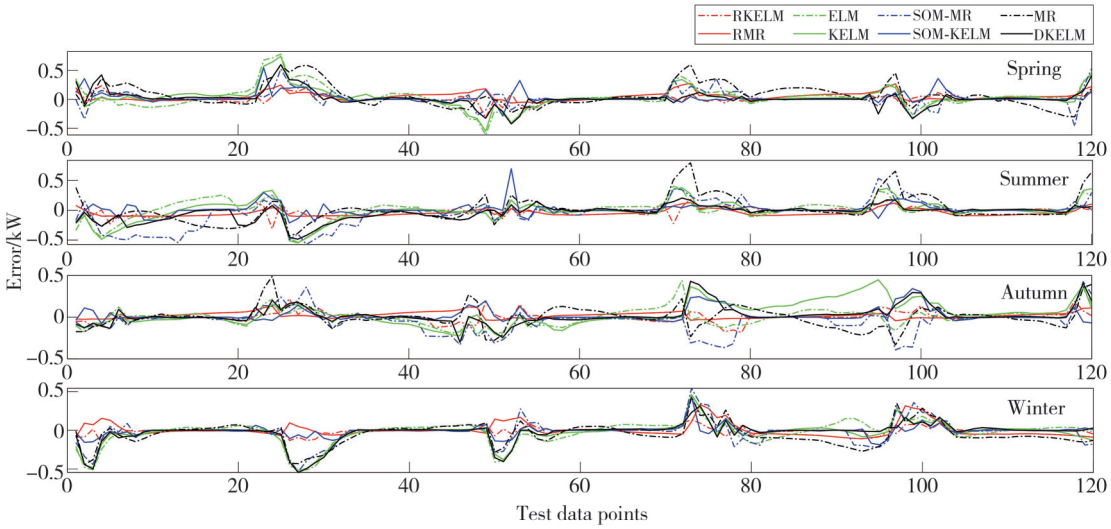


Fig. 9 Forecasting error of different methods in different seasons

4 Conclusions

Aiming at improving the forecasting accuracy of short-term PV power generations, two kinds of forecasting approaches based on pure SOM and two-level algorithm were proposed, including SOM-MR, SOM-KELM, RMR, and RKELM. The proposed methods were applied to the short-term PV power forecasts of three different solar power plants provided by the GEFCom2014. Moreover, comparisons between different methods were implemented. MR, ELM, and KELM were used as benchmark methods. Compared with the single global MR and KELM, the forecasting results showed that the accuracy of the proposed combined SOM-FCM and KELM method has been significantly improved. Among these, according to the evaluation indexes, the RMR and RKELM methods have distinguished performance from the rest, which obtained the highest degree of fit to the actual outputs. However, the proposed models failed to quantitatively analyze and process the uncertainty exist in PV power generation. For future research directions, it is possible to consider combining multi-core SOM with other more competitive forecasting methods to deal with uncertainties in photovoltaic power generation.

Acknowledgement

This work was supported by National Natural Science

Foundation of China (No. 51467008); Gansu Provincial Department of Education Industry Support Program (No.2021CYZC-32).

Declaration of conflicting interests

The authors have no conflict of interests related to this publication.

References

[1] DAS U K, TEY K S, SEYEDMAHMOUDIAN M, et al. Forecasting of photovoltaic power generation and model optimization: A review. Renewable and Sustainable Energy Reviews, 2018, 81: 912-928.

[2] SOBRI S, KOOHI-KAMALI S, RAHIM N A. Solar photovoltaic generation forecasting methods: A review. Energy Conversion and Management, 2018, 156: 459-497.

[3] PAN M, LI C, GAO R, et al. Photovoltaic power forecasting based on a support vector machine with improved ant colony optimization. Journal of Cleaner Production, 2020, 277: 123948.

[4] LI P, ZHOU K, LU X, et al. A hybrid deep learning model for short-term PV power forecasting. Applied Energy, 2020, 259: 114216.

[5] YANG Z, MOURSHED M, LIU K, et al. A novel competitive swarm optimized RBF neural network model for short-term solar power generation forecasting. Neurocomputing, 2020, 397: 415-421.

[6] QU J, QIAN Z, PEI Y. Day-ahead hourly photovoltaic power forecasting using attention-based CNN-LSTM neural network embedded with multiple relevant and target variables prediction pattern. Energy, 2021, 232: 120996.

[7] FU W, ZHANG K, WANG K, et al. A hybrid approach for multi-step wind speed forecasting based on two-layer decomposition, improved hybrid DE-HHO optimization and KELM. Renewable Energy, 2021, 164: 211-229.

[8] YE R, DAI Q. MultiTL-KELM: A multi-task learning algorithm for multi-step-ahead time series prediction. Applied Soft Computing, 2019, 79: 227-253.

[9] XIAO L, SHAO W, JIN F, et al. A self-adaptive kernel extreme learning machine for short-term wind speed forecasting. Applied Soft Computing, 2021, 99: 106917.

[10] WU S, CHOW T W. Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density. Pattern Recognition, 2004, 37 (2): 175-188.

[11] LEHTOKANGAS M, SAARINEN J, KASKI K, et al. A network of autoregressive processing units for time series modeling. Applied Mathematics and Computation, 1996, 75 (2-3): 151-165.

[12] BARRETO G A, ARAUJO A F R. Identification and control of dynamical systems using the self-organizing map. IEEE Transactions on Neural Networks, 2004, 15 (5): 1244-1259.

[13] SIMON G, LENDASSE A, COTTRELL M, et al. Time series forecasting: Obtaining long term trends with self-organizing maps. Pattern Recognition Letters, 2005, 26 (12): 1795-1808.

[14] SOUZA A H J, BARRETO G A, CORONa F. Regional models: A new approach for nonlinear system identification via clustering of the self-organizing map. Neurocomputing, 2015, 147: 31-46.

[15] VESANTO J, ALHONIEMI E. Clustering of the self-organizing map. IEEE Transactions on Neural Networks, 2000, 11 (3): 586-600.

[16] HUANG G B, ZHOU H, DING X, et al. Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2011, 42 (2): 513-529.

[17] ANDRAS P. Kernel-kohonen networks. International Journal of Neural Systems, 2002, 12 (2): 117-135.

[18] YAN B, XU N, XU L P, et al. An improved partitioning algorithm based on FCM algorithm for extended target tracking in PHD filter. Digital Signal Processing, 2019, 90: 54-70.

[19] HALKIDI M, BATISTAKIS Y, VAZIRGIANNIS M. On clustering validation techniques. Journal of Intelligent Information Systems, 2001, 17 (2): 107-145.

[20] HONG T, PINSON P, FAN S, et al. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. International Journal of Forecasting, 2016, 32 (3): 896-913.

[21] BHAMMER B, HITZLER P. Perspectives of neural-symbolic integration. Berlin, Heidelberg: Springer, 2007.

[22] MATIAS T, SOUZA F, ARAÚJO R, et al. Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine. Neurocomputing, 2014, 129: 428-436.

[23] ZHU P, ZHU W, HU Q, et al. Subspace clustering guided unsupervised feature selection. Pattern Recognition, 2017, 66: 364-374.

[24] ZHANG Y, WANG J. GEFCom2014 probabilistic solar power forecasting based on k-nearest neighbor and kernel density estimator//IEEE Power & Energy Society General Meeting, July 26-30, 2015, Denver, Colorado, USA. New York: IEEE. 2015: 1-5.

基于SOM-FCM和KELM组合方法的短期光伏功率预测

刘齐波，李　军*

兰州交通大学 自动化与电气工程学院，甘肃 兰州 730070

摘　要：为了提高短期光伏发电预测的精度，本文提出了一种将聚类后的自组织映射网络（SOM）与优化的核极值学习机（KELM）方法相结合的混合预测模型。首先，利用SOM来对训练数据集进行初始划分。然后，利用模糊C均值（FCM）对训练好的SOM网络进行聚类操作，同时利用Davies-Bouldin指数（DBI）来确定最佳聚类的大小。最后，在每个数据分区中，通过结合差分演化算法优化的KELM方法来建立区域KELM模型，或者结合最小二乘估计的多元线性回归（MR）方法来构建区域MR模型。此外，本文还提出了基于SOM的不同局部多元回归模型。将提出的结合SOM-FCM和KELM的混合预测模型分别应用于GEFCom2014三个不同太阳能电站，进行提前一小时的发电功率预测。与其他预测模型相比，光伏发电站1的平均绝对误差（MAE）降低了61.41%，光伏发电站2的MAE降低了60.19%，光伏发电站3的MAE降低了58.92%。光伏发电站1的均方根误差（RMSE）降低了52.06%，光伏发电站2的RMSE降低了54.56%，光伏发电站3的RMSE降低了51.43%。实验结果表明，提出的结合SOM-FCM和KELM的方法可显著提高预测准确性。

关键词：光伏发电；功率预测；自组织映射神经网络；区域建模方法；优化的核极限学习机（KELM）方法

引用格式：LIU Qibo, LI Jun. Short-term PV power forecasting based on combined SOM-FCM and KELM method. Journal of Measurement Science and Instrumentation, 2024, 15(2): 204-215.