Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2021.015922

ARTICLE

Forecasting Model of Photovoltaic Power Based on KPCA-MCS-DCNN

Huizhi Gou[1,2,*] **and Yuncai Ning**[1]

[1] School of Management, China University of Mining and Technology-Beijing, Beijing, 100083, China

[2] China Energy Investment Corporation, Beijing, 100011, China

[*] Corresponding Author: Huizhi Gou. Email: ghz_0262@163.com

Received: 24 January 2021 Accepted: 21 April 2021

ABSTRACT

Accurate photovoltaic (PV) power prediction can effectively help the power sector to make rational energy planning and dispatching decisions, promote PV consumption, make full use of renewable energy and alleviate energy problems. To address this research objective, this paper proposes a prediction model based on kernel principal component analysis (KPCA), modified cuckoo search algorithm (MCS) and deep convolutional neural networks (DCNN). Firstly, KPCA is utilized to reduce the dimension of the feature, which aims to reduce the redundant input vectors. Then using MCS to optimize the parameters of DCNN. Finally, the photovoltaic power forecasting method of KPCA-MCS-DCNN is established. In order to verify the prediction performance of the proposed model, this paper selects a photovoltaic power station in China for example analysis. The results show that the new hybrid KPCA-MCS-DCNN model has higher prediction accuracy and better robustness.

KEYWORDS

Photovoltaic power prediction; kernel principal component analysis; modified cuckoo search algorithm; deep convolutional neural networks

Abbreviations

ARIMA	Autoregressive integrated moving average model
ARMA	Autoregressive moving average model
ARMAX	Autoregressive moving average model with exogenous inputs model
ARX	Autoregressive model with exogenous input
BPNN	Back propagation neural network
CNN	Convolutional neural networks
CS	Cuckoo search algorithm
DCNN	Deep convolutional neural networks
GA	Genetic algorithm
KPCA	Kernel principal component analysis
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MCS	Modified cuckoo search algorithm
PCA	Principal component analysis

PSO	Particle swarm optimization
PV	Photovoltaic
RMSE	Root mean square error
SVM	Support vector machine

1 Introduction

With the world's energy crisis becoming increasingly serious and the environmental pollution caused by the burning of fossil fuels intensifying, the development of clean and renewable energy has become the consensus of the entire human race [1]. Among them, the conversion and utilization of solar energy, with its advantages of less restricted environmental conditions and low equipment maintenance costs, has received widespread attention and been vigorously promoted in many countries. The use of photovoltaic power generation has undoubtedly had a significant alleviating effect on the current dual crisis of energy and environment [2].

Photovoltaic power is affected by real-time weather and diurnal alternation, and is significantly volatile, random and intermittent. Fluctuations in the total amount of PV power on the grid will have a significant impact on the safety and stability of the power system, as well as on the scheduling of the power system, the marketing of power and the bidding of the power generation companies for the grid [3]. Therefore, accurate PV power forecasting is of great value and significance.

In terms of PV power prediction, there have been related studies at home and abroad, in which the prediction methods are mainly divided into two categories: physical methods and statistical methods [4]. The approach of the physical method, which addresses the logical relationship between meteorological data and PV power, investigates the key influencing factors of PV power based on physical principles. With regard to this approach, existing studies often adopt satellite cloud maps to obtain data on the movement, coordinates and size of the clouds in the sky, which leads to predictions of the variability of ground irradiance and ultimately to PV power predictions [5]. The physical method does not require a large amount of historical data and is suitable for new PV plants, but requires detailed geographical information of the PV plant and data such as module parameters. It has a complex modeling process. It is also difficult to simulate some extreme abnormal weather conditions and slow changes in the environment and PV module parameters over time. In summary, the model is poorly resistant to disturbances and not very robust [6].

Statistical methods refer to the employment of historical data to find the intrinsic laws between data changes and data relationships, and thus to accomplish forecasting [7]. For example, the time series methods represented by autoregressive moving average model (ARMA), autoregressive integrated moving average model (ARIMA), autoregressive model with exogenous input (ARX) and autoregressive moving average model with exogenous inputs model (ARMAX) [8] are mainly applicable to the situation where the weather conditions are more stable, while this method does not have high prediction accuracy due to the simpler conditions. Classification regression methods represented by support vector machines (SVM) and decision trees classify PV power into specific classes and situations based on the idea of clustering to filter similar days closer to the target in a smaller range [9]. Neural network algorithms represented by back propagation neural network (BPNN) and wavelet neural network have been widely applied in research in recent years with the rapid development of artificial intelligence [10]. In addition, random forest algorithms exist, i.e., decision trees based on data sampling are built and interlinked to achieve a PV power prediction model capable of filtering noise and fluctuations [11]. Another approach is the probabilistic prediction algorithm, which gives the probability distribution of the PV power taken at a certain time period [12]. From this, it can be found that the statistical model part of the existing research mainly uses shallow learning algorithms such as SVM and BPNN, which can only learn the information contained in the data from lower dimensions and cannot fully reflect the characteristics of the information, so it is necessary to introduce deep learning algorithms into PV power prediction research.

The true computational power of neural networks came into play in 2006 when Professor Hinton of the University of Toronto pioneered the "multi-layer structure, layer-by-layer learning" deep neural network [13]. The pioneering of deep neural networks has attracted a great deal of attention from academia and industry, and with in-depth research and development of deep learning, it has become a popular tool in the field of data analysis in the era of big data [14]. This approach has made breakthroughs in areas such as signal recognition and natural language processing, and continues to set various records within various application areas at an astonishing rate. Deep convolutional neural networks were proposed by Krizhevsky et al. in 2012 by introducing the concept of deep learning on top of the traditional convolutional neural networks (CNN) [15]. This network is widely applied because it can effectively avoid complex data processing and can learn data features autonomously. In fact, DCNN is the first learning algorithm that can successfully train a multilayer network [16]. The DCNN model achieves optimization of the neural network structure by autonomously feeling convolution, weight sharing, subsampling and multiple perceptual layers for local features [17]. The DCNN model not only reduces the number of neurons and weights but also makes the input features displacement, scaling and warping invariant using the pooling operation, thus improving the accuracy and robustness of network training [18].

Although DCNN has shown good prediction performance, it still suffers from the problem of blind parameter selection and therefore needs to be optimized by choosing a suitable intelligent algorithm [19]. The cuckoo search algorithm (CS) is an emerging heuristic algorithm proposed by Professor Xinshe Yang and S. Deb of Cambridge University in 2009 [20]. Compared with particle swarm algorithm and genetic algorithm, this algorithm has a relatively high performance for solving optimization problems. It is inspired by the cuckoo's parasitic hatchling breeding method to solve optimization problems. The algorithm is favored by many research scholars for its simple structure and few control parameters, and has become a current research hotspot [21]. However, in the process of optimization, it was found that the CS algorithm has disadvantages such as the tendency to fall into local optimality [22]. In order to overcome the shortcomings of the CS algorithm and improve the accuracy and efficiency of the optimization, this paper introduces the concept of random weights for improvement, and employs random weights to change the update method of the bird's nest position in order to improve convergence speed of the algorithm and achieve the global optimal search.

Furthermore, there are many factors affecting PV power that are coupled with each other, and if all the influencing factors are used as input indicators for the prediction model, there will be a large amount of redundant data, so feature selection is also important [23]. Principal component analysis (PCA), as a multivariate statistical method that combines multiple variables into a small number of variables [24], is better at dealing with individual indicators that have a strong linear relationship. However, the relationship between the factors affecting PV power is often non-linear. In contrast, principal component analysis is a linear method that cannot obtain higher-order features of the data and ignores the non-linear information of the data while reducing the dimensionality [25]. Therefore, in this paper, kernel principal component analysis is adopted to reduce the dimensionality of the input variables while retaining the non-linear information among the input variables by mapping the initial input variables to a high-dimensional feature space through a non-linear transformation.

To sum up, this paper constructs a PV power prediction model based on a modified CS algorithm to optimize DCNN, and applies KPCA to reduce PV power influencing factors. The remainder of the article is organized as follows. The second part introduces the algorithms used in paper, which include KPCA, modified cuckoo search algorithm and DCNN model. The third section constructs a complete framework for PV power prediction. The accuracy and robustness of the proposed models are tested and analyzed in the fourth section using practical examples. The fifth section summarizes the research results of paper.

2 Basic Theory

2.1 Kernel Principal Component Analysis

PCA, as a multivariate statistical method that combines multiple variables into a few, has a good treatment when dealing with individual indicators that have a strong linear relationship [26]. However, the relationship between the factors affecting PV power is often non-linear. And principal component analysis is a linear method that cannot obtain the higher-order characteristics of the data and ignores the non-linear information of the data while reducing dimensionality. Therefore, this paper utilizes KPCA to extract key factors, which can effectively deal with the non-linear relationship between variables and is widely employed in the comprehensive analysis of multiple indicators, and can achieve more reasonable results than PCA.

Kernel principal component analysis is a non-linear principal component model that maps the initial input variables to a high-dimensional feature space through a non-linear transformation, reducing the dimensionality of the input variables while retaining the non-linear information between them, in order to accurately extract the important information features between the variables to obtain the main input indicators. This methodology can compress information contained in a large number of indicator variables into a small number of composite variable indicators that reflect the original information characteristics, and deal with the non-linear relationships between variables through the analysis of the composite variable indicators, while ensuring that the loss of information in the original data is minimized. The underlying steps are as follows [27].

Set a group of random vectors $X = \{x_1, x_2, \ldots, x_\mathrm{n}\}^T$ containing variables, where $x_k \in R^N (k = 1, 2, \ldots, m)$ and m represents the number of input samples. The original input sample data set is $M = [a_1, a_2, \ldots, a_\mathrm{n}]^T = [b_1, b_2, \ldots, b_m]$. The projection of data set into the space F through the nonlinear mapping Φ is $M = [\varphi(b_1), \varphi(b_2), \ldots, \varphi(b_m)]$, satisfying $\sum_{k=1}^{m} \tilde{\varphi}(b_i) = 0$.

The covariance matrix is obtained as follows:

$$C^F = \begin{bmatrix} C_{11}, C_{12}, \ldots, C_{1\mathrm{n}} \\ \ldots \\ C_{n1}, C_{n2} \ldots, C_{nn} \end{bmatrix} \tag{1}$$

$$C = \frac{1}{m} \sum_{i}^{m} \varphi(b_i) \varphi(b_i)^T \tag{2}$$

The eigenvalue and eigenvector can be obtained as follows:

$$\mathbf{C}^F\mathbf{W}^F = \lambda^F\mathbf{W}^F \tag{3}$$

where λ^F is the eigenvalue and $\mathbf{W}^F = \sum_{k=1}^{M} \alpha_k\tilde{\varphi}(x_k)$ is the eigenvector corresponding to the eigenvalue. Define a symmetric matrix K whose elements are:

$$\mathbf{K}(x_k, x_j) = (\varphi(x_k)\cdot\varphi(x_j)) \tag{4}$$

$\tilde{\mathbf{K}}$ can be obtained by matrix centralization:

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{I}_n\mathbf{K} - \mathbf{K}\mathbf{I}_n + \mathbf{I}_n\mathbf{K}\mathbf{I}_n \tag{5}$$

where $\mathbf{I}_n$ is n×n matrix, $\mathbf{I}_{ij} = 1/n$, and Eq. (3) can be simplified as:

$$m\lambda^F M = \tilde{\mathbf{K}}M \tag{6}$$

As a result, the kernel principal component can be calculated by referring to extraction technique of traditional PCA.

2.2 *Modified Cuckoo Search Algorithm*

Cuckoo search algorithm is a novel heuristic algorithm proposed by Professor Xinshen Yang and S. DEB of Cambridge University in 2009 [28]. The CS algorithm outperforms particle swarm optimization (PSO) and genetic algorithm (GA) in solving optimization problems mainly owing to its fewer parameters, simplicity of operation, ease of implementation, excellent stochastic search paths, and strong merit-seeking capability. Its outperformance is mainly manifested in better convergence and stronger robustness. It is inspired by the breeding method of cuckoo parasitizing and brooding. Currently, cuckoo search algorithm has been widely applied in project scheduling, engineering optimization and other fields. Due to its simple structure and few control parameters, this algorithm is favored by many researchers and has become a research hotspot.

2.2.1 Cuckoo Search Algorithm

The core idea of CS is the breeding pattern of cuckoo and Levy flight [29].

(1) Breeding pattern of cuckoo

According to the long-term observation and research of entomologists, cuckoos do not raise their offspring, but secretly put their nestling birds in the nests of other birds. If they are not found by the owner of the nest, the offspring will be raised. All along, cuckoo does not build nests or hatch young eggs. It will search for birds which have similar shapes and sizes of eggs. These birds also have similar breeding period and similar feeding habits. Cuckoo will watch other birds. When they leave the nest, cuckoo will quickly lay eggs in Orioles, skylarks and other birds' nests, and let them hatch on their own behalf. Because the color and size of the eggs are very similar, cuckoo will remove one egg from its original nest before laying eggs, and at the same time they give birth to one of its own, so that its young can enjoy the care of other birds. The cuckoo's characteristic is to find a nest, parasitize and incubate. CS is a simulation of this behavior. The core idea is to regard the nest selected by cuckoo as the distribution of solution in space, and whether the nest location is good or not symbolizes the fitness value of the solution to the problem. The series of proceedings of the cuckoo searching and deciding on a nest represents the whole process of optimization performed by the algorithm.

(2) Levy flight

CS adopts Levy flight search method, which satisfies random distribution of heavy tail. The walking step length is short distance and long distance, which will show up alternately. Employing Levy flight mode can increase search space, expand population diversity and jump out of local optimum. Several important parameters in Levy distribution are characteristic index α, scale σ, displacement χ and direction parameter β. The definition of Levy distribution is the Fourier transform of its characteristic function.

$$\begin{aligned} p_{\alpha,\beta}(k;\mu,\sigma) &= F\{p_{\alpha,\beta}(x;\mu,\sigma)\} = \int\limits_{-\infty}^{\infty} dxe^{ikx} p_{\alpha,\beta}(x;\mu,\sigma) \\ &= \exp\left[iuk - \sigma^{\alpha}|k|^{\alpha}\left(1 - i\beta\tfrac{k}{|k|}\varpi(k,\alpha)\right)\right] \end{aligned} \tag{7}$$

where $\varpi(k,\alpha) = \begin{cases} \tan\frac{\pi\alpha}{2}, \alpha \neq 1, 0 \prec \alpha \prec 2 \\ -\frac{2}{\pi}\ln|k|, \alpha = 1 \end{cases}$

The probability density function of Levy flight distribution is shown below:

$$p_{\alpha,\beta}(x) = \begin{cases} \dfrac{1}{\sqrt{2\pi}} x^{-\frac{2}{3}} \exp\left(-\dfrac{1}{2x}\right), & x \geq 0 \\ 0, & x \prec 0 \end{cases}, \tag{8}$$

where $\alpha = \frac{1}{2}$, $\beta = 1$.

The jump distribution probability density function of Levy flight is presented in the following formula:

$$\lambda(x) \approx |x|^{-1-\alpha}, 0 \prec \varepsilon \prec 2 \tag{9}$$

Since Levy flight is a function of second-order divergence, its jumping in motion is very large.

2.2.2 Modification of Cuckoo Search Algorithm

CS has been successfully applied to the field of nonlinear optimization, and has obtained good optimization results. It has excellent local and global convergence, and it needs fewer parameters. Consequently, it has a strong robustness and high search efficiency. However, in the process of optimization, it has been found that Cuckoo search algorithm emerges with some drawbacks such as slow convergence speed and long operation time [30]. In order to overcome the shortcomings of CS and improve the accuracy and efficiency of optimization, the concept of random weight is introduced to improve CS. The random weight is adopted to change the update mode of bird's nest position. The ultimate goal of these measures is to improve the convergence speed of the algorithm and realize the global optimal search.

Random weight is a dynamic approach to select weights from Gaussian distribution. This method can avoid CS falling into local optimum in the initial stage of search, and the random appearance of large and small weights can also improve the slow convergence speed and low precision of the algorithm in the later stage.

$$w = 0.5 + \frac{\text{randm}()}{2} \tag{10}$$

where randm() represents a random number and obeys normal distribution. According to the literature [31], the formula is to generate a value randomly between [0.5, 1].

$$w = r_{\min} + (r_{\max} - r_{\min}) * \text{normrnd}() + \sigma * \text{randm}() \tag{11}$$

where $r_{\max}$ and $r_{\min}$ represent the maximum and minimum values of randomly selected weights, respectively. normrnd ( ) represents a random number and obeys uniform distribution. σ represents deviation.

The random weight obeys Gaussian distribution, $w \sim N(\theta, \sigma)$. Due to the random appearance of large and small weights, the algorithm can find the optimal solution more easily and get the balance of local and global search. Using the random weight can change the updating mode of the nest position, hence this paper proposes a cuckoo optimization algorithm based on random weight. The specific flow is as follows:

Step 1: Determine the optimization function $f(x)$, set the objective function dimension d, definition domain $[-l, l]$, discovery probability P_0 (the probability of new nest replacing old nest) and population number n. Initialize the population n and generate d-dimensional vector $(x_1, x_2, \cdots, x_n)^T$ randomly. The initial position of n nests is $S_i(i = 1, 2, \cdots, n)$.

Step 2: Calculate objective function f_i of each nest and record the current optimal solution. The reciprocal of mean absolute percentage error is defined as the objective function.

Step 3: Keep the optimal position of the previous nest S_i^t, generate weights $w \sim N(\theta, \sigma)$ randomly, replace the position of the previous generation with a new position and update it. The formula is as follows:

$$S_i^{(t+1)} = w * S_i^t + \alpha \oplus \text{Levy}(\lambda) \tag{12}$$

where t represents the number of iterations, α is the step size, and $\oplus$ is the point-to-point multiplication. This formula represents a random walking equation. The first term of the current equation and the second term of the transition probability equation determine the position of the next generation. Levy flight is a random search path whose step size satisfies Levy distribution.

Step 4: Compare the updated nest position S_i^{t+1} with the previous position S_i^t. If the current position is better than the previous position, in other words $f_i^{t+1} > f_i^t$, $S_i^{t+1}$is regarded as the current optimal position, otherwise, the previous position S_i^t remains unchanged.

Step 5: Compare the random number of the possibility of the nest owner to discover the eggs of foreign birds compared with the probability of discovery (the parameter value has been determined in the parameter setting). If $r \le P_0$, the current position will be recorded as the optimal position; otherwise, it will be turned to Step 2, and the position of the nest will be changed randomly to get a new set of positions for comparison.

Step 6: Output the global optimal solution which satisfies the above conditions, including $f_i^{t+1} > f_i^t$ and $r \le P_0$.

2.3 Deep Convolutional Neural Network

A deep convolutional neural network is an artificial neural network with deep learning capability, mainly characterized by locally connected and shared weights of neurons in the same layer. Convolutional neural networks are generally composed of multiple feature extraction layers and a fully connected layer at the end, with each feature extraction layer consisting of a convolutional layer and a sub-sampling layer. The neuron nodes between layers in deep convolutional neural networks are no longer in the form of full connectivity, but rather the neuron nodes of each adjacent layer are linked to only the upper layer neuron nodes that are close to it exploiting the layer space correlation. In other words, local connectivity is achieved, which greatly reduces the parameter size of the network [32]. The structure of DCNN is displayed in Fig. 1.

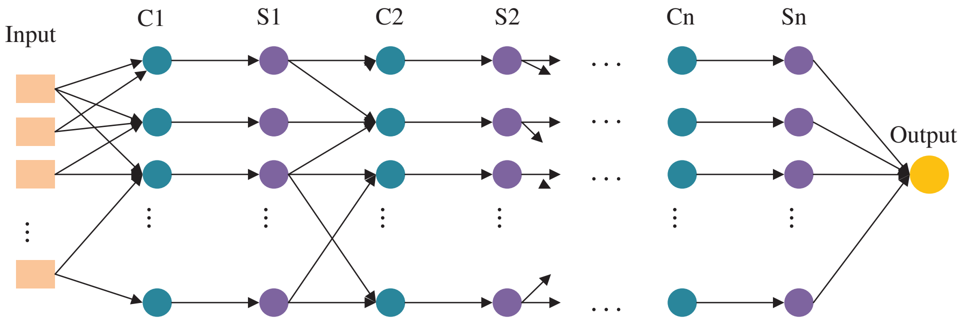

Figure 1: The typical DCNN structure

A classic DCNN structure mainly consists of the input layer, the convolutional layer, the subsampling layer and the fully connected layer. The convolutional layer is mainly utilized for feature extraction through convolutional kernels to achieve feature vector extraction and enhancement [33]. In the convolutional layer, the output features of the previous layer are convolved with the convolutional kernel for a convolutional operation, and the feature output of this convolutional layer is calculated by weighting the activation function, as shown in the following equation:

$$x_j^l = f\left(\sum_{j=m}^{k} x_j^{l-1} w_j^l + \theta_j^l\right), \quad j = 1, 2, \ldots, n; 0 < m \le k \le n \tag{13}$$

where $f(I) = \frac{1}{1+e^{-I}}, I = \sum_{j=m}^{k} x_j^{l-1} w_j^l + \theta_j^l (j = 1, 2, \ldots, n; 0 < m \le k \le n)$, x_j^l is the value of corresponding position on the output dataset of the first layer; x_j^{l-1} is the value of corresponding position on the input dataset of layer $l-1$; j takes the value of m to k to denote that it is a local connection; w_j^l is the weight value of convolution kernel; and θ_j^l is the threshold value.

The subsampling layer is mainly employed to subsample the features of the convolutional layer by performing a "pool averaging" or "pool maximization" operation [34] for each sample pool of $n \times n$ size in the subsampling layer to extract sample features as follows:

$$x_j^l = g(x_j^{l-1}) + \theta_j^l \tag{14}$$

where x_j^l, x_j^{l-1} and θ_j^l represent the same meaning as Eq. (13); g represents the function that selects the average or highest value. By pooling operations, the complexity of convolution layer is effectively reduced, the overfitting phenomenon is avoided, and the fault tolerance of the feature vector to microdeformations of data features is improved, enhancing the computational performance and robustness of the algorithm.

The data obtained after the above convolutional and subsampling layers will eventually be connected to the fully connected layer [35]. The fully connected layer is formulated as follows:

$$x^l = f(I^l), I^l = W^l x^{l-1} + \theta^l \tag{15}$$

where $f(\sim)$ and θ^l represent the same meaning as in Eq. (14), W_l is the weight value when l calculating the layer to $l-1$ layer, and x^l is the output data.

In the aforementioned calculation process, each convolutional kernel is repeatedly applied to all the input data by sliding, and multiple sets of output data are obtained by convolving several different convolutional kernels, with the same convolutional kernel having the same weight, and the output data of different sets are combined and then output to the sub-sampling layer. The sub-sampling layer takes the output data of the previous convolutional layer as input data, first sets the range of value locations, then fills the range with the average or maximum value of the range by sliding, and finally combines these data to obtain the reduced dimensional data and outputs the result through the fully connected layer [36].

The application of DCNN for PV power prediction has two major advantages: firstly, it allows for the existence of malformed data; secondly, it reduces the number of some parameters through local connectivity and weight sharing, which improves the efficiency and accuracy of PV power prediction. However, the number of parameters that need to be trained in the prediction application needs to be determined subjectively, which affects the stability of the prediction results. To further improve the operational efficiency and prediction accuracy, this paper will adopt the MCS algorithm to optimize the parameters of DCNN.

3 Implementation of PV Power Prediction Model

Based on the aforementioned theory, the KPCA-MCS-DCNN prediction model is constructed in this paper. Fig. 2 illustrates the flow of PV power prediction based on KPCA-MCS-DCNN.

As depicted in Fig. 2, the model firstly performs feature dimensionality reduction on the impact factor of PV power by KPCA method, then applies MCS to optimize the parameters of DCNN, and finally employs DCNN to train the sample set to obtain the optimal prediction model. The detailed procedures are as follows:

(1) Select initial input variables (xi) and process data. By analyzing the PV power characteristics, generate the initial set of input variables X={xi,i = 1,2,... n}, and standardize data of each input factor (xi).

(2) Extract features via KPCA. After Step (1), the original input variable matrix X is generated, and the Gaussian kernel function is selected for the non-linear mapping function. After the non-linear transformation of Eqs. (1) to (6), the kernel principal components are extracted according to the criterion that the cumulative variance contribution rate τ is higher than 95%, and the new input variable matrix is finally formed.

(3) Initialize parameters. Initialize the weights w and thresholds θ in DCNN by randomly selecting a number of values from a smaller set of values as the weights and all thresholds for each layer.

(4) Optimize DCNN with MCS. If the conditions are met, the optimal parameters are obtained; if the conditions are not met, the MCS optimization algorithm is executed again until the set of solutions that meet the conditions is obtained.

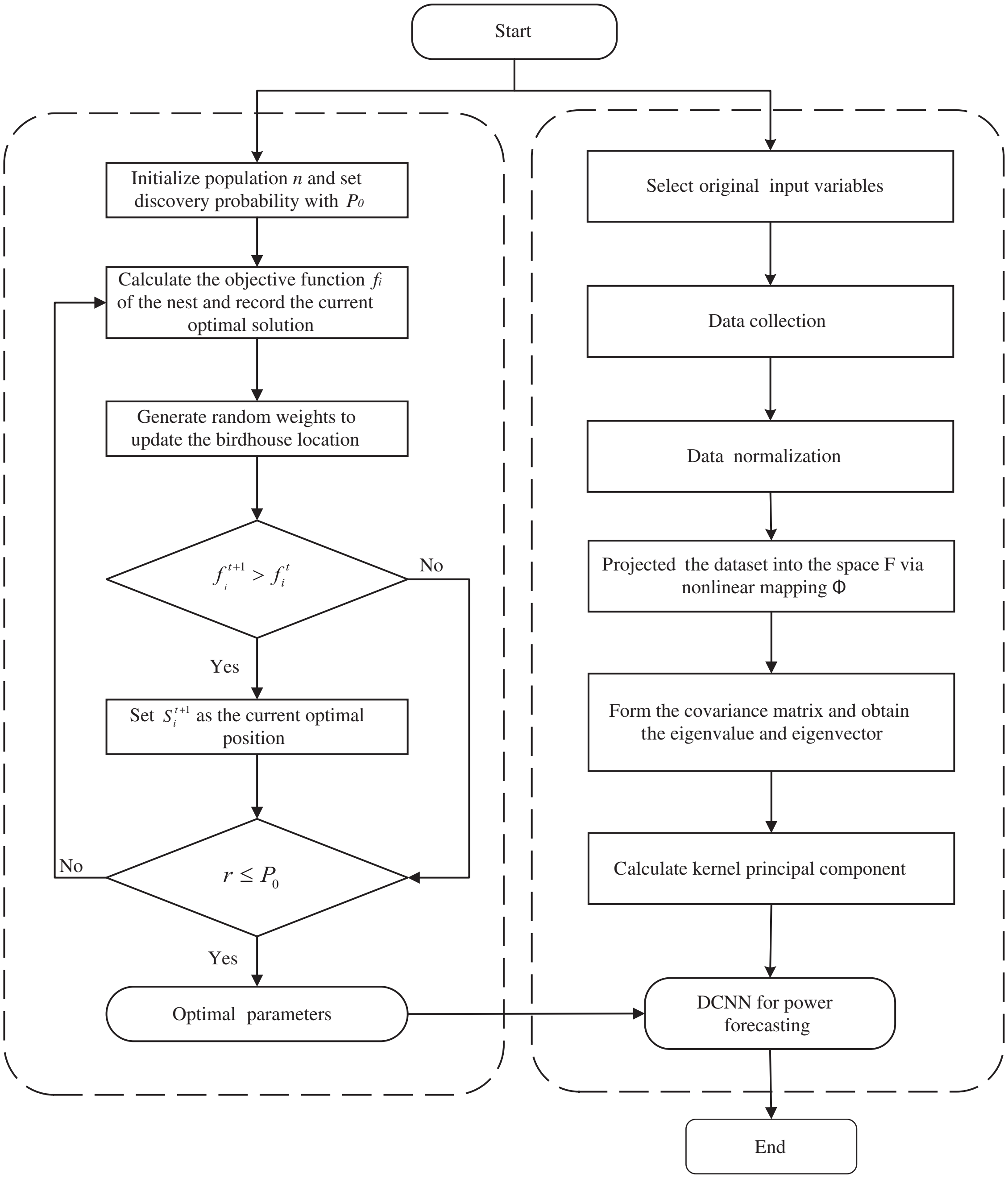
Start
Initialize population n and set discovery probability with P_0
Calculate the objective function f_i of the nest and record the current optimal solution
Generate random weights to update the birdhouse location
$f_i^{t+1} > f_i^t$
No
Yes
Set S_i^{t+1} as the current optimal position
$r \le P_0$
No
Yes
Optimal parameters
Select original input variables
Data collection
Data normalization
Projected the dataset into the space F via nonlinear mapping Φ
Form the covariance matrix and obtain the eigenvalue and eigenvector
Calculate kernel principal component
DCNN for power forecasting
End

Figure 2: Prediction flow based on KPCA and DCNN optimized by MCS

(5) Simulation prediction. The prediction model obtained from the above optimization training is employed to predict PV power, and the proposed model is compared with MCS-DCNN, CS-DCNN, DCNN and BPNN. Furthermore, the results are analyzed and evaluated by adopting root mean square error (RMSE), mean absolute percentage error (MAPE) and mean absolute error (MAE). These three indicators are calculated as follows [37]:

$$RMSE = \sqrt{(\frac{1}{N})\sum_{t=1}^{N}(y_t - y_t^*)^2} \tag{16}$$

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{y_t - y_t^*}{y_t}\right| \times 100 \tag{17}$$

$$MAE = \left(\frac{1}{N}\right)\sum_{i=1}^{N}|y_t - y_t^*| \tag{18}$$

where y_t is the actual PV power value, y_t^* is the predicted PV power value and N is the number of data sets. The smaller the value of the above indicators, the higher the prediction accuracy.

4 Case Study

4.1 KPCA Analysis

The data used for the calculations in this section is from a PV farm in China. In this paper, weather data and PV power data are selected as the initial input features for the model. In particular, weather data contains 12 independent variables: total precipitation, cloud ice content, surface air pressure, relative humidity, total cloud cover, horizontal wind speed, vertical wind speed, atmospheric temperature, surface solar radiation, ground thermal radiation, extra-atmospheric solar radiation and sediment amount. In addition to these factors, the PV power data at T-i ($i = 1, 2, 3, 4, 5, 6$) time points are also selected as the initial input features.

The variance contribution and cumulative contribution of the 18 indicators are illustrated in Tab. 1. In order to more visually observe and compare the calculated results of PCA and KPCA, Figs. 3 and 4 are plotted.

As can be seen from Tab. 1 and Figs. 3 and 4, the contribution rate of the first principal component of KPCA reached 70.39%, covering 70.39% of information of original data. After the principal component analysis of each indicator, the contribution rate of the first principal component was obtained to be lower, only 32.56%, which was much smaller than the contribution rate of the first principal component of KPCA. What's more, if the cumulative contribution rate exceeds 95%, PCA requires the extraction of 7 principal components, which is closer to the original number of indicators than KPCA, which only requires the extraction of 3 principal components. It can be concluded that KPCA is significantly stronger than PCA in terms of dimensionality reduction and problem simplification, and the extracted principal components retain sufficient information of the original data.

Table 1: Results of PCA and KPCA calculations

Ingredients	PCA		KPCA	
	Contribution of variance/%	Cumulative variance contribution/%	Contribution of variance/%	Cumulative variance contribution/%
1	32.56	32.56	70.39	70.39
2	25.29	57.85	16.29	86.68
3	16.15	74	8.92	95.6
4	9.57	83.57	3.16	98.76
5	5.95	89.52	0.77	99.53
6	3.28	92.8	0.35	99.88
7	2.25	95.05	0.06	99.94
8	2.08	97.13	0.05	99.99
9	1.22	98.35	0.01	100
10	0.99	99.34	0	100
11	0.35	99.69	0	100
12	0.19	99.88	0	100
13	0.09	99.97	0	100
14	0.02	99.99	0	100
15	0.01	100	0	100
16	0	100	0	100
17	0	100	0	100
18	0	100	0	100

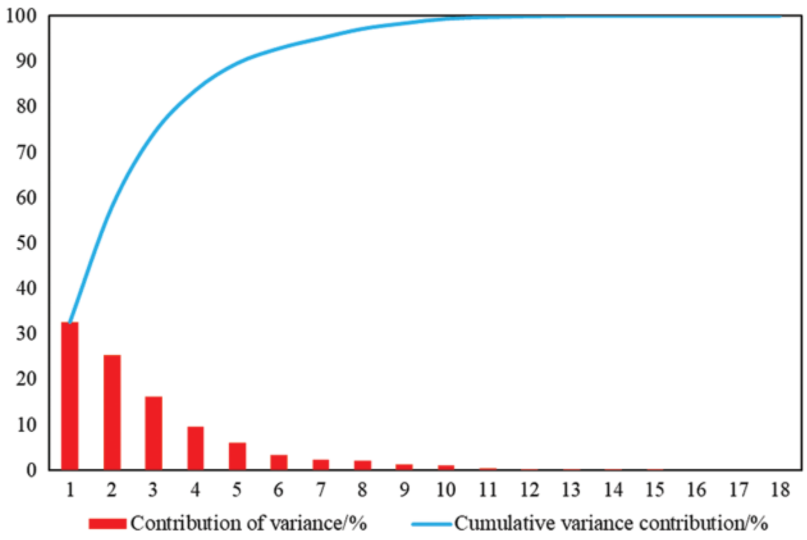

Figure 3: Total variance of PCA

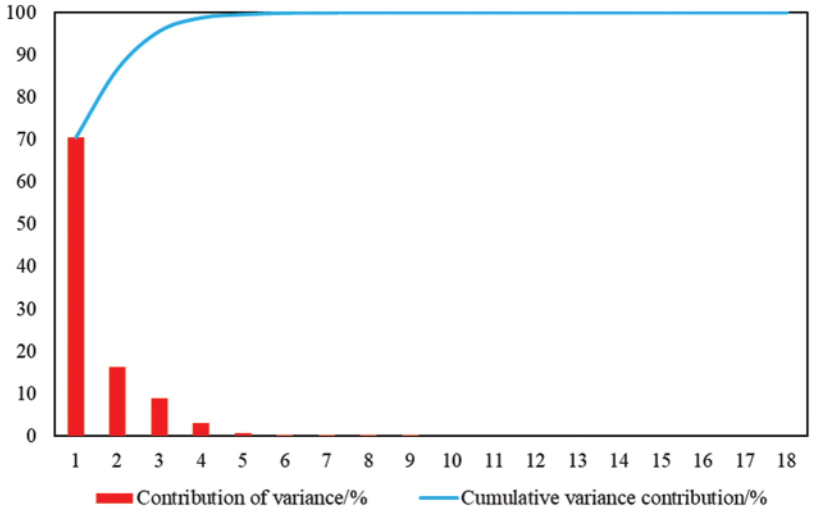

Figure 4: Total variance of KPCA

The coefficients of the selected principal components are found by dividing each autonomous component loading vector by the arithmetic square root of the eigenvalues of each autonomous component, and the output component matrix is calculated as displayed in Tab. 2. The components in Tab. 2 form the final input indicators for the prediction model.

Table 2: Component matrix for KPCA

Factor	Component		
	1	2	3
x1	−0.770	−0.072	−0.744
x2	0.683	0.262	0.310
x3	−0.454	0.243	0.756
x4	−0.290	0.575	−0.798
x5	0.471	−0.010	0.464
x6	0.429	−0.637	0.416
x7	−0.734	0.803	−0.849
x8	−0.789	−0.085	−0.523
x9	−0.121	0.672	−0.837
x10	−0.092	−0.171	0.805
x11	0.303	0.560	−0.433
x12	−0.217	−0.544	0.664
x13	−0.795	−0.766	−0.188
x14	0.733	0.003	−0.262
x15	−0.466	0.012	−0.363
x16	0.470	0.419	−0.578
x17	0.023	−0.481	0.778
x18	−0.109	0.864	0.991

4.2 Prediction Results Based on the Proposed Model

This section compares the different forecast effects for the spring, summer, autumn and winter. April 2019, July 2019, October 2019 and January 2020 are taken as representatives of the spring, summer, autumn and winter respectively for forecasting. Eight days are selected as the test sample, and the remaining data are employed as the training sample. Fig. 5 displays the comparison between the actual and predicted PV power values in the test sample set.

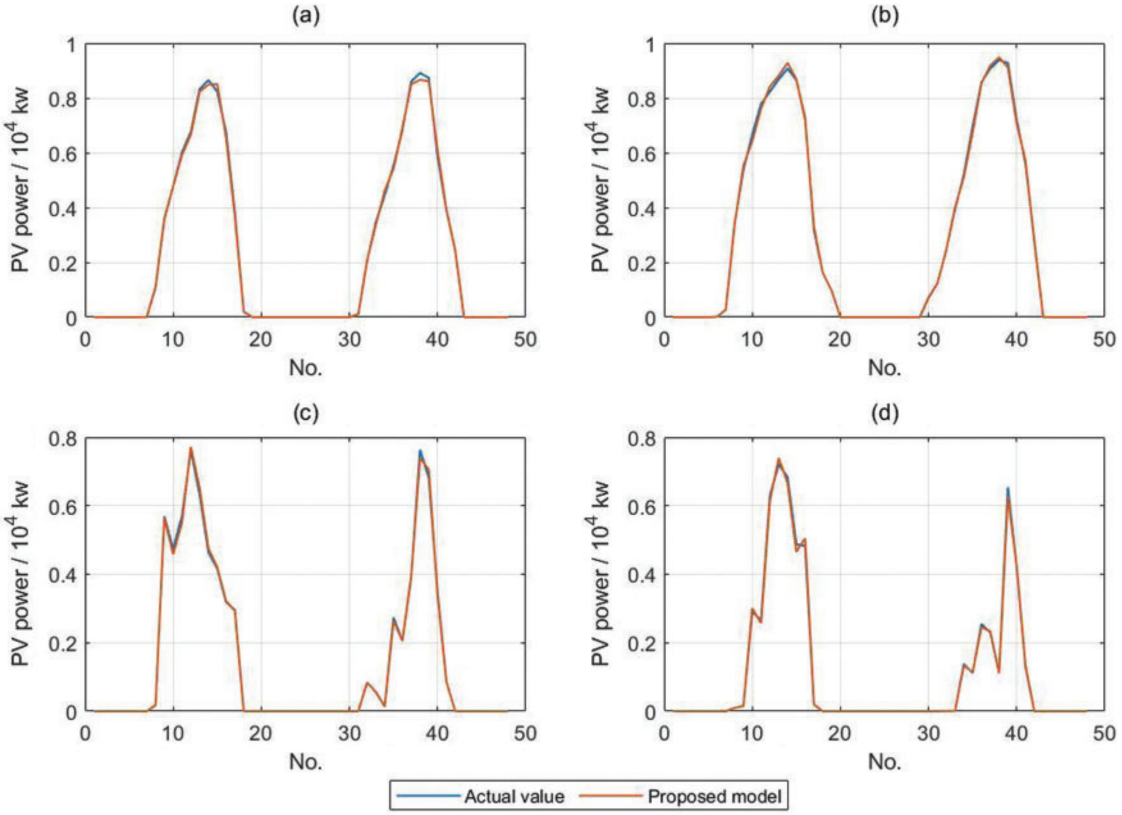

Figure 5: Comparison of PV power prediction results for four seasons: (a) Spring; (b) Summer; (c) Autumn; (d) Winter

As can be seen in Fig. 5, the PV curve is fuller and smoother in spring and summer. The summer curve is particularly pronounced, mostly on sunny days when the PV output is stable and the power values are high. In autumn, there are spikes and jitters in the PV output due to variable weather, and power values are also generally lower. In winter, the PV curve is narrower, which can be interpreted as shorter sunrise times and shorter maintenance of maximum daily light hours. The RMSE, MAPE and MAE of the test set are shown in Tab. 3, demonstrating the overall prediction effectiveness of the model proposed in this paper. In general, the spring and summer seasons give better predictions, while winter has the worst prediction accuracy, which is in line with our general perception.

Table 3: Evaluation of the accuracy of PV power prediction in four seasons

Parameters	Spring	Summer	Autumn	Winter
RMSE	1.236	1.119	1.333	1.556
MAPE	1.820	1.568	1.894	2.015
MAE	0.798	0.673	0.951	1.167

4.3 Comparative Analysis

To demonstrate the prediction performance of KPCA-MCS-DCNN, four models, namely MCS-DCNN, CS-DCNN, DCNN and BPNN, are selected for comparison in this paper. The spring test sample is discussed below as an example.

Fig. 6 shows the prediction results of MCS-DCNN, CS-DCNN, DCNN, BPNN and the model proposed in this paper for the PV power test set.

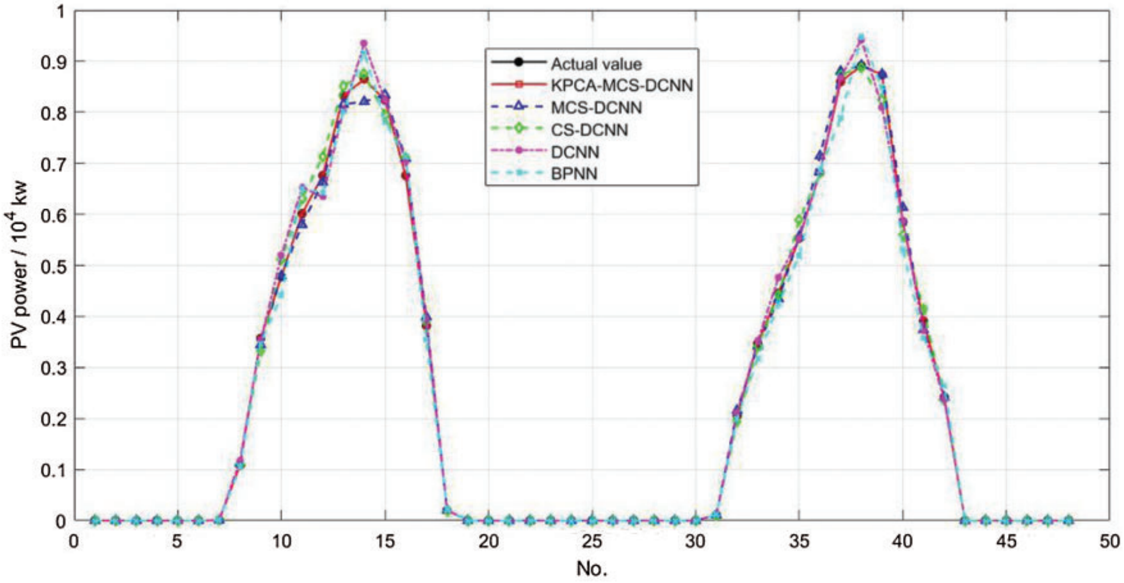

Figure 6: Comparison of predicted results

First, the maximum and minimum relative distances between the actual power values and the predicted values can be seen in Fig. 6. In the BPNN model, the maximum relative distance is 0.071134 and the minimum relative distance is 0.000064; in the DCNN model, the maximum and minimum relative distances are 0.069996 and 0.000059, respectively. Both values of DCNN are smaller than those of BPNN, indicating that the prediction accuracy of DCNN is relatively higher than that of DCNN. In the CS-DCNN model, the maximum and minimum relative distances are 0.049436 and 0.000036, respectively. The difference between the maximum and minimum relative distances is 0.049401, which is smaller than that of the BPNN and DCNN models, indicating that the stability of the CS-DCNN power prediction model is higher than that of the BPNN and DCNN. In the KPCA-MCS-DCNN power prediction model proposed in this paper, the maximum relative distance is only 0.026776, the minimum relative distance is 0.000007, and the deviation between these two distances is only 0.026769. These three values are smaller for the KPCA-MCS-DCNN model compared to the other four prediction models, suggesting that the proposed power prediction model has higher prediction accuracy and superior prediction stability.

Tab. 4 and Fig. 7 demonstrate the RMSE, MAPE and MAE of the individual models for the overall prediction results.

As can be seen from Tab. 4 and Fig. 7, the RMSE of the model proposed in this paper is calculated to be 1.236, while the RMSE of the MCS-DCNN, CS-DCNN, DCNN and BPNN models are calculated to be 1.738, 2.059, 2.825 and 3.354, respectively. This implies that the prediction results of the model proposed in this paper have small errors and the highest overall accuracy of prediction. Similarly, the calculated results of MAPE (1.820) and MAE (0.798) of the KPCA-MCS-DCNN model are both optimal. In short, RMSE, MAPE and MAE of KPCA-MCS-DCNN are all minimum values. It is worth noting that the smaller the values of these three indicators are, the stronger the robustness of the model is. Therefore, it can be concluded that the robustness of KPCA-MCS-DCNN is better. Compared with the MCS-DCNN model, KPCA overcomes the adverse effects of unconventional data on model training caused by sudden changes in the influencing factors. The prediction performance of the MCS-DCNN model is better than that of the CS-DCNN model, indicating that the modified CS algorithm plays a more satisfactory optimization role. The overall prediction performance of the DCNN model is better than that of the BPNN model, indicating that the overall prediction performance of the DCNN model, a type of deep learning algorithm, is significantly superior. Overall, the KPCA-MCS-DCNN model has the best prediction performance, followed by the MCS-DCNN model, CS-DCNN model and DCNN model, and the BPNN model has the worst prediction performance.

Table 4: RMSE, MAPE, and MAE values for all models

Parameters	KPCA-MCS-DCNN	MCS-DCNN	CS-DCNN	DCNN	BPNN
RMSE	1.236	1.738	2.059	2.825	3.354
MAPE	1.820	2.685	3.672	4.053	5.773
MAE	0.798	1.281	1.711	2.002	2.822

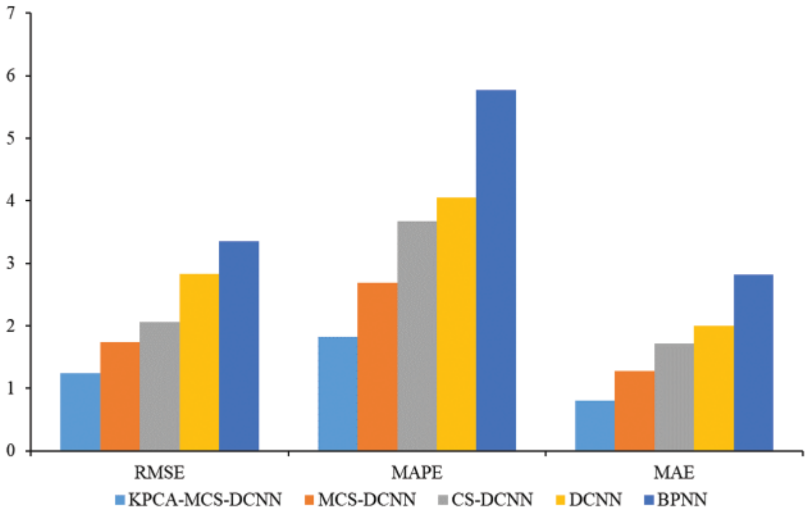

Figure 7: RMSE, MAPE and MAE of the predicted results

In conclusion, the proposed model can effectively reduce the PV power prediction error by optimizing the DCNN model with the MCS algorithm. KPCA model can not only reduce the noise data of the input variables and improve the validity of the input information, but also ensure the integrity of the input information, thus improving the accuracy and robustness of PV power prediction. The effectiveness of the proposed PV power prediction model is demonstrated by the data calculation results.

5 Conclusions and Discussion

Solar energy, as a low operating cost, wide range of applications, inexhaustible, environmentally friendly and harmless renewable energy, has a broad development prospect in the field of electricity. However, the power output of photovoltaic power generation utilizing solar energy as the energy source is greatly influenced by the weather and fluctuates significantly. The instability of PV power has a large impact on the power grid, making it very difficult to plan and control the power system. Consequently, achieving more accurate PV power prediction can effectively help the power sector to make reasonable energy planning and dispatching decisions and promote PV consumption. In this paper, a novel approach for PV power prediction based on KPCA-MCS-DCNN is proposed, in which the key factors are extracted employing KPCA. These key factors are brought into DCNN for learning and training, and MCS is applied to optimize the parameters in DCNN, thereby enhancing its prediction accuracy. An empirical analysis is carried out on a PV plant in China as an example, and MCS-DCNN, CS-DCNN, DCNN and BPNN are selected for comparison with the proposed model. The experimental results verify that the proposed model has better prediction accuracy than the comparison model and can achieve good prediction results, thus providing a new idea and reference for PV power prediction. The main conclusions are summarized as follows:

(1) The input indexes of the PV power prediction model are downscaled and analyzed by KPCA, while PCA is selected for comparison. It can be concluded that the dimensionality reduction of KPCA is significantly stronger than that of PCA, and KPCA ensures sufficient information of the original data even with fewer principal components.

(2) The innovative KPCA-MCS-DCNN model is proposed and applied to PV power prediction with RMSE of 1.236, MAPE of 1.820, and MAE of 0.798 in the spring test dataset. On the basis of KPCA to extract principal components, this hybrid model optimizes the DCNN model employing the MCS algorithm. Hence, a complex relationship between input data and PV power is constructed to predict short-term PV power.

(3) MCS-DCNN, CS-DCNN, DCNN and BPNN are selected for comparison and it is concluded that the proposed power prediction model has higher prediction accuracy and more satisfactory robustness. The different prediction results of spring, summer, autumn and winter seasons are obtained utilizing KPCA-MCS-DCNN and excellent prediction results are all achieved. All of these indicate that the robustness of the proposed model in this paper is better.

In brief, this paper starts from dimensionality reduction of PV power prediction impact factors, and then proposes a hybrid deep learning model based on MCS-DCNN to predict short-term PV power and compares it with other four intelligent prediction models. In future research, more advanced deep learning models can be employed for PV power prediction to obtain more accurate PV power prediction results. In addition, there is room to improve the prediction speed of deep learning models.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Stewart, A. D., Gardiner, M., MacDonald, J., Williams, H. (2021). The effect of harness suspension on a simulated maintenance task efficacy in the renewable energy industry. *Applied Ergonomics, 90,* 103247. DOI 10.1016/j.apergo.2020.103247.
2. Zhong, T., Zhang, K., Chen, M., Wang, Y., Zhu, R. et al. (2021). Assessment of solar photovoltaic potentials on urban noise barriers using street-view imagery. *Renewable Energy, 168,* 181–194. DOI 10.1016/j.renene.2020.12.044.
3. Wang, Y., Zhu, L., Xue, H. (2020). Ultra-short-term photovoltaic power prediction model based on the localized emotion reconstruction emotional neural network. *Energies, 13(11),* 2857. DOI 10.3390/en13112857.
4. Guo, X., Gao, Y., Zheng, D., Ning, Y., Zhao, Q. (2020). Study on short-term photovoltaic power prediction model based on the stacking ensemble learning. *Energy Reports, 6,* 1424–1431. DOI 10.1016/j.egyr.2020.11.006.
5. Shi, M., Yin, R., Wang, Y., Li, D., Han, Y. et al. (2020). Photovoltaic power interval forecasting method based on kernel density estimation algorithm. *IOP Conference Series: Earth and Environmental Science*, vol. 615, no. 1, pp. 12062. Changsha, China, IOP Publishing.
6. Zhou, Y., Zhou, N., Gong, L., Jiang, M. (2020). Prediction of photovoltaic power output based on similar day analysis, genetic algorithm and extreme learning machine. *Energy, 204,* 117894. DOI 10.1016/j.energy.2020.117894.
7. Louzazni, M., Mosalam, H., Khouya, A., Amechnoue, K. (2020). A non-linear auto-regressive exogenous method to forecast the photovoltaic power output. *Sustainable Energy Technologies and Assessments, 38,* 100670. DOI 10.1016/j.seta.2020.100670.
8. Li, Y., Su, Y., Shu, L. (2014). An ARMAX model for forecasting the power output of a grid connected photovoltaic system. *Renewable Energy, 66,* 78–89. DOI 10.1016/j.renene.2013.11.067.
9. Malvoni, M., Giorgi, M. G. D., Congedo, P. M. (2016). Data on support vector machines (SVM) model to forecast photovoltaic power. *Data in Brief, 9(C),* 13–16. DOI 10.1016/j.dib.2016.08.024.
10. Malvoni, M., De Giorgi, M. G., Congedo, P. M. (2014). Photovoltaic power forecasting using statistical methods: Impact of weather data. *IET Science Measurement & Technology, 8(3),* 90–97. DOI 10.1049/iet-smt.2013.0135.
11. Zheng, Z. W., Chen, Y. Y., Huo, M. M., Zhao, B. (2011). An overview: The development of prediction technology of wind and photovoltaic power generation. *Energy Procedia, 12,* 601–608. DOI 10.1016/j.egypro.2011.10.081.
12. Fanchette, Y., Ramenah, H., Tanougast, C., Benne, M. (2020). Applying johansen VECM cointegration approach to propose a forecast model of photovoltaic power output plant in Reunion Island. *AIMS Energy, 8(2),* 179–213. DOI 10.3934/energy.2020.2.179.
13. Ghanim, T. M., Khalil, M. I., Abbas, H. M. (2020). Comparative study on deep convolution neural networks DCNN-based offline arabic handwriting recognition. *IEEE Access, (99),* 1. DOI 10.1109/ACCESS.2020.2994290.
14. Dong, Q., Liu, B., Hu, Z. (2020). Non-uniqueness phenomenon of object representation in modeling IT cortex by deep convolutional neural network (DCNN). *Frontiers in Computational Neuroscience, 14,* 35. DOI 10.3389/fncom.2020.00035.
15. Choi, J. Y., Lee, B. (2020). Ensemble of deep convolutional neural networks with gabor face representations for face recognition. *IEEE Transactions on Image Processing, 29,* 3270–3281. DOI 10.1109/TIP.83.
16. Yu, J., Schumann, A. W., Sharpe, S. M., Li, X., Boyd, N. S. (2020). Detection of grassy weeds in bermudagrass with deep convolutional neural networks. *Weed Science, 68(5),* 545–552. DOI 10.1017/wsc.2020.46.

17. Wei, J., He, J., Zhou, Y., Chen, K., Tang, Z. et al. (2019). Enhanced object detection with deep convolutional neural networks for advanced driving assistance. *IEEE Transactions on Intelligent Transportation Systems, 21(4),* 1572–1583. DOI 10.1109/TITS.6979.

18. Basha, S. S., Vinakota, S. K., Pulabaigari, V., Mukherjee, S., Dubey, S. R. (2021). Autotune: Automatically tuning convolutional neural networks for improved transfer learning. *Neural Networks, 133,* 112–122. DOI 10.1016/j.neunet.2020.10.009.

19. Huang, Y., Ahmad, S., Fan, J., Shen, D., Yap, P. T. (2021). Difficulty-aware hierarchical convolutional neural networks for deformable registration of brain MR images. *Medical Image Analysis, 67,* 101817. DOI 10.1016/j.media.2020.101817.

20. Guedes, K. S., de Andrade, C. F., Rocha, P. A., Mangueira, R. D. S., de Moura, E. P. (2020). Performance analysis of metaheuristic optimization algorithms in estimating the parameters of several wind speed distributions. *Applied Energy, 268,* 114952. DOI 10.1016/j.apenergy.2020.114952.

21. Meng, X., Chang, J., Wang, X., Wang, Y. (2019). Multi-objective hydropower station operation using an improved cuckoo search algorithm. *Energy, 168,* 425–439. DOI 10.1016/j.energy.2018.11.096.

22. Chen, X., Yu, K. (2019). Hybridizing cuckoo search algorithm with biogeography-based optimization for estimating photovoltaic model parameters. *Solar Energy, 180,* 192–206. DOI 10.1016/j.solener.2019.01.025.

23. Zhu, H., Shi, Y., Wang, H., Lu, L. (2020). New feature extraction method for photovoltaic array output time series and its application in fault diagnosis. *IEEE Journal of Photovoltaics, 10(4),* 1133–1141. DOI 10.1109/JPHOTOV.5503869.

24. Liang, Y., Niu, D., Hong, W. C. (2019). Short term load forecasting based on feature extraction and improved general regression neural network model. *Energy, 166,* 653–663. DOI 10.1016/j.energy.2018.10.119.

25. Olanrewaju, L., Oyebiyi, O., Misra, S., Maskeliunas, R., Damasevicius, R. (2020). Secure ear biometrics using circular kernel principal component analysis, chebyshev transform hashing and bose–Chaudhuri–Hocquenghem error-correcting codes. *Signal, Image and Video Processing, 14(5),* 847–855. DOI 10.1007/s11760-019-01609-y.

26. Djerida, A., Zhao, Z., Zhao, J. (2020). Background subtraction in dynamic scenes using the dynamic principal component analysis. *IET Image Processing, 14(2),* 245–255. DOI 10.1049/iet-ipr.2018.6095.

27. Zhou, T., Peng, Y. (2020). Kernel principal component analysis-based Gaussian process regression modelling for high-dimensional reliability analysis. *Computers & Structures, 241,* 106358. DOI 10.1016/j.compstruc.2020.106358.

28. Kang, T., Yao, J., Jin, M., Yang, S., Duong, T. (2018). A novel improved cuckoo search algorithm for parameter estimation of photovoltaic (PV) models. *Energies, 11(5),* 1060. DOI 10.3390/en11051060.

29. Chen, X., Wang, J. (2016). A novel hybrid cuckoo search algorithm for optimizing vehicle routing problem in logistics distribution system. *Journal of Computational and Theoretical Nanoscience, 13(1),* 114–119. DOI 10.1166/jctn.2016.4776.

30. Puspaningrum, A., Suheryadi, A., Sumarudin, A. (2020). Implementation of cuckoo search algorithm for support vector machine parameters optimization in pre collision warning. *IOP Conference Series Materials Science and Engineering, 850,* 12027. DOI 10.1088/1757-899X/850/1/012027.

31. Xu, X. M. (2017). *Optimal investment decision of complex power grid based on power demand and investment capacity* (*Ph.D. Thesis*). North China Electric Power University, China.

32. Yang, Y., Xie, R., Jia, W., Chen, Z., Yang, Y. et al. (2021). Accurate and automatic tooth image segmentation model with deep convolutional neural networks and level set method. *Neurocomputing, 419,* 108–125. DOI 10.1016/j.neucom.2020.07.110.

33. Pasyar, P., Mahmoudi, T., Kouzehkanan, S. Z. M., Ahmadian, A., Arabalibeik, H. et al. (2021). Hybrid classification of diffuse liver diseases in ultrasound images using deep convolutional neural networks. *Informatics in Medicine Unlocked, 22,* 100496. DOI 10.1016/j.imu.2020.100496.

34. Pedziwiatr, M. A., Kümmerer, M., Wallis, T. S., Bethge, M., Teufel, C. (2021). Meaning maps and saliency models based on deep convolutional neural networks are insensitive to image meaning when predicting human fixations. *Cognition, 206,* 104465. DOI 10.1016/j.cognition.2020.104465.

35. Gorban A. N., Mirkes E. M., Tukin I. Y. (2020). How deep should be the depth of convolutional neural networks: A backyard dog case study. *Cognitive Computation, 12(1),* 388–397. DOI 10.1007/s12559-019-09667-7.

36. Li, W., Zhang, X., Peng, Y., Dong, M. (2021). Spatiotemporal fusion of remote sensing images using a convolutional neural network with attention and multiscale mechanisms. *International Journal of Remote Sensing, 42(6),* 1973–1993. DOI 10.1080/01431161.2020.1809742.

37. Xu, P., Aamir, M., Shabri, A., Ishaq, M., Aslam, A. et al. (2020). A new approach for reconstruction of imfs of decomposition and ensemble model for forecasting crude oil prices. *Mathematical Problems in Engineering,* 1325071. DOI 10.1155/2020/1325071.