applied sciences

MDPI

Article

Adaptive Solar Power Forecasting based on Machine Learning Methods

Yu Wang, Hualei Zou *, Xin Chen, Fanghua Zhang and Jie Chen

Department of Electrical Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing 210016, China; yuwang15@nuaa.edu.cn (Y.W.); chen.xin@nuaa.edu.cn (X.C.); zhangfh@nuaa.edu.cn (F.Z.); chen_jie@nuaa.edu.cn (J.C.)

* Correspondence: zouhualei@nuaa.edu.cn; Tel.: +86-183-051-59650

Received: 19 September 2018; Accepted: 7 November 2018; Published: 12 November 2018

Abstract: Due to the existence of predicting errors in the power systems, such as solar power, wind power and load demand, the economic performance of power systems can be weakened accordingly. In this paper, we propose an adaptive solar power forecasting (ASPF) method for precise solar power forecasting, which captures the characteristics of forecasting errors and revises the predictions accordingly by combining data clustering, variable selection, and neural network. The proposed ASPF is thus quite general, and does not require any specific original forecasting method. We first propose the framework of ASPF, featuring the data identification and data updating. We then present the applied improved k-*means* clustering, the least angular regression algorithm, and BPNN, followed by the realization of ASPF, which is shown to improve as more data collected. Simulation results show the effectiveness of the proposed ASPF based on the trace-driven data.

Keywords: adaptive solar power forecasting; machine learning; *k-means*; BPNN; LARS

1. Introduction

In recent years, as the growing electrical power load demand and the claim on reducing greenhouse gases generated by the exhaustion of fossil fuels, Smart Grid (SG) and Microgrid (MG) featuring renewable energy (RE) have developed quickly, to satisfy consumption and mitigate pollution [1,2]. However, renewable energy is typically intermittent and weather-dependent, such as in the case of wind and solar power, which is a big challenge to the planning, operation, unit commitment and energy scheduling of MG or SG [3–5]. If the RE power generation can be predicted accurately, the efficiency of energy management can be greatly improved [6]. Thus, many researchers focus on predicting the RE power generation [7,8]. For solar power prediction, statistical methods and machine learning methods are two commonly used approaches [3,8–11]. Statistical methods are basically applied through regression function or probability distribution function (PDF), which estimates the relationship between weather variables and solar intensity [9,12,13]. Machine learning techniques learn the characteristics of the sample data, and then achieves the prediction by training models, such as Neural Networks (NN) and Support Vector Machines (SVM) [10,11].

For statistical methods, the authors of [12] generate a set of solar scenarios by assuming the forecasting errors follow a normal distribution. In [13], the forecasting error of the renewable energy is simulated via multiple scenarios from the Monte Carlo method under a Beta PDF. This method has an obvious dependence on the predictive error distribution function which is normally difficult to obtain. On the other hand, for machine learning, auto-regressor is adopted to predict the two-ahead solar power generation in [14]. Three layers NN is applied to build the forecasting model, the data of the previous day, highest temperature and weather type of the forecasting day are taken as input variables in [15], which forecasts very well for sunny days. Besides the highest temperature, the lowest temperature and average temperature are chosen as the input data to predict photovoltaic (PV) output power in [16]. However, the generalization ability of NN is limited, and thus the overall prediction effect by using only NN is not precise enough in many cases.

Although the forecasting results by any one of the above methods are not satisfactory overall, the forecasting accuracy can be further improved by merging them together. Some existing works focus on the combination of artificial neural network and data mining for more accurate solar power [17]. K-means clustering with nonlinear auto-regressive neural networks are adopted to forecast solar irradiance in [18], and *k-means* with artificial neural networks are used to predict solar irradiance in [19]. However, the traditional *k-means* is sensitive to the initial center, in practice, if the cluster center is given randomly, the *k-means* often converges to one or more clusters which contains very few or even no data point. Therefore, many researchers focus on the cluster centers initialization to improve the performance of the *k-means* [20,21]. The weather data is of high-dimension, which makes the improved *k-means* clustering in [21] a good fit to our method. Because it is proposed to tackle the high-dimension data initial cluster centers. Appropriate classification can improve the prediction accuracy, however, key factors that affect the prediction accuracy can be very different in different clusters. It is thus very necessary to analyze the most important factors in each cluster. The widely used variable selection methods are the forward selection regression, forward stage-wise regression and LASSO [22,23]. In practice, after clustering, there may be very little data in some groups, so the above mentioned methods tend to be too aggressive to eliminate too many correlated and useful predictors. However, the least angular regression (LARS) [24], which is particularly suitable for situations where the feature dimension is higher than the sample number, and during the calculation it keeps all variables selected into the regression model, and provides a simple structure for computing. We therefore choose LARS to select the most important factors in each group. In fact, the predicting methods with only key factors are not sufficient to have very accurate results [22], because the environment is highly complicated.

Therefore, an adaptive revising method on the original forecasting is highly desirable. Motivated by this, we propose an adaptive solar power forecasting (ASPF) method in this paper, which can revise the original forecastings adaptively according to the historical forecasting errors and the weather data. It combines data clustering, variable selection and artificial neural network (ANN) to achieve highly accurate revision on PV power predictions by learning the weather data and day-ahead forecasting errors intelligently and adaptively. The function of the ASPF is shown in Figure 1. Through ANN, ASPF learns the characteristics of forecasting errors between the predicted and actual PV power. By clustering the data, the capture of the forecasting characteristics could be highly improved. In addition, ASPF classifies the predicted day into similar day or non-similar (defined in Section 2), based on which the original forecastings are revised accordingly. The details are shown in Section 2. Also, ANN is adaptively upgraded so that ASPF improves as more data collected. Please note that ASPF actually captures the characteristics of forecasting errors and revises the predictions accordingly, and thus it does not require any specific original forecasting method, which can be a machine learning method or a statistical one. This makes ASPF a quite general method which can be used in many cases.

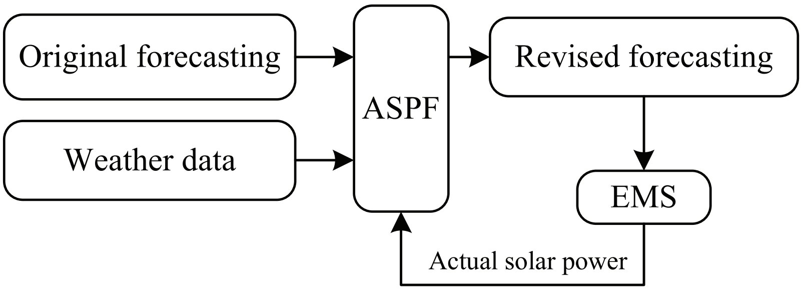


Figure 1. The function of ASPF.

The remainder of this paper is organized as follows. We present the principle of ASPF and its framework in Section 2. The algorithms to realize ASPF are proposed in Section 3. The adaptive solar power forecasting is presented in Section 4. We perform the simulation studies in Section 5. Section 6 concludes this paper.

2. Principle and Framework of the ASPF

In this paper, the realization of ASPF mainly depends on the stored historical solar power data and the learning network which is updated according to the actual needs. In the database, we store the original predictive solar power P_{ori}, the revised solar power P_{rev}, and the actual solar power data P_{act}. As more data collected, ASPF becomes more adaptive. Under certain condition, the historical solar power data P_{sdm} are taken as the revised data P_{rev} directly, which is named as *Similar Day Mode (SDM)*. On the other hand, when the condition of SDM does not hold, which is named as *Non-similar Day Mode (NDM)*, ASPF modifies the original predictions P_{ori} by the output from a learning and updating network obtained from Back-propagation neural networks (BPNN) [25]. It learns the characteristics of differences between P_{ori} and P_{act}. In this way, ASPF predicts more precisely for both SDM and NDM.

Furthermore, whatever the system is working in SDM or NDM, the network can be updated according to that whether the error between P_{rev} and P_{act} is within an acceptable range or not. If the error is out of the allowed range, the network in ASPF is updated. We now present the mode judging standard, ASPF framework, and updating rules in this section. The notations used in this paper are summarized in Table 1.

Table 1. Notation.

Symbol	Description	Symbol	Description
P_{ori}	the original predictive solar power	C_k^{giv}	initial given clustering center in group k
P_{rev}	the revised solar power	C_k^*	optimal clustering center in group k
P_{act}	the actual solar power	$D_n^{(k)}$	density function value of the nth data in order to calculate the initial center of $k+1$th cluster
P_{sdm}	the historical solar power	D_k^*	the kth group initial cluster center
Δ_1	the threshold to judge the similar day	D_w	inner distance in the same group
$\Delta_{1,up}$	the updated value of Δ_1	D_b	the minimum distance between different groups
Δ_2	acceptable error between P_{rev} and P_{act}	I_k	intra-group and inter-group index
$\Delta_{2,up}$	the updated value of Δ_2	N_k	the number of data in kth cluster
R_s	signal to trigger the network updating	$x_{k,g}$	the gth data in group k
x	the vector contains P_{ori} and weather data	c_{or}	correlation coefficient of residual
N	the total number of x	$\hat{o}_n$	the predictive value of the nth data
k	clustering number	w	the weight of neural network
K	the maximum number of clusterings	b	the bias value of neural network
C_k	the kth clustering center	h	the number of hidden neurons
NT	total number of similar days or non-similar day	NL	the number of days with the MxAE less than Δ_2
NB	the number of days with the MxAE larger than Δ_2	SI	the indicator to record whether there is a similar day
TS	the number of similar days when $SI \geq 1$	WD	the indicator to record whether MxAE in P_{sdm}/P_{rev} and $P_{act} \leq \Delta_2$
BD	the indicator to record whether MxAE in P_{sdm}/P_{rev} and $P_{act} \geq \Delta_2$	WD'	the indicator to record WD state after NN updated
BD'	the indicator to record BD state after NN updated	NS	the indicator without similar day

2.1. Mode Judgment

In order to quickly determine the similar days, *K-means* clustering [26,27] is used to divide the data into several groups firstly, and in every group, clustering center is adopted to judge whether there is a similar day in the current database rapidly, as shown in Figure 2. The input vector $x \in R^{q\times1}$ contains P_{ori} and key variables which affect the original predicting accuracy. x is then compared with clustering centers C_k to be clustered into group k, if it has one or more similar days, and the closest historical solar power P_{sdm} can be found in the database. Here, a variable denoted as Δ_1 is chosen as a threshold for the similar historical data, and if the error between x and C_k is larger than Δ_1, there is no similar day.

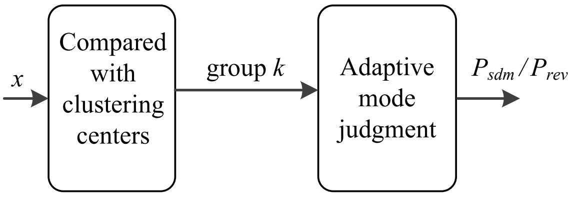


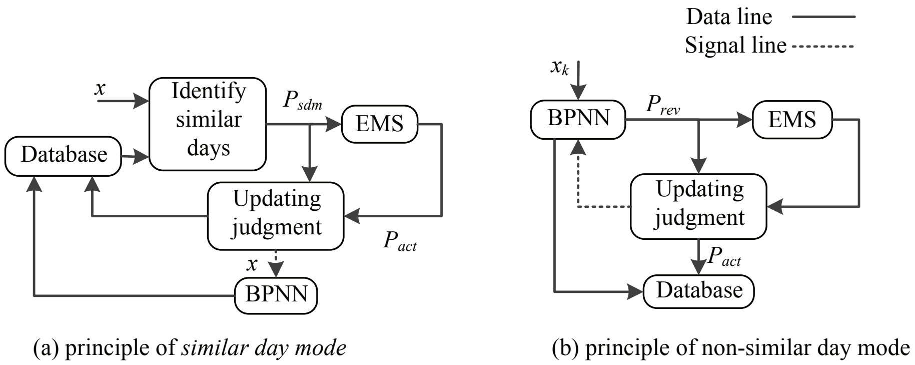
Figure 2. The principle of the *Identify similar days*.

Considering the importance of clustering, we propose an improved *k-means* method to group the data, which is shown in Section 3.1. Please note that in different groups, the key weather variables may be different. For example, solar power is more closely related to temperature in sunny days than cloudy days. Therefore, we adopt a model-selection method called the least angular regression, also known as LARS [24], to analyze and find out the most correlated weather variables in different clusters.

The predicting network is thus acquired by combining the improved *k-means* clustering and LARS, which keeps updating for more accurate predictions. Besides, the trained BPNN, the weight w and the bias value b are used to determine the value of Δ_1, which is thus closed related to the updated network. As the accuracy of the updated network increases, the similarity judgment is getting more precisely as well. We present the thorough steps to determine and update Δ_1 in Sections 3 and 4.

2.2. Framework of the ASPF in SDM and NDM

In order to show the principle of ASPF more clearly, we explain it from the following two aspects, namely, the operation of ASPF with similar days in SDM, and without similar day in NDM. The block diagrams are shown in Figure 3a,b, respectively.



(a) principle of *similar day mode* (b) principle of non-similar day mode

Figure 3. The ASPF schematic in different mode.

Figure 3a shows the case when ASPF is working in SDM. x is compared with every data in *group k* until the closest data is found, which will be used as the revised PV power. Later, the actual solar power P_{act} will be recorded in *EMS* when the whole day of operation ends. In order to testify the effectiveness of ASPF in SDM, P_{sdm} and P_{act} are compared in *Updating judgment*. If the largest difference between these two vectors at any time slot is less than a certain value, P_{sdm} is chosen as P_{rev}, otherwise the network will be updated.

Figure 3b shows the case when ASPF is working in NDM. Because no similar day is found in the current database, ASPF learns the new data through BPNN and decides if the BPNN is updated or not. Different from SDM, P_{rev} is acquired by the BPNN, and the difference between P_{rev} and P_{act} determines whether the network is updated in NDM. In this way, ASPF absorbs more useful data and thus becomes better in predictions.

2.3. Rules for Network Updating and Feedback

Figure 4 shows the rule of the *Updating judgment* function in Figure 3. P_{rev} is the day-ahead revised predictive PV power obtained from SDM or NDM, which is compared with P_{act}, and Δ is the maximum absolute error between P_{rev} and P_{act}. Δ_2 represents acceptable error threshold between the revised solar power and the actual PV power in the current adaptive system. In addition, Δ_2 is updating as more data collected and processed, and will become smaller and smaller. It is also an indicator to measure the accuracy of predictions.

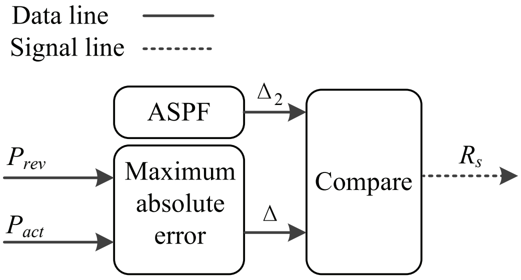


Figure 4. The principle of *Updating judgment*.

In the *Updating judgment*, if Δ is larger then Δ_2, a trigger signal $R_s \in [0,1]$ is produced to activate BPNN. $R_s = 1$ and BPNN is updated; otherwise, BPNN remains unchanged. Thus, the *Updating judgment* is mainly applied to renew the BPNN to guarantee accurate predictions. Here, Δ_2 is calculated by function Equations (9) and (17) in Sections 3 and 4. We apply this both for SDM and NDM.

Through the above three parts, we compose the ASPF, and the complete realizations in every detail are presented next.

3. The Algorithms to Realize ASPF

This section presents the algorithms to realize ASPF. We first propose an improved *k-means* clustering method to divide the data into several groups. Then, LARS is presented to find the most relevant variables in every group. Followed by this, we discuss the compensation network for each group. We also present the feedback and updating mechanism.

3.1. Improved k-means Clustering

Figure 5 shows a typical daily solar power profile in different weather conditions, i.e., sunny, cloudy and rainy. *sunny* 1 and *sunny* 2 represent the sunny day in winter and summer, respectively. *cloudy* and *rainy day*'s solar power are different from the sunny days'. Therefore, considering the differences between these different weather types, we first cluster the data into several specific groups.

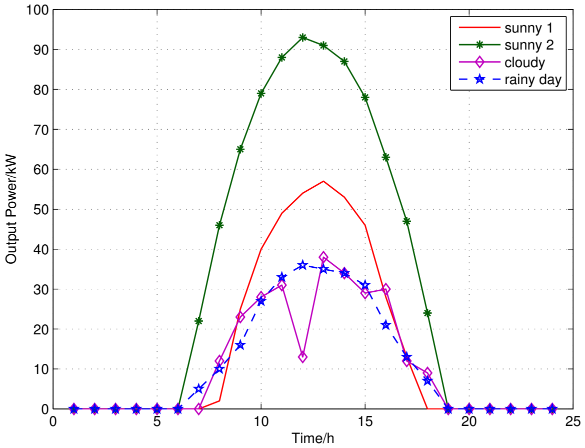


Figure 5. The solar power for different weather types.

k-means is an effective, robust and widely used clustering technique [28]. It aims to cluster the data $\mathbb{N} = [1, 2, \cdots, N]$ observations into K clusters, in which each observation belongs to the cluster with the nearest mean, and different clusters have low similarity with each other. It is implicitly assumed that features within a cluster are *"homogeneous"* and that different clusters are *"heterogeneous"* [26,29]. Here, we adopt *k-means* as a basis for clustering our data.

Given a dataset $\mathbb{S} = \{x_n\}_{n=1}^{N}$ of N points and the number K of desired clusters, for each group, the corresponding generalized centroids are usually denoted as $\mathbb{C} = \{C_1, C_2, \cdots, C_K\}$, and $C_k \in R^{q\times 1}$. The standard *k-means* finds the optimal cluster centers $\{C_1^*, C_2^*, \cdots, C_K^*\}$ such that the sum of the 2-norm distance between each point x_n, and the nearest cluster center C_k is minimized [20,21]:

$$\min : \sum_{k=1}^{K} \left(\sum_{n=1}^{N} \eta_{n,k} \parallel x_n - C_k \parallel^2 \right) \tag{1}$$

where $\eta_{n,k} \in [0, 1]$, is the coefficient which represents the degree of how the nth sample belongs to the kth cluster. For simplicity, $\eta_{n,k} = 1/2$ in this paper. C_k is changed from C_k^{giv} to C_K^* during the calculation, and C_k^{giv} denotes the initial given clustering center for the kth group.

As mentioned in Section 1, the traditional *k-means* is sensitive to the initial center C_k^{giv}, and different initializations can lead to different final results. In recirculation iteration calculation, the *k-means* often converges to one or more clusters which contains very few or even no data point if the cluster center is given randomly.

To overcome this problem, we employ a simplified method which combines potential function and density function to get the initial cluster centers [21]. Firstly, for a given K, the density function of any sample x_n is:

$$D_n^{(0)} = \sum_{j=1}^{N} \left(1 + f_{r_d} \parallel x_n - x_j \parallel^2\right)^{-1} \tag{2}$$

where f_{r_d} is the function of effective radius of the neighborhood density, and the more data around sample x_n, the greater $D_n^{(0)}$ is. $f_{r_d} = (\alpha N(N-1))/(\sum_{n=1}^{N} \sum_{j=1}^{N} \parallel x_n - x_j \parallel^2)$, and α is one constant. For a known sample set, the value of f_{r_d} is easy to get. After the density of each data point has been computed, the data point x_1^* with the highest density will be selected as the first cluster center $C_1^{giv} = x_1^*$, and $D_1^* = \max\{D_n^{(0)}, n \in \mathbb{N}\}$ is the density value. Then, the density function in the subsequent cluster becomes:

$$D_n^{(k)} = D_n^{(k-1)} - D_k^* \left(1 + f_{r_d} \parallel x_n - x_k^* \parallel^2\right)^{-1} \tag{3}$$

where $k = 1, 2, \cdots, K-1$, and $D_k^* = \max\{D_n^{(k-1)}, n \in \mathbb{N}\}$, so the remaining cluster centers $C_k^{giv} = x_k^*$ can be identified as x_k^* according D_k^*. Based on this result, the function Equation (1) will be convergent through few iterations, and no initial clustering will be empty.

k-means consists of two phases: one for determining the initial centroids and the other for assigning data points to the nearest clusters and then recalculating the cluster means. The second phase is necessary for finding the optimal k, and the data within the same type are very similar, while the data from different types are not close.

In this paper, the method discussed in [30] is adopted to obtain optimal number of k. Here, an inner distance in the same group is defined as:

$$D_w(n,k) = \frac{1}{N_k - 1} \sum_{g=1}^{N_k} \| x_{k,g} - x_{k,n} \|^2 \tag{4}$$

where $D_w(n,k)$ is the average distance from nth data in cluster k to other data in the same group, and N_k denotes the number of data in the kth cluster, $n = [1, 2, \cdots, N_k]$. If $D_w(n,k)$ is small, the data in this cluster is also compact correspondingly.

The distance for data between different groups is defined as:

$$D_b(n,k) = \min_{1 \le k \le K, k \ne n} \frac{1}{N_k} \sum_{g=1}^{N_k} \| x_{k,g} - x_{k,n} \|^2 \tag{5}$$

where $D_b(n,k)$ represents the minimal average distance of the nth data from cluster k to other clusters. In addition, the larger value of the $D_b(n,k)$ is, the feature of *"heterogeneous"* is more clear. In order to synthesize the two factors, $D_b(n,k)$ and $D_w(n,k)$, the linear combination is used to balance the relationship between them. At the same time, in order to enable the indicator to analyze the validity of the data clustering and to avoid the index from being affected by the dimension, the fractional operation is employed. The index to calculate the optimal k is defined as:

$$I_k(n,k) = \frac{D_b(n,k) - D_w(n,k)}{D_b(n,k) + D_w(n,k)} \tag{6}$$

where $I_k(n,k) \in [-1, 1]$. Since the $D_b(n,k)$ requires at least two clusters, k starts from 2 to K. From Equation (6), it can be seen that when the cluster number k is taken from 2 to K, the optimal k is confirmed if the $I_k(n,k)$ gets the maximal value. Usually, $K \le \sqrt{N}$, $K = Int(\sqrt{N})$ [31].

Eventually, based on the condition of getting a suitable initial cluster centers and the optimal number of k, pseudo-code for the improved *k-means* clustering is summarized in Algorithm 1.

Algorithm 1: Improved *k-means* Algorithm

```
Step 1: For each $n \in \mathbb{N}$, $k \in [2, K]$.
Step 2: In each cluster $k$, the initial cluster center $C_k^{giv}$ is obtained by executing Equations (2) and (3), then $C_k$ & $N_k$ can be calculated by Equation (1).
Step 3: Find and record $I_k$ based on Equations (4)–(6). Compare $k$ with $K$, and if $k < K$, $k := k + 1$ and repeat Step 2; otherwise, jump to Step 4.
Step 4: Acquire the maximal $I_k$ and the corresponding number $k^*$, recalculate Equations (2) and (3), then the optimal clustering data is obtained through Equation (1), and the optimal centers are $C_k^*$, weather types are $G_k$, $k = 1, 2, \cdots, k^*$.
```

In this way, solar power data is divided into several groups by the proposed improved *k-means* clustering. According to the standardized-residual method and Lagrange Interpolation Polynomial [32], the abnormal data is removed and the remaining data are normalized.

3.2. The Least Angular Regression Algorithm

A typical regression can be expressed as: $\hat{\boldsymbol{\mu}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$, where $\hat{\boldsymbol{\mu}}$ is the current estimate of the vector of responses $\boldsymbol{y}$ with M predictors, $\boldsymbol{X}$ is one matrix with $N \times M$, N is the number of sample, and then current correlation will be: $\hat{\boldsymbol{c}}_{or} = \boldsymbol{X}'(\boldsymbol{y} - \hat{\boldsymbol{\mu}})$, and there will exist a j such that $|\hat{c}_{or,j}|$ is maximized. Then μ will be updated by rule of $\hat{\mu}_{m+1} = \hat{\mu}_m + \hat{\gamma}\text{sign}(\hat{c}_{or,j})X_j$, in LARS, $\hat{\gamma}$ is endogenously chosen so that the algorithm proceeds equiangular between the variables in the most correlated set (hence the 'least angle direction') until the next variable is found. Then the next variable joints the active set, and their coefficient are moved together in a way that keeps their correlations tied and decreasing. This process is continued until all the variables in the model, and end at the full least-squares fit [24].

Actually, the LARS algorithm begins at $\hat{\mu}_0 = 0$, supposing $\hat{\boldsymbol{\mu}}$ is current estimate and let $\hat{\boldsymbol{c}}_{or} = \boldsymbol{X}'(\boldsymbol{y} - \hat{\boldsymbol{\mu}})$. Define $\mathcal{A}$ as a subset of the indices corresponding to variable with largest absolute correlations, $\hat{C}_{or} = \max\{|\hat{c}_{or,m}|\}$ and $\mathcal{A} = \{m : |\hat{c}_{or,m}| = \hat{C}_{or,m}\}$. Defining $s_m = \text{sign}(\hat{c}_{or,m})$, $m \in \mathcal{A}$ and active matrix corresponding to $\mathcal{A}$ is $X_{\mathcal{A}} = (s_m X_m)$. Letting $\mathcal{G}_{\mathcal{A}} = X'_{\mathcal{A}} X_{\mathcal{A}}$ and $A_{\mathcal{A}} = (1'_{\mathcal{A}} \mathcal{G}_{\mathcal{A}}^{-1} 1_{\mathcal{A}})^{-\frac{1}{2}}$, $1_{\mathcal{A}}$ is a vector of $1'$s of length equaling $|\mathcal{A}|$, the size of $\mathcal{A}$. Then the equiangular vector can be defined as $\boldsymbol{u}_{\mathcal{A}} = X_{\mathcal{A}} \rightarrow_{\mathcal{A}}$, where $\rightarrow_{\mathcal{A}} = A_{\mathcal{A}} \mathcal{G}_{\mathcal{A}}^{-1} 1_{\mathcal{A}}$, is the unit vector making equal angles, less than 90°, with the columns of $X_{\mathcal{A}}$ which satisfies $X'_{\mathcal{A}} \boldsymbol{u}_{\mathcal{A}} = A_{\mathcal{A}} 1_{\mathcal{A}}$, and $\| \boldsymbol{u}_{\mathcal{A}} \|^2 = 1$, and $\boldsymbol{a} = X' \boldsymbol{u}_{\mathcal{A}}$. Then the $\boldsymbol{\mu}$ will be updated in the next step of the LARS algorithm is:

$$\hat{\boldsymbol{\mu}}_{\mathcal{A}+} = \hat{\boldsymbol{\mu}}_{\mathcal{A}} + \hat{\gamma} \boldsymbol{u}_{\mathcal{A}} \tag{7}$$

where

$$\hat{\gamma} = \min_{m \in \mathcal{A}^c}{}^{+} \left\{ \frac{\hat{C}_{or} - \hat{c}_{or,m}}{A_{\mathcal{A}} - a_m}, \frac{\hat{C}_{or} + \hat{c}_{or,m}}{A_{\mathcal{A}} + a_m} \right\} \tag{8}$$

$\min^{+}$ indicates that the minimum is taken over only positive components within each choice of m, $a_m \in \boldsymbol{a}$. Finally, dependency variables are sorted by relevance level, and we can easily get the important predictors by LARS.

After the dataset $\mathbb{S}$ is divided in several styles in Section 3.1, and these subsets are defined as $S_1, S_2, \cdots, S_{k^*}$, respectively. Then each subset is analyzed by LARS to reduce the influence of uninformative predictors, and we can focus on variables which is significant for the differences in prediction.

3.3. BPNN-Based Solar Power Compensation

The BPNN, one of the most popular techniques in the filed of neural network, is a kind of supervised learning NN. It uses the steepest gradient descent method to reach very small approximation. Theoretically, BPNN can approximate any nonlinear functions [25], in fact, for the PV power prediction, the relationship between its input data and forecasting value is very complicated [8]. So it is suitable to adopt the BPNN in our ASPF to learn this feature. Therefore, a three-layered feed-forward neural network and the BP learning algorithm are adopted together to compensate the predictive error of solar power in this paper.

The input nodes $\{P_b^k(t)\}_{b=1}^{B}$ of the BPNN to the hidden layer is decided by the result from Section 3.2, x consists of the $\{P_b^k(t)\}_{b=1}^{B}$ from $t = 1$ to T. B denotes the number of the input nodes, and $P_b^k(t)$ is the kth group of inputing predictive message at time slot t. For the number of hidden neurons (HN) in the hidden layer, the method in [33] is used to get the upper bound on the number of hidden neurons, HN_{max}. Then the hidden neurons h is ranged from 1 to HN_{max}, and the model is trained for several times according the method in [34], and the forecasting mean square error from the testing data is calculated and recorded for comparisons. The optimal h^* is acquired in the end, which is the number occuring mostly in the iteration with the less mean square error.

Initializing the weight is another important step in BPNN, and initialization of a network involves assigning initial values for the weights of all connecting links, here, initial values for the weights are set in the small random values between -0.3 and $+0.3$ [33]. Momentum coefficient and learning rate in BPNN are chosen from $[0.021, 0.029]$ and $[0.345, 0.539]$, respectively. Besides, in order to overcome the possible overfitting problem in forecasting, regularization is employed of the parameter ranged in $[0.87, 0.9]$ for k^* groups of models.

Finally, Maximum Absolute Error (MxAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) between the revised solar power and the actual value are used to quantify the performance of each group. MxAE, RMSE and MAPE are widely used statistical measurements of the accuracy of the models [22]. In addition, in each group, MxAE, RMSE and MAPE are defined as follows:

$$\text{MxAE} = \max\ |y_n - \hat{y}_n|,\ n = 1, 2, \cdots, N_k \tag{9}$$

$$\text{RMSE} = \sqrt{\frac{1}{N_k}\sum_{n=1}^{N_k}(y_n - \hat{y}_n)^2} \tag{10}$$

$$\text{MAPE} = \frac{100}{N_k}\sum_{n=1}^{N_k}\left|\frac{y_n - \hat{y}_n}{y_n}\right| \tag{11}$$

where N_k is the number of predicted points in type k, y_n is actual solar power for the nth point, and $\hat{y}_n$ represents the revised solar power.

3.4. Closed-Loop Feedbacks and Updates

ASPF not only has the ability to compensate the error existed in original predictive solar power day-ahead, but also uses the historical solar actual power as the revised value. So, it is necessary to search for a similar day in the system. Besides, it should be feasible to use P_{sdm} as a substitute of P_{rev}, and important to guarantee this P_{sdm} with high accuracy. All above requirements are accomplished by the closed-loop feedback and updating mechanism.

In this paper, we use Δ_2 to define the MxAE of the current forecasting model. In addition, the criteria for judging similar days is related to network parameters and determined by the maximal absolute error of the current BPNN. with the input data of $|x_i - x_j|$, BPNN can get the difference of the predictive PV power and the actual values, if this difference is less than Δ_2, their corresponding actual values will be replaced by each other with acceptable errors. So the actual solar power of the similar data can be used as the revised value directly when this similarity meets certain conditions. Here, the specific relationship of the Δ_1 with the BPNN and this condition are shown below.

The original forecasting models are built in Section 3.3, and after the models are trained, MxAE, weight ω and bias b can be obtained from BPNN. BPNN can be formulated as the non-linear function, expressed as:

$$\hat{y} = f(x^{'}\omega + b) \tag{12}$$

and for any input data $x_i, x_j \in \mathbb{S}$, if it satisfies:

$$\begin{cases} |y_i - \hat{y}_i| \leqslant \Delta_2 \\ |y_j - \hat{y}_j| \leqslant \Delta_2 \end{cases} \tag{13}$$

and the function $f(\cdot)$ has the properties of monotonically increasing and derivable. So $|\hat{y}_i - \hat{y}_j| \leqslant |f(|x_i - x_j|^{'}\omega + b)|$.

In case 1, if $y \geqslant f(x^{'}\omega + b)$, the actual solar power will be larger than the revised one. Thus,

$$(y_i - y_j) \leqslant (f(x_i^{'}\omega + b) - f(x_j^{'}\omega + b)) \tag{14}$$

so,

$$|y_i - y_j| \leqslant (f(|x_i - x_j|^{'}\omega + b)) \leqslant \Delta_2 \tag{15}$$

In case 2, if $y \leqslant f(x^{'}\omega + b)$, the relationship of Equation (15) also can be obtained.

Therefore, with the difference obtained by input data $|x_i - x_j|$, $f(\cdot)$ is known, if the result of $f(\cdot)$ is less then Δ_2, y_i and y_j will be replaced by each other with acceptable errors. On the other hand, when the prediction model is invariant, by comparing the similarity of predicted input information, the actual solar power of similar historical day can be used as the revised PV power directly if it satisfies certain condition. In addition, we give this condition below.

In practice, through comparing the maximal $|x_i - x_j|$ and the corresponding value calculated by the $f(\cdot)$ is known, it can be found that if $\Delta_1 \leqslant \max|x_i - x_j|$, then $|y_i - y_j| \leqslant \Delta_2$. So the initial Δ_1 can be chosen as $\max|x_i - x_j|$, and thus Δ_1 has a close relationship with the updated network. As the accuracy of the updating network increases, the precision of the similar days' solar power prediction becomes higher and higher. So it is reasonable to adopt historical actual solar power as the day-ahead revised data when the similar input message exists. Our proposed method is available for reusing the historical data to achieve better forecasting revision.

4. Adaptive Solar Power Forecasting Algorithm

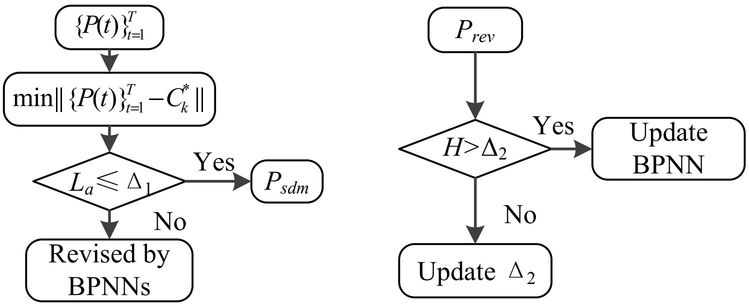
Based on the Algorithms in Section 3, in this section, we present the complete ASPF.

4.1. Effectiveness of the ASPF

As more data collected, there will be more and more historical data in database, so it is not proper to use an invariant parameter as Δ_1 to search for the similar day. Meanwhile, the potential function in [21] has a good performance of slow decaying. Based on this, the decaying of Δ_1 is defined as:

$$\Delta_{1,up} = \Delta_1 e^{-\frac{2}{\pi}(1-\frac{L_a}{L_{ave}})\arctan(NL)} \tag{16}$$

where Δ_1 is current value, $\Delta_{1,up}$ represents the updated value of Δ_1 in the presence of multiple similar days, $L_{ave} = \frac{1}{NL}\sum_{l=1}^{NL}\mathbb{L}(l)$, and $\mathbb{L} = \{L(n) \leqslant \Delta_1\}$, $L_a = \min\{\mathbb{L}\}$. In Equation (16), Δ_1 is updated only when more than two similar days exist. The flow chart in SDM is shown in Figure 6a.



(a) flowchart in *similar day mode* (b) flowchart in *non-similar day mode*

Figure 6. The flow chart for searching similar days.

Similarly, the forecasting error gradually decreases with the increasing amount of the testing data. In addition to this, in the system, the networks can be re-trained when the predictive error beyond the Δ_2. Meanwhile, if MxAE of the testing day $H(d) = \max|\hat{y}_d(t) - y_d(t)|, t = 1, 2, \cdots, T$ is less than the initial Δ_2, it is then to decrease Δ_2, obtained from Equation (9), where d represents the index of the day. The decreasing of Δ_2 follows

$$\Delta_{2,up} = \Delta_2 e^{-(1-\frac{H(d)}{\Delta_2})\frac{H(d)}{\Delta_2}} \tag{17}$$

where $\Delta_{2,up}$ is the updated value of Δ_2. The flow chart of updating BPNN and Δ_2 is shown in Figure 6b.

Figure 7 shows the complete flow chart of ASPF, in which the forecasting accuracy can be improved gradually, and the historical data is thus used more effectively. Moreover, according to the similar daily information, ASPF adaptively identifies and compensates for the predicted solar power day-ahead.

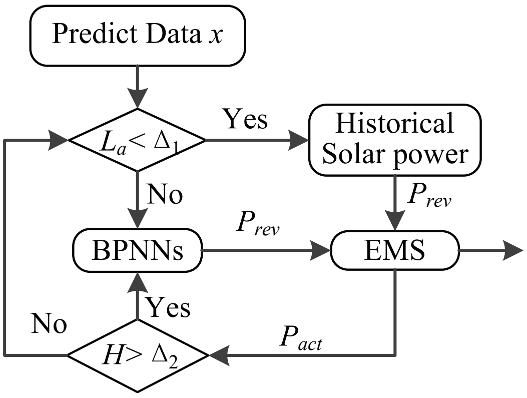


Figure 7. The flow chart for ASPF in power systems.

4.2. The Algorithm of whole System

Based on the above analysis, we now summarize the complete steps of the algorithm to realize the ASPF in Algorithm 2.

Algorithm 2: Adaptive Solar Power Forecasting Algorithm

Step 1: The historical data is divided into different groups by Algorithm 1, and corresponding clustering centers C_k^* are confirmed.
Step 2: LARS is used to each group to get its important predictors, and BPNN is employed to learn the features in every group, subsequently, according the principle in Section 3.4, Δ_1 and Δ_2 is setup.
Step 3: For a new P_{ori} and its important predictors, compares with C_k^*, if the maximal error is less than Δ_1, go to Step 4; otherwise, go to Step 5.
Step 4: P_{ori} and its important predictors compares with every data in *group k*, and finds the most similar day as P_{rev}, and update Δ_1 according formulation Equation (16), then go to Step 6.
Step 5: P_{ori} is revised by BPNN, and the day-ahead PV power P_{rev} is obtained, go to Step 6.
Step 6: $\Delta = |P_{act} - P_{rev}|$, if any element in Δ is larger than Δ_2, update the BPNN, storage P_{act} and P_{ori} in EMS, then go to Step 3; otherwise, go to Step 7.
Step 7: According to formulation Equation (17) update Δ_2, and go to Step 3.

5. Simulation Results

In this section, we test the performance of ASPF based on the trace obtained from a small power system in China. In order to validate the proposed algorithm, we apply it in the original forecasting PV power which is predicted by radial basis function NN and Multiple Linear Regression (MLR). The algorithms including the improved *k-means*, LARS, BPNN and ASPF are verified via simulations and comparisons.

5.1. Data Description

In this paper, the small power system consists of micro-diesel, photovoltaic array, lithium battery cabinet and load demand. The simulations are performed to verify the operation of the proposed algorithm. The data used in ASPF comes from the actual factory which locates in Hekou town, Nantong city, China($32.49^\circ N, 120.83^\circ E$). For PV power forecasting, the prediction can be performed every 2 h, 1 h, 0.5 h, or 15 min according to users' demand. All the simulations complete in the MATLAB, and in our ASPF the computation time mainly depends on the BPNN which only takes several seconds one time in Matlab. So the results of ASPF can be used in day-ahead energy management, and it can be applied to the real-time energy management too. In this paper, the parameter for the PV power adaptive compensation time interval is set to 1 h based on our system. In this system, solar power is mainly supplied to the canteen and the water pumps in the factory. According to the historical data, we obtain the original forecasting PV power in NN and MLR respectively, and store them in EMS.

5.2. Predicting Compensation of ASPF

The data of original solar power is from 1 January to 31 December 2016 in Hekou town. For the *k-means* algorithm, the initial cluster centers calculated by the potential function need less iterations compared with traditional stochastic way, as shown in Figure 8a. As explained in Section 3.1, the range of k is from 2 to 19. In addition, the initial cluster centers can be found fast and better to determine the optimal number of clusters, and effectively avoid the occurrence of non-clustering situation. Equations (4)–(6) are used to determine the optimal number of clusters. Based on the methods described in Section 3.1, the number of clusters for different k under NN and MLR are shown in Figure 8b, where we can see that the optimal number of clusters is $k^* = 2$ in NN, and $k^* = 3$ in MLR.

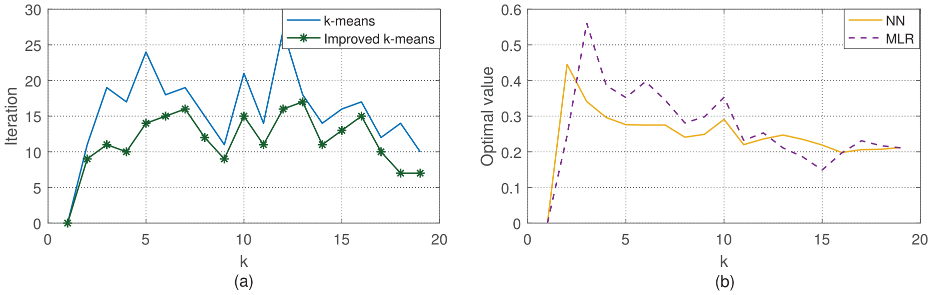


Figure 8. (**a**) The iterations in traditional and improved *k-means*. (**b**) The curve of determining the optimal number of clusters.

In order to modify the differences in original method for forecasting day-ahead solar power, LARS is used to analyze the key factors which affect the forecasting most in NN and MLR. In addition, the critical parameters are different in NN and MLR, shown in Tables 2 and 3. The forecasting results of NN and MLR for different groups are summarized in Tables 4 and 5, which show that our proposed method has greatly improved the accuracy of PV prediction no matter the PV power is predicted by machine learning or MLR.

Table 2. The results of variable selection by LARS in NN.

Pattern	Key Factors		
group 1	Original solar power	Air temperature	
group 2	Original solar power	Air temperature	Sunshine duration

Table 3. The results of variable selection by LARS in MLR.

Pattern	Key Factors			
group 1	original PV power	Sunshine duration	Air temperature	Relative humidity
group 2	original PV power	Relative humidity	Air temperature	Sunshine duration
group 3	original PV power	Air temperature	Sunshine duration	

Table 4. Comparison of forecasting results in NN.

	RMSE (kW)	MAPE (%)	MxAE (kW)
group 1	3.66	4.30	9.476
NN	9.31	16.42	22.34
group 2	1.74	7.04	8.33
NN	6.44	31.91	18.14

Table 5. Comparison of forecasting results in MLR.

	RMSE (kW)	MAPE (%)	MxAE (kW)
group 1	3.02	7.43	10.01
MLR	9.01	51.04	25.51
group 2	3.87	8.39	9.67
MLR	9.59	44.73	20.46
group 3	2.68	6.54	8.54
MLR	11.99	34.66	23.14

5.3. Performance Evaluation of ASPF

We now validate that our ASPF can adaptively revise the solar power which is predicted by NN. We randomly select 100 days of data from 1 January to 31 October 2016, and store the data as the initial database, which includes both predictive information and PV actual power. The data in the database gradually accumulates as the number of days increases in this system. Data from 1 November to 30 December 2016 are used to test the effectiveness of the ASPF. By comparing with the cluster centers C_1^* and C_2^*, $group$ 1 has 23 data points, and $group$ 2 has 37. Here, $d_1 = 1, 2, \cdots, 23$, $d_2 = 1, 2, \cdots, 37$.

For these 60 testing days, the ASPF results are in Table 6. *NL* is the number of days whose MxAE is smaller than Δ_2, and *NB* denotes the number of days whose MxAE is larger than Δ_2. Through Table 6, it can be observed that there are 8 times with similar day, and 15 times with no similar day in $group$ 1. Due to the stored data in $group$ 1 is less than in $group$ 2, there are only 4 times that the historical actual value can be accepted as revised solar power. The non-similar day data are revised by our proposed BPNN, and the results are ideal as their MxAEs are less than Δ_2. Moreover, the similar day number of $group$ 2 is more than that in group 1 because there are more data in $group$ 2 in the database. There is also one non-similar day's result predicted by BPNN over the Δ_2, because the weather changes are more complex in non-sunny circumstances which lead to larger errors. However, the overall performance of our proposed methods is very stable and reliable.

Table 6. The results of ASPF.

$^{.5}$**Pattern**	**Similar Days**			**Non-Similar Day**		
	NT	***NL***	***NB***	***NT***	***NL***	***NB***
group 1	8	4	4	15	15	0
group 2	17	12	5	20	19	1

The comparisons between the proposed method and the historical data are shown in Figures 9 and 10. Here, P_{sdm} denotes the corresponding historical actual solar power if similar days exist. P_{ori} is the predicted solar power by original method. P_{act} represents the actual solar power of the day. Figure 9a shows the result when there is a similar day in $group$ 1 and when $d_1 = 1$, P_{sdm} is used as the revised PV power. In addition, at the night of the day, we can know the actual solar power $P_{act}(d_1)$ by EMS, the MxAE between the P_{sdm} and $P_{act}(d_1)$ will be gotten, here MxAE is 9.112, less than initial Δ_2, which is 9.476. In Figure 9b, there are 3 similar days, and the actual MxAE between P_{sdm} and $P_{act}(13)$ is 8.351, larger than the current Δ_2. In Figure 9d, the data of the 13rd day are used to retrain the BP neural network. When $d_1 = [2,3,4,5,6,8,9,10,11,12]$, there is no similar day, so BPNN is used to get the day-ahead revised solar power as shown in Table 7. And the MxAEs in these days are less than 9.476, so Δ_2 gradually reduces. Δ_2 does not update when BPNN is not working as shown in Figure 9d for the 6th and the 7th day. Figure 9c illustrates the Δ_1 in EMS, and if there is no similar day or the number of similar days is 1, Δ_1 is constant. When the number of similar days is larger than 1, it means several similar days are in the database, and Δ_1 should be lowered in order to find possible similar days as the data increases. In this way, the efficiency of ASPF keeps improving. Please note that, we plot the results for sunny day in Figures 9 and 10, however, our simulation results are not only effective for sunny days, but also for PV power revision in other weather types, such as cloudy, rainy day. It can be seen in Tables 7 and 8, because the data in *group* 1 or *group* 2 can be any weather types.

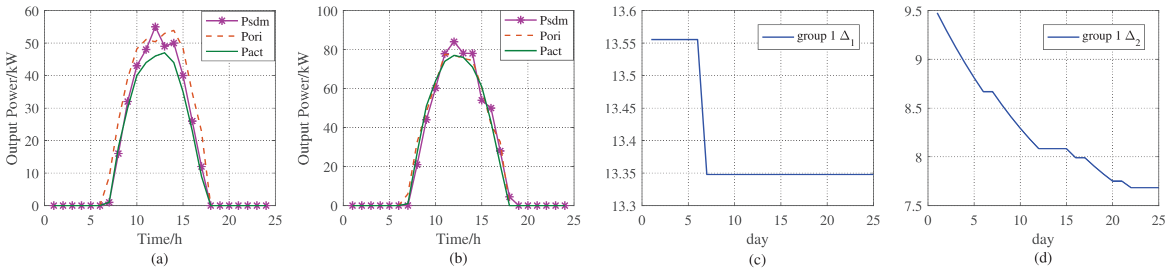


Figure 9. The results of *group* 1 in ASPF. (**a**) denotes that there is a similar day, and the MxAE between the actual solar power and historical value is less than Δ_2; on the contrary, (**b**) shows the results that the MxAE is larger than Δ_2; (**c**) plots a curve that the Δ_1 in *group* 1 changes with the similar days' message; (**d**) shows the Δ_2 in *group* 1 changing with MxAE.

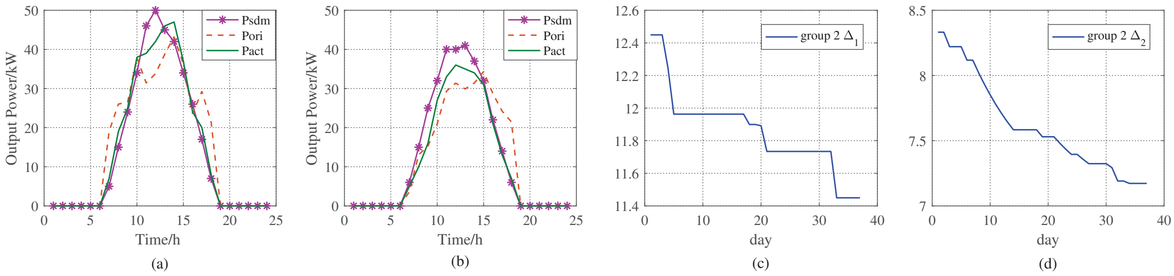


Figure 10. The results of *group* 2 in ASPF. (**a**) denotes that there is a similar day, and the MxAE between the actual solar power and historical value is less than Δ_2; on the contrary, (**b**) shows that the MxAE is larger than Δ_2; (**c**) plots a curve that the Δ_1 in *group* 2 changes with the similar days' message; (**d**) shows the Δ_2 in *group* 2 changing with MxAE.

Table 7. The results of ASPF in *group* 1.

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
SI	1	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	1	1	0	0	1	0	0
TS	1	0	0	0	0	0	3	0	0	0	0	0	1	1	1	0	1	1	0	0	1	0	0
WD	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
BD	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0
NS	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	0	0	1	1	0	1	1
WD'	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	0	0	1	1	0	1	1
BD'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8. The results of ASPF in *group* 2.

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
SI	1	1	0	1	1	0	1	0	0	0	0	0	0	0	1	1	1	1	0	1	1	0	0	0	1	0	0	1	1	1	0	0	1	0	0	1	0
TS	1	2	0	5	5	0	1	0	0	0	0	0	0	0	2	1	1	2	0	2	5	0	0	0	1	0	0	1	1	1	0	0	1	0	0	7	0
WD	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0
BD	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
NS	0	0	1	0	0	1	0	1	1	1	1	1	1	1	0	0	0	0	1	0	0	1	1	1	0	1	1	0	0	0	1	1	0	1	1	0	1
$WD^{'}$	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	0	1	0	0	0	0	1	1	0	1	1	0	1
$BD^{'}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

The results from *group* 2 are shown in Figure 10. The number of similar days increase, the change in Δ_1 and Δ_2 are even more apparent than *group* 1. This change depends on the number of operation days. Tables 7 and 8 record the detailed steps in *group* 1 and *group* 2. Here, SI is an indicator of the existence of similar days: 1 indicates similar days exist, while 0 means no similar day. TS represents the number of similar days when SI $\geqslant$ 1. WD is the MxAE in P_{sdm}/P_{rev}, and P_{act} is less than Δ_2. In contrast, BD indicates that the MxAE is greater than Δ_2. NS is the opposite of SI, and NS = 1 means no similar day. $WD^{'}$ and $BD^{'}$ are used to record the states of our revised prediction performance after updating.

In Table 7, BD is 1 in the 13th day, which means the similar day's actual value has a large deviation from the 13th day's actual solar power. In addition, then, ASPF retrains the network using the 13th day's data at the end of the 13th day. When there is similar day close to the 13th day at the 18th day, the 13th day's actual solar power is adopted as day-ahead revised power and the MxAE is less than Δ_2. The same situation also happens in *group* 2, as shown in Table 8. In *group* 2, at the 3rd day, there is no similar day, solar power is revised by BPNN and then stored in database as shown in Table 8. In addition, the 25th day is similar to the 3rd day's information, so the 3rd day's actual solar power is used as the 25th day's revised power, and the result is acceptable as shown in Figure 10d. This reveals that as the number of data increases, the results of adopting similar day's actual power become more accurate, and the predictive input message can be identified and compensated adaptively.

Through the above simulations, it is reasonable to use the similar day's actual value as the solar forecast power if the forecasting data satisfies the certain condition, which limits the deviation of the predicted data be less than Δ_1. Also, because BPNN updates the network when MxAE exceeds Δ_2, this also ensures that BPNN can effectively predict the data in different situations. In the case of 98%, BPNN can get satisfactory results in this system.

Furthermore, in order to verify the wide applicability of our proposed ASPF algorithm, we run similar simulation for the PV power predicted by MLR, and compare the result. Firstly, 150 days data from 1 January 2016 to 31 August 2016 are randomly selected as the initial database to get the MLR predictive PV power. Then, we select 120 days from 1 September, to 31 December 2016 as testing data. According to the results of *k-means*, the data of the testing days is classified into three groups, which contain 43, 51, and 26 days, respectively. In addition, the ASPF simulation results are shown in Figure 11. In Figure 11a, Δ_1 gradually decreases as the number of days increases, because the data in *group* 1 has more accurate predictions than *group* 2 from Table 5. As Δ_2 decreases in Figure 11b, the network updates continuously, and the number of similar days increases so that Δ_1 becomes smaller correspondingly. For Figure 11c,e, Δ_1 does not change, because in *group* 2 and *group* 3, the number of the similar days are less than in *group* 1. Δ_2 in Figure 11d,f decrease gradually, and thus the accuracy of our method improves continuously. In summary, the proposed ASPF obtains high-precision prediction values in both *non-similar days* and *similar day mode* under different predicting methods.

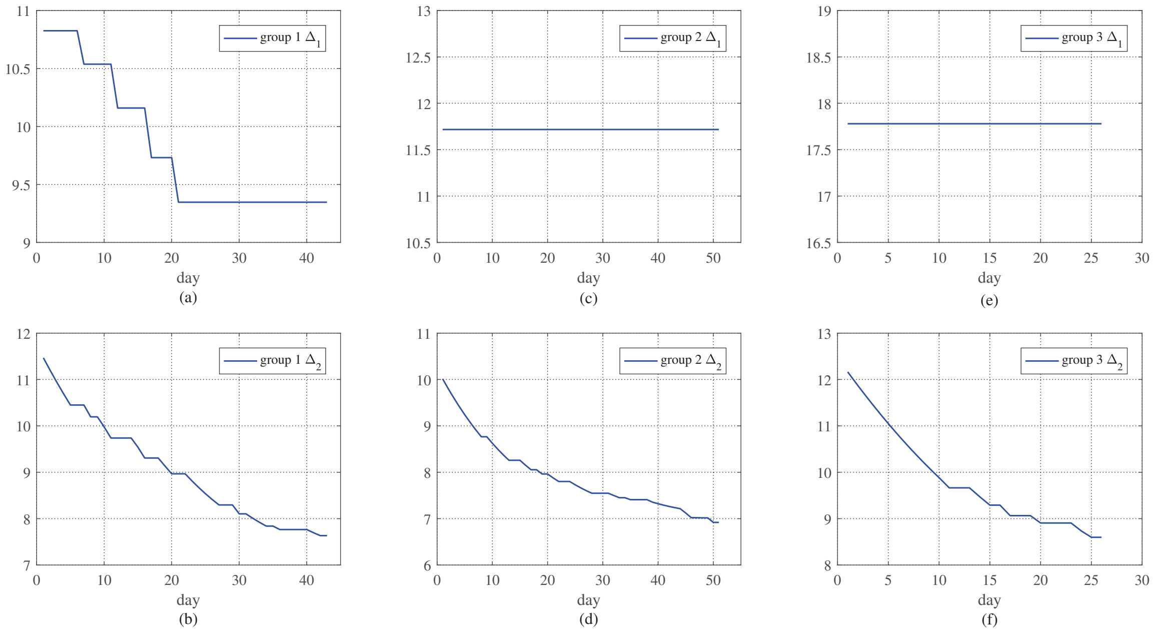


Figure 11. The results of ASPF applied to compensate the solar PV predicted by MLR. (**a**,**c**,**e**) are the trend of Δ_1 in *group* 1, *group* 2 and *group* 3. (**b**,**d**,**f**) denote the Δ_2 in *group* 1, *group* 2 and *group* 3 respectively.

6. Conclusions

In this paper, we propose an adaptive solar power forecasting algorithm for highly precise solar power forecasting. We firstly present the adaptive framework of the algorithm, integrating the improved *k-means* clustering, LARS and BPNN. ASPF is thus able to select the important variables, compensate the predictions from different methods, and update the BPNN adaptively. We also present these algorithms in details. Finally, we evaluate the proposed ASPF through the simulated experiments. We verify the validity of these algorithms, such as, improved *k-means*, LARS, BPNN and ASPF. Furthermore, we testify the effectiveness of the ASPF algorithm for different prediction models. The simulation results show that our proposed method greatly improves the accuracy of solar power, and keeps improving as more data collected. For future work, we consider to apply ASPF in other application, such as energy scheduling mentioned in Introduction.

Author Contributions: Y.W. proposed the conceptualization of adaptive forecasting the solar power; Y.W., H.Z., F.Z. and J.C. worked together to investigate the proper method to realize this adaptive solar power forecasting; H.Z., X.C. and F.Z. validated the performance in Matlab, and the test data is provided by X.C.; H.Z. and Y.W. analyzed the data and wrote the paper; X.C., F.Z. and J.C. supervised the all the entire process.

Funding: The authors gratefully acknowledge financial support for this project provided by the NSF of China under Grants No. 51607087 and the Fundamental Research Funds for the Central Universities of China, NO. XCA17003-06.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Y.; Mao, S.; Nelms, R.M. *Online Algorithms for Optimal Energy Distribution in Microgrids*; Springer International Publishing: Berline, Germany, 2015.
2. Wang, Y.; Mao, S.; Nelms, R.M. On Hierarchical Power Scheduling for the Macrogrid and Cooperative Microgrids. *IEEE Trans. Ind. Inform.* **2015**, *11*, 1574–1584. [CrossRef]
3. Liang, H.; Zhuang, W. Stochastic Modeling and Optimization in a Microgrid: A Survey. *Energies* **2014**, *7*, 2027–2050. [CrossRef]

4. Meng, L.; Sanseverino, E.R.; Luna, A.; Dragicevic, T.; Vasquez, J.C.; Guerrero, J.M. Microgrid supervisory controllers and energy management systems: A literature review. *Renew. Sustain. Energy Rev.* **2016**, *60*, 1263–1273. [CrossRef]
5. Wang, Y.; Mao, S.; Nelms, R.M. Online Algorithm for Optimal Real-Time Energy Distribution in the Smart Grid. *IEEE Trans. Emerg. Top. Comput.* **2013**, *1*, 10–21. [CrossRef]
6. Wang, Y.; Shen, Y.; Mao, S.; Cao, G.; Nelms, R.M. Adaptive Learning Hybrid Model for Solar Intensity Forecasting. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1635–1645. [CrossRef]
7. Soman, S.S.; Zareipour, H.; Malik, O.; Mandal, P. A review of wind power and wind speed forecasting methods with different time horizons. In Proceedings of the North American Power Symposium, Arlington, TX, USA, 26–28 September 2010; pp. 1–8.
8. Antonanzas, J.; Osorio, N.; Escobar, R.; Urraca, R.; Martinez-De-Pison, F.J.; Antonanzas-Torres, F. Review of photovoltaic power forecasting. *Solar. Energy* **2016**, *136*, 78–111. [CrossRef]
9. Bacher, P.; Madsen, H.; Nielsen, H.A. Online short-term solar power forecasting. *Solar. Energy* **2009**, *83*, 1772–1783. [CrossRef]
10. Liu, J.; Fang, W.; Zhang, X.; Yang, C. An Improved Photovoltaic Power Forecasting Model with the Assistance of Aerosol Index Data. *IEEE Trans. Sustain. Energy* **2015**, *6*, 434–442. [CrossRef]
11. Shi, J.; Lee, W.J.; Liu, Y.; Yang, Y.; Wang, P. Forecasting Power Output of Photovoltaic Systems Based on Weather Classification and Support Vector Machines. *IEEE Trans. Ind. Appl.* **2015**, *48*, 1064–1069. [CrossRef]
12. Su, W.; Wang, J.; Roh, J. Stochastic Energy Scheduling in Microgrids with Intermittent Renewable Energy Resources. *IEEE Trans. Smart Grid* **2014**, *5*, 1876–1883. [CrossRef]
13. Osório, G.J.; Lujano-Rojas, J.M.; Matias, J.C.O.; Catalão, J.P.S. Including forecasting error of renewable generation on the optimal load dispatch. In Proceedings of the 2015 IEEE Eindhoven PowerTech, Eindhoven, The Netherlands, 29 June–2 July 2015; pp. 1–6.
14. Palma-Behnke, R.; Benavides, C.; Lanas, F.; Severino, B.; Reyes, L.; Llanos, J.; Sáez, D. A Microgrid Energy Management System Based on the Rolling Horizon Strategy. *IEEE Trans. Smart Grid* **2013**, *4*, 996–1006. [CrossRef]
15. Chen, C.; Duan, S.; Cai, T.; Liu, B. Smart energy management system for optimal microgrid economic operation. *Renew. Power Gener. Iet* **2011**, *5*, 258–267. [CrossRef]
16. Ding, M.; Wang, L.; Bi, R. An ANN-based Approach for Forecasting the Power Output of Photovoltaic System. *Proc. Environ. Sci.* **2011**, *11*, 1308–1315. [CrossRef]
17. Yesilbudak, M.; Çolak, M.; Bayindir, R. A review of data mining and solar power prediction. In Proceedings of the IEEE International Conference on Renewable Energy Research and Applications, Birmingham, UK, 20–23 November 2016; pp. 1117–1121.
18. Benmouiza, K.; Cheknane, A. Forecasting hourly global solar radiation using hybrid *k*-means and nonlinear autoregressive neural network models. *Energy Convers. Manag.* **2013**, *75*, 561–569. [CrossRef]
19. Mccandless, T.C.; Haupt, S.E.; Young, G.S. A regime-dependent artificial neural network technique for short-range solar irradiance forecasting. *Renew. Energy* **2016**, *89*, 351–359. [CrossRef]
20. Huang, X.; Ye, Y.; Zhang, H. Extensions of Kmeans-Type Algorithms: A New Clustering Framework by Integrating Intracluster Compactness and Intercluster Separation. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 1433–1446. [CrossRef] [PubMed]
21. Chiu, S.L. Fuzzy model identification based on cluster estimation. *J. Intell. Fuzzy Syst.* **1994**, *2*, 267–278. (In Chinese)
22. Tang, N.; Mao, S.; Wang, Y.; Nelms, R.M. Solar Power Generation Forecasting with a LASSO-Based Approach. *IEEE Int. Things J.* **2018**, *5*, 1090–1099. [CrossRef]
23. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer: Berline, Germany, 2001.
24. Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R. Least angle regression. *Ann. Stat.* **2004**, *32*, 407–451.
25. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 399–421. [CrossRef]
26. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A K-Means Clustering Algorithm. *J. Roy. Stat. Soc.* **1979**, *28*, 100–108. [CrossRef]

27. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An efficient *k-means* clustering algorithm: analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 881–892. [CrossRef]
28. Ratanamahatana, C.A.; Lin, J.; Gunopulos, D.; Keogh, E.; Vlachos, M.; Das, G. Mining Time Series Data. *Data Min. Knowl. Discov. Handb.* **2005**, 1069–1103.
29. Modha, D.S.; Spangler, W.S. Feature Weighting in *k*-Means Clustering. *Mach. Learn.* **2003**, *52*, 217–237. [CrossRef]
30. Zhou, S.B.; Xu, Z.Y.; Tang, X.Q. Method for determining optimal number of clusters in K-means clustering algorithm. *J. Comput. Appl.* **2010**, *30*, 1995–1998. [CrossRef]
31. Rezaee, M.R.; Lelieveldt, B.P.F.; Reiber, J.H.C. A new cluster validity index for the fuzzy c-mean. *Pattern Recognit. Lett.* **1998**, *19*, 237–246. [CrossRef]
32. Liu, L.; Zhai, D.; Jiang, X. Current situation and development of the methods on bad-data detection and identification of power system. *Power Syst. Prot. Control* **2010**, *38*, 143–147.
33. Govindaraju, R.S.; Rao, A.R. Artificial Neural Networks in Hydrology. *J. Hydrol. Eng.* **2000**, *5*, 124–137.
34. Jiao, B.; Ye, M. Determination of Hidden Unit Number in a BP NeuraI Network. *J. Shanghai Dianji Univ.* **2013**, *16*, 113–116, 124. (In Chinese)