energies

MDPI

Article

A Hybrid GA–PSO–CNN Model for Ultra-Short-Term Wind Power Forecasting

Jie Liu ^{1,2}, Quan Shi ^{1,2}, Ruilian Han ^{1,2} and Juan Yang ^{1,2,*}

[1] Center for Energy Environmental Management and Decision-Marking, China University of Geosciences, Wuhan 430074, China; liujie2018@cug.edu.cn (J.L.); s1052686265@163.com (Q.S.); rlhan_1127@163.com (R.H.)
[2] School of Economics and Management, China University of Geosciences, Wuhan 430074, China
* Correspondence: yangjuan@cug.edu.cn or yj_sail@163.com

Abstract: Accurate and timely wind power forecasting is essential for achieving large-scale wind power grid integration and ensuring the safe and stable operation of the power system. For overcoming the inaccuracy of wind power forecasting caused by randomness and volatility, this study proposes a hybrid convolutional neural network (CNN) model (GA–PSO–CNN) integrating genetic algorithm (GA) and a particle swarm optimization (PSO). The model can establish feature maps between factors affecting wind power such as wind speed, wind direction, and temperature. Moreover, a mix-encoding GA–PSO algorithm is introduced to optimize the network hyperparameters and weights collaboratively, which solves the problem of subjective determination of the optimal network in the CNN and effectively prevents local optimization in the training process. The prediction effectiveness of the proposed model is verified using data from a wind farm in Ningxia, China. The results show that the MAE, MSE, and MAPE of the proposed GA–PSO–CNN model decreased by 1.13–9.55%, 0.46–7.98%, and 3.28–19.29%, respectively, in different seasons, compared with Single–CNN, PSO–CNN, ISSO–CNN, and CHACNN models. The convolution kernel size and number in each convolution layer were reduced by 5–18.4% in the GA–PSO–CNN model.

Keywords: convolutional neural network; ultra-short-term; wind power forecasting; hybrid; genetic algorithm; particle swarm optimization

check for updates

Citation: Liu, J.; Shi, Q.; Han, R.; Yang, J. A Hybrid GA–PSO–CNN Model for Ultra-Short-Term Wind Power Forecasting. *Energies* **2021**, *14*, 6500. https://doi.org/10.3390/en14206500

Academic Editor: Adrian Ilinca

Received: 30 August 2021
Accepted: 5 October 2021
Published: 11 October 2021

1. Introduction

The unsustainability of conventional energy sources, in addition to their knock-on effects on local- and global-scale climate, necessitates the use of naturally occurring energies [1]. The need for the use of renewable energy helps human beings transmit the world to future generations [2]. Developing wind power is an important measure to alleviate energy shortage [3], reduce environmental pollution, and combat global climate change. The global installed capacity of wind power is increasing rapidly. In 2020, the global installed capacity reached 651 GW, which is over three times greater than the 2010 total. The demand for generating more electricity is absorbing huge attention at a rapid pace [4]. Wind power is the fifth-greatest power source in the world after coal, natural gas, water power, and nuclear power [5]. Wind power is generated by the kinetic energy of the air moving around a boiler turbine [6]. It is affected by a variety of meteorological factors, including temperature, humidity, and pressure, with strong randomness, volatility, and instability [7,8]. After accessing the power grid, wind power introduces a new factor of uncertainty to the grid and may have a negative impact on the grid integrity in terms of the power quality, system security, and system stability due to its rapid changes and its intermittent and diffuse nature [9]. Accurate wind power forecasting can effectively reduce the fuzziness of wind power and reduce uncertainty to a smaller error range. It is advisable to adjust the dispatching plan and optimize the grid structure to reduce the damage to the stability of the grid caused by wind power grid connection and to improve the wind power utilization rate and economic benefits [10].

A variety of models have been proposed for wind power forecasting. Early research mainly focused on physical models based on numerical weather forecasting [11,12] and statistical models (persistence method [13], ARIMA [14,15]). Physical models are based on influencing factors (such as contour lines, air pressure, and temperature) and equipment characteristics around the wind farm to establish complex mathematical models. Li et al. [16] proposed a physical approach to wind power prediction. Cassola and Burlando [17] used Kalman filtering to correct the physical model output and improve the wind speed forecasting performance. This type of model usually requires significant computing resources and time, with poor results in short-term prediction [18]. In contrast, statistical models do not require complex hydrodynamic and thermodynamic models. They require less work and perform better in short-term forecasting. Chen and Yu [19] used Kalman filtering to improve the persistence method, which reduces the prediction error and improves the stability of the system. Milligan et al. [20] used the auto-regressive moving average (ARMA) model to forecast the power of wind farms in Oregon, USA. Based on improvements to the autoregressive (AR) model, Zhang et al. [21] proposed an innovative autoregressive dynamic adaptive (ARDA) model, which was more accurate than the ARMA model, with faster calculation and better dynamic adaptability to data fluctuations. In addition, some scholars have proposed a combination model, such as Liu et al. [22] proposed an ultra short-time forecasting method based on the Takagi–Sugeno fuzzy model for wind power and wind speed. Liu et al. [23] developed a forecasting system based on a data pretreatment strategy, a modified multi-objective optimization algorithm, and several forecasting models to further enhance wind power forecasting performance and stability.

With the development of computer technology, shallow artificial neural network (ANN) models based on artificial intelligence have been widely used in wind power prediction. Traditional methods have higher requirements on the time series of historical data and often use linear methods to deal with nonlinear loads. Artificial intelligence methods have gradually become a research hotspot in recent years [24]. ANN models can effectively capture the nonlinear relationships in data with a stronger fitting ability and wider application prospects. Han et al. [25] proposed a BP neural network model for wind power prediction based on the Tabu search algorithm and demonstrated that the model could improve prediction accuracy and convergence speed with appropriate input parameters. Han et al. [26] reported that the training speed and prediction accuracy of an RBF network was better than those of a BP network in 3 h wind power prediction. Carolin Mabel and Fernandez [27] used a fully connected neural network (FNN) to forecast the wind energy power of the Muppandal wind farm in Tamil Nadu (India), reporting good agreement with the measured values of the actual wind farm, verifying the accuracy of the model. Wang et al. [28] proposed a hybrid model based on empirical mode decomposition and an Elman neural network (ENN) to predict wind speed. Similar wind power prediction research using ANNs can be found in [29–31].

With the limited information extracted from complex wind power data by shallow neural network models, it is difficult to conduct a comprehensive analysis of the factors affecting wind speed [32]; the requirements of deep data mining cannot be met [33–35]. Recently, deep learning network models (such as the deep belief network (DBN), recurrent neural network (RNN), and convolutional neural network (CNN)) have become popular solutions for wind power prediction. Unlike a shallow ANN, a deep neural network can spontaneously extract effective information from a large number of influencing factors, with a stronger nonlinear mapping ability [36,37]. Among different deep neural networks, a CNN can directly process multi-dimensional data samples, which is conducive to the extraction of inherent data features. Its unique weight-sharing structure can reduce the number of parameters, reducing the complexity of the network and effectively preventing overfitting. Several studies have applied CNNs to wind speed prediction. Solas et al. [38] proposed a short-term wind power prediction method based on a CNN and demonstrated better wind power forecasting performance than ARIMA and Gradient Supercharger.

Wang et al. [39] decomposed raw wind power data into different frequencies using wavelet transform and effectively trained the CNN to improve prediction accuracy. Liu et al. [40] proposed a novel approach to predicting short-term wind power by converting time series into images, using a CNN to analyze them. Compared with state-of-the-art techniques in wind power prediction (GRU, LSTM, and RNN), superior performance was demonstrated.

To further improve CNN performance, some studies integrated different methods into the CNN prediction model. Wang et al. [18] combined the information of nearby wind farms and used a CNN for short-term wind speed prediction to overcome the problem of insufficient historical wind speed data from new wind farms. Ju et al. [41] combined the Light-GBM classification algorithm with a CNN. The research results showed that this model had better prediction accuracy than SVM, Light-GBM, DNN, and CNN when considering the volatility of ultra-short-term wind power. Similarly, Liu et al. [42] combined wavelet packet decomposition with CNN. Yildiz et al. [43] combined variational mode decomposition with CNN. Zhang et al. [44] and Kuang et al. [45] introduced bidirectional LSTM to CNN, forming a novel hybrid wind speed forecasting model.

Although a CNN provides good prediction performance and wide applicability, efficiently determining the hyperparameters of the network (such as the number of convolutional layers and the size and number of kernels) and the connection weights (weights and bias) is always difficult. Traditional research usually uses manual parameter adjustment or empirical values to determine the relevant parameters, which requires significant manpower and material resources and cannot always produce the optimal parameter combination [46]. To obtain the optimal CNN parameters, some studies have applied an intelligent optimization algorithm such as a genetic algorithm (GA) or a particle swarm optimization (PSO) algorithm in the model. Oullette et al. [47] used a GA to obtain the optimal connection weight of a CNN. To solve the problem of slow training speed, Fedorovici et al. [48] combined the Gravity Search Algorithm, PSO, and a BP algorithm to prevent the network from falling into a local minimum while ensuring its fast convergence. Albeahdili et al. [49] used the connection weight and error function of a CNN as the particle and fitness function of a PSO to improve the backpropagation stage of the CNN. Huang et al. [50] fused an improved, simplified PSO with the stochastic gradient descent method (SGD) to complete the training of a CNN. These studies focused mainly on the optimization of CNN connection weights; the structure of the network was still determined subjectively by trial-and-error and experience.

A few studies have optimized the CNN network hyperparameters (such as the size and number of convolution kernels). Chen et al. [51] used a GA algorithm to optimize the number of nodes in the hidden layer and the training function of a CNN, and proposed a short-term wind power prediction model. Dong et al. [52] used a GA–PSO hybrid algorithm to optimize the hyperparameters of the neural network framework. These studies focused only on the structural optimization of the CNN; the learning of network connection weights used the BP algorithm. In actual problems, efficiently determining the optimal hyperparameter combination of the network while preventing the model from falling into a local optimal condition still requires further research.

Ultra-short-term prediction is in the range of minutes and hours, usually within 6 h, which is helpful in optimizing frequency modulation and spinning reserve capacity [53]. Ultra-short-term forecasting is mainly used to control the daily operation of wind farm units and real-time dispatching of a power grid; it is a forecasting method with the highest theoretical accuracy. It is the key index of power system real-time dispatching and the foundation of the intraday and real-time electricity market, has an important influence on the operation, control, and planning of a power system [54,55]. Improving the prediction accuracy and reliability can reduce the cost of equipment, optimize the grid structure, and improve the utilization rate of wind power and power quality. Decision-makers can determine a reasonable scheduling scheme according to the prediction results to ensure the safe and economic operation of the power system [56]. Ultra-short-term forecasting is the basis of probability prediction. The slight improvement in prediction accuracy is important in promoting the safe and economic operation of the power system [57]. Thus, ultra-short-term wind power prediction is critical.

Prediction errors for ultra-short-term wind power are caused mainly by inherent random factors and external random factors [51]. The inherent random factors include defects or imperfections in the prediction system itself. The external random factors include incomplete data or input data errors. When making an ultra-short-term wind power forecast, fan operation data and meteorological data are required [10]. The integrity and authenticity of these data have a great impact on prediction accuracy. Unlike the capability of generating conventional resources (coal and fossil fuels) according to the consumption and at specific and accurate schedules, the production of renewable energies is variable, which can change dramatically from time to time [58], further increasing the difficulty of ultra-short-term wind power prediction.

Some studies have shown that the accuracy of ultra-short-term wind power prediction depends largely on the performance of correlation mining between data [59–61]. With the complex space–time relationship of wind power output and inherent characteristics such as nonlinear randomness and intermittence, deep learning is an effective approach and can improve the accuracy of ultra-short-term wind power prediction. CNNs have been selected over other deep architectures as they are particularly suitable for tuning data with a grid-based structure [62]. A time-series represents a grid of samples spaced equally in time; CNNs are able to model this type of data. In addition, a CNN can use the information in a time series to extract common features at different levels from high-dimensional time-series data, with more stable predictions. Unlike other deep learning models, a CNN uses weight sharing technology, which has fewer parameters, reducing the amount and complexity of data and improving the learning efficiency and overall generalization of the model.

In summary, CNN has been widely used in complex wind power forecasting. Integrating the intelligent algorithm GA-PSO into CNN provides a feasible idea for determining the CNN network structure (such as the number of convolution layers, the size, and the number of convolution kernels, etc.) or the training connection weights (weights and biases). However, most existing research only focuses on optimizing network connection weights, and the network structure parameters are still subjectively determined. Thus, to further improve ultra-short-term wind power prediction accuracy and address the shortcomings of traditional CNN prediction models in which the hyperparameters are subjectively difficult to determine and likely to fall into a local optimum, a hybrid GA–PSO–CNN model based on a mix-encoding GA–PSO algorithm is proposed in this study for ultra-short-term wind power prediction. The model uses related wind farm influencing factors as inputs (wind speed, wind direction, temperature, etc.), conducts hybrid coding design for relevant parameters of the CNN, and uses a GA–PSO hybrid algorithm to synergistically optimize the hyperparameters and connection weights in the CNN, which solves the problem of subjective determination of the optimal network in the CNN and effectively prevents local optimization in the training process. By doing so, this paper makes novel contributions on two aspects: on the one hand, the GA-PSO hybrid algorithm in the proposed model realizes the automatic optimization of hyperparameters in CNN. The model can effectively determine the optimal hyperparameters combination, avoiding tedious and inefficient manual tuning. On the other hand, in the CNN training process, the global search capability of PSO is combined with the fast convergence of the BP algorithm to prevent the model from falling into a local optimum during the training process, which further improves the performance of the prediction model.

2. Methodology

2.1. Brief Introduction to CNN

As a representative network of deep learning, a CNN is a feedforward neural network containing a convolutional structure. CNN research began in the 1980s. Time delay network and LeNet-5 were the earliest CNN structures. With the development of deep learning theory and numerical computing equipment, the CNN has developed rapidly and is used in computer vision [63], natural language processing [64], and many other fields.

Unlike other neural networks, a CNN provides local connection, weight sharing, and pooling operation. These features can simplify the network structure by reducing the number of weights and the complexity of feature extraction. CNN structure consists of a convolution layer, a pooling layer, a full connection layer, and an output layer, as shown in Figure 1.

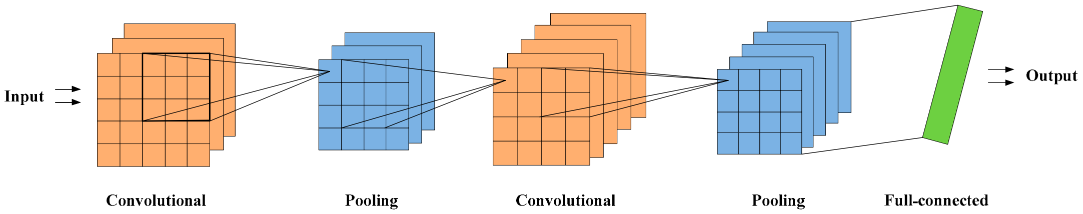


Figure 1. Convolutional neural network structure.

2.1.1. Convolutional Layer

The convolutional layer is the core of a CNN and is used to extract data features. The weights of the convolutional layer are shared by all feature mapping channels, which are also known as convolutional kernels or filters. The unique "weight sharing" significantly reduces the number of network parameters and is suitable for deep networks. The calculation formula of the convolutional layer is expressed as

$$y_j^l = \sum_{i \in C_{l-1}} x_i^{l-1} * w_j^l + b_j^l, j \in C_l \tag{1}$$

where x_i^{l-1} and y_j^l represent the output feature map of the ith channel at the $(l-1)$th layer and the jth output map at the lth layer, respectively; w_j^l and b_j^l are the weight and bias of the lth layer, respectively; C_{l-1} and C_l denote the number of channels in the $(l-1)$th layer and the lth layer, respectively.

2.1.2. Pooling Layer

After the convolution layer extracts the features, the pooling layer selects the features and filters the information of the output feature images. Its functions are to enlarge the local features of the image, speed up the calculation, and reduce the probability of overfitting. The maximum pooling method is a commonly used pooling method; it is calculated as

$$p^{l(i,j)} = \max_W \left\{ a^{l(i,t)} \right\} \tag{2}$$

where $a^{l(i,t)}$ represents the activation value of the tth neuron of the ith channel in the lth layer, $p^{l(i,j)}$ is the value after pooling, and W represents the range of the pooling area.

2.1.3. Full Connection Layer

The full connection layer of a CNN is equivalent to the hidden layer of a traditional feedforward neural network and is used to obtain the output by a nonlinear combination of extracted features. It is calculated as

$$y^l = w^l x^{l-1} + b^l \tag{3}$$

where x^{l-1} is the input of the $(l-1)$th layer; w^l, b^l, and y^l are the weight, bias, and output, respectively, of the lth layer.

2.1.4. Activation Layer

The activation layer is connected after the convolution layer and the full connection layer. Usually, Sigmoid is used as the activation function in output layers, and ReLU is used in the others.

$$\alpha_{\text{sigmoid}}(x) = \frac{1}{1+e^{-x}} \tag{4}$$

$$\alpha_{\text{ReLU}}(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \tag{5}$$

where x is the input variable of the layer.

2.2. GA–PSO Hybrid Algorithm

PSO is a random search algorithm developed by Eberhart and Kennedy in 1995 that simulates the foraging behavior of birds [65]. PSO has fewer internal parameters, a simple structure is easily implemented, and must only abstract the solution of the specific problem without relying on the principle of the problem. PSO can quickly iterate and converge. However, due to the update of the particle position in the particle swarm, it is mainly evolved by comparing its own position with the surrounding position and the current optimal position in the swarm particle. The single mode makes its convergence rate inefficient in the later stage of the calculation, and it is easy to fall into local optimum [66]. In comparison, GA can improve the diversity of solutions due to the existence of different evolutionary modes such as crossover and mutation. However, when GA is solved to a certain range, it often leads to a large number of redundant iterations of inaction, long calculation time, and reduced solution efficiency. Therefore, it is natural to think of combining PSO and GA to improve the performance of the algorithm.

In the later stages of the algorithm, the differences among the individuals in the particle swarm were reduced, resulting in a significant decline in the local search ability of the algorithm, and susceptibility to the local optimum. Moreover, the search scope of PSO was not sufficiently detailed; the global optimal solution can be missed. The global search ability and local search ability of the algorithm were both insufficient for complex high-dimensional problems. PSO was suitable for high-dimensional optimization problems with multiple local optimal solutions and low accuracy requirements.

In this study, a GA–PSO hybrid algorithm was proposed by combining PSO and the GA mutation operation. After the completion of the PSO iteration in each generation, the current global optimal particle was found and saved. The mutation operation was conducted at specific sites for each particle. This step can increase the population richness of the particle swarm such that the algorithm maintains a strong local search ability in later iterations. After the mutation operation, a particle is randomly selected from the new population and replaced with the previously saved global optimal particle to ensure that the global optimal particle enters the next iteration of the algorithm, which improves the convergence speed and searchability of the algorithm. As an integrated algorithm, PSO–GA combines the merits of GA and PSO but avoids their drawbacks. In addition, PSO–GA has stronger efficient optimization capability than GA or PSO alone [67]. The GA–PSO hybrid algorithm flowchart is shown in Figure 2. More details can be found in reference [68].

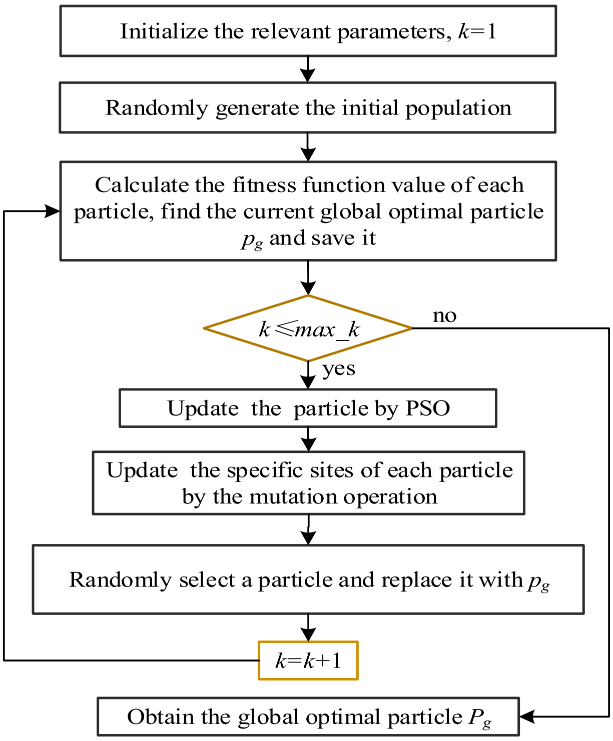

Initialize the relevant parameters, k=1
Randomly generate the initial population
Calculate the fitness function value of each particle, find the current global optimal particle p_g and save it
$k \leqslant max\_k$
no
yes
Update the particle by PSO
Update the specific sites of each particle by the mutation operation
Randomly select a particle and replace it with p_g
k=k+1
Obtain the global optimal particle P_g

Figure 2. GA–PSO hybrid algorithm flowchart.

3. Proposed GA–PSO–CNN Prediction Model

Hyperparameters in neural networks have always been a challenging problem, and they have a great impact on neural network performance. Empirical parameters in traditional research may only be applicable to a specific study; manual adjustment of parameters requires significant manpower and material resources. The gradient descent method also has obvious disadvantages in the neural network as it can easily fall into the local minimum point without obtaining the global optimum. Thus, a CNN hybrid prediction model (GA–PSO–CNN) optimized by a GA–PSO algorithm is proposed in this study. The model conducts a mixed coding design for the relevant parameters of the neural network; the hyperparameters use binary coding, and the connection weights use real number coding. The GA–PSO algorithm is used to iteratively optimize the whole particle; a mutation operator is added in the iteration process of binary coding to increase the diversity of the population, allowing the model to search for more hyperparameter combinations. The overall structure of the GA–PSO–CNN model is shown in Figure 3.

The GA–PSO–CNN model realizes synchronous optimization of hyperparameters and connection weights, avoiding cumbersome and inefficient manual parameter adjustment and empirical methods without the defects of the traditional gradient descent method to improve the accuracy and robustness of the prediction model.

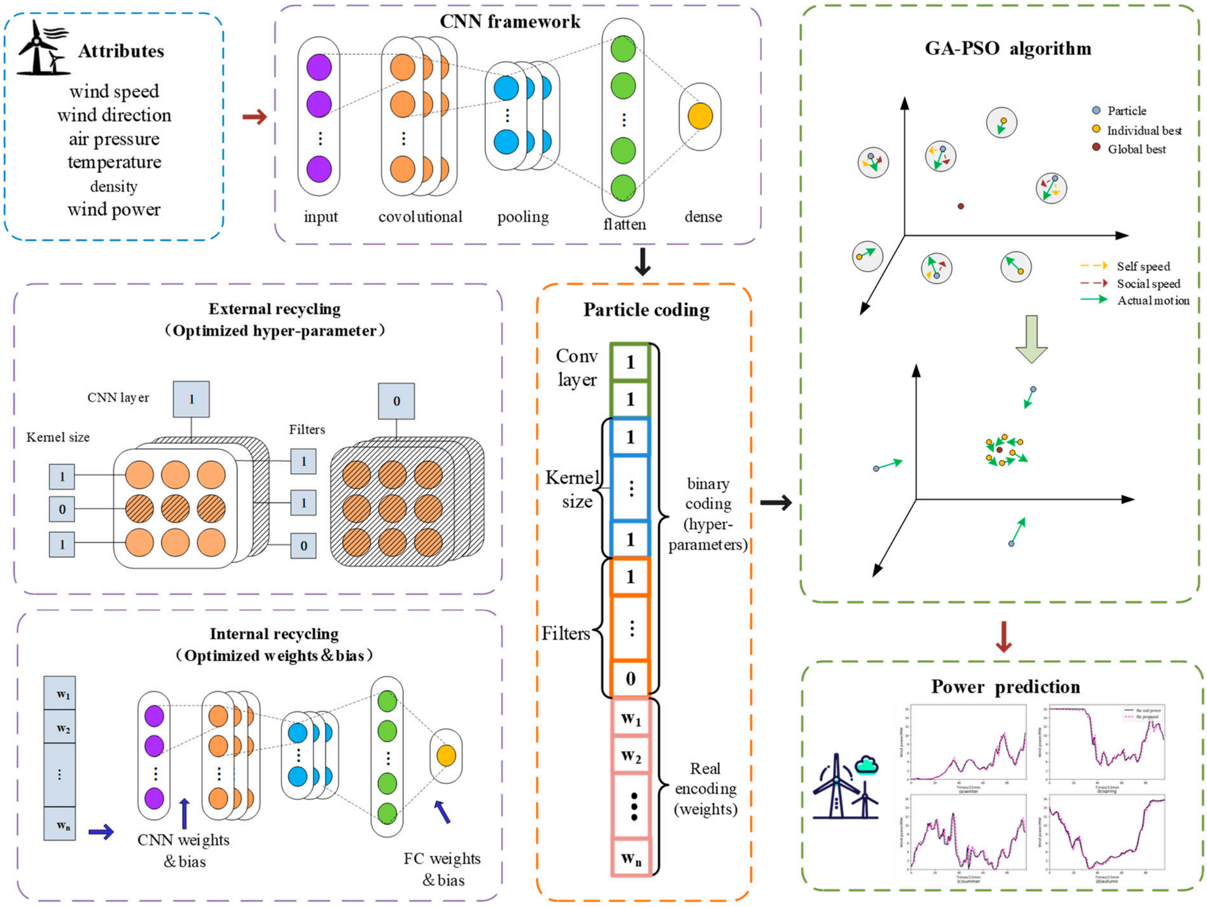


Figure 3. GA–PSO–CNN framework.

3.1. Coding Scheme

In a CNN, the performance of the model is closely related to the hyperparameters. In this study, the number of convolutional layers, the kernel size, and the number of filters in each layer are the hyperparameters that must be optimized in the prediction model. The values of the hyperparameters are all integers; binary coding is used in the front part of the particle, which ensures that the integer type of the hyperparameters remains unchanged in the iteration process and is convenient for mutation operations in the genetic algorithm.

A specific coding example of the parameters in the GA–PSO–CNN model is shown in Figure 4. The hyperparameter value was obtained by summing the binary values controlling all positions of the parameter.

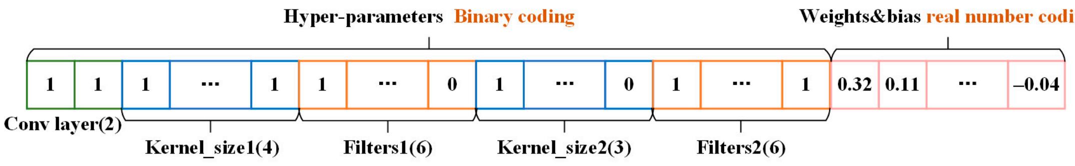


Figure 4. Particle example.

The latter part of the particle was used to optimize the connection weights of the CNN. Different hyperparameter values established different CNN structures. The length of the particle was determined by the maximum hyperparameter value in the CNN. In the algorithm, according to the specific hyperparameter value, the weights of the corresponding position in the particle were iterated, and the other positions remain unchanged.

3.2. Selection of Mixed Algorithms

CNNs are widely used to predict ultra-short-term wind power. The model architecture depends mainly on the number of convolutional layers, the kernel size, the number of filters, and other relevant parameters. Different hyperparameter combinations greatly affect the model performance. In this study, the PSO algorithm was used to update the position of the binary code of the particle to realize automatic hyperparameter optimization. Considering that the solution richness of the PSO algorithm decreased in the later iterations, the GA mutation operation was added to the PSO to ensure that particles were not the same and to enhance the local optimization ability of the algorithm in later iterations. ISSO can effectively optimize complex parameter combinations in the solution space [69]. Compared with the PSO algorithm, ISSO has been demonstrated to more efficiently converge to high-quality solutions and is more suitable for continuous problems [70]. Thus, ISSO is used for updating iterations in the real number coding of particles.

3.3. Initialization Method

The initialization of weights in deep learning models is important for the training of the model. If the initial weights are too small, the signal gradually shrinks in the process of transmission, and differences are difficult to generate, which is not conducive to model convergence. If the initial weights are too large, the signal increases layer by layer in the transmission process, resulting in gradient explosion. Xavier initialization was used in this study to ensure that a moderate initial weight was generated [55]. The Xavier initialization is shown in Equation (6).

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{fan_{in} + fan_{out}}}, \frac{\sqrt{6}}{\sqrt{fan_{in} + fan_{out}}}\right] \tag{6}$$

where fan_{in} and fan_{out} represent the number of nodes in the input and output, respectively. This method generates an evenly distributed initial weight value to activate each layer.

3.4. Fitness Function

In this study, the average absolute error of the prediction model using the training set was selected as the fitness function [71]:

$$fitness = \frac{1}{N}\sum_{i=1}^{N}|t_i - y_i| \tag{7}$$

where N represents the number of samples in the training set; $t(i)$ and $y(i)$ represent actual data and predicted data, respectively.

3.5. Model Process and Framework

Most previous studies only considered a single CNN problem (such as difficulty in hyperparameter tuning or local minimums). The proposed model optimizes both parts simultaneously to improve the performance of the neural network. The flow chart of the GA–PSO–CNN model is shown in Figure 5. The model uses GA–PSO combined with a BP algorithm to optimize the hyperparameters and connection weights of the network. The optimization process is described as follows. The basic formulas can be referred to separately [72,73].

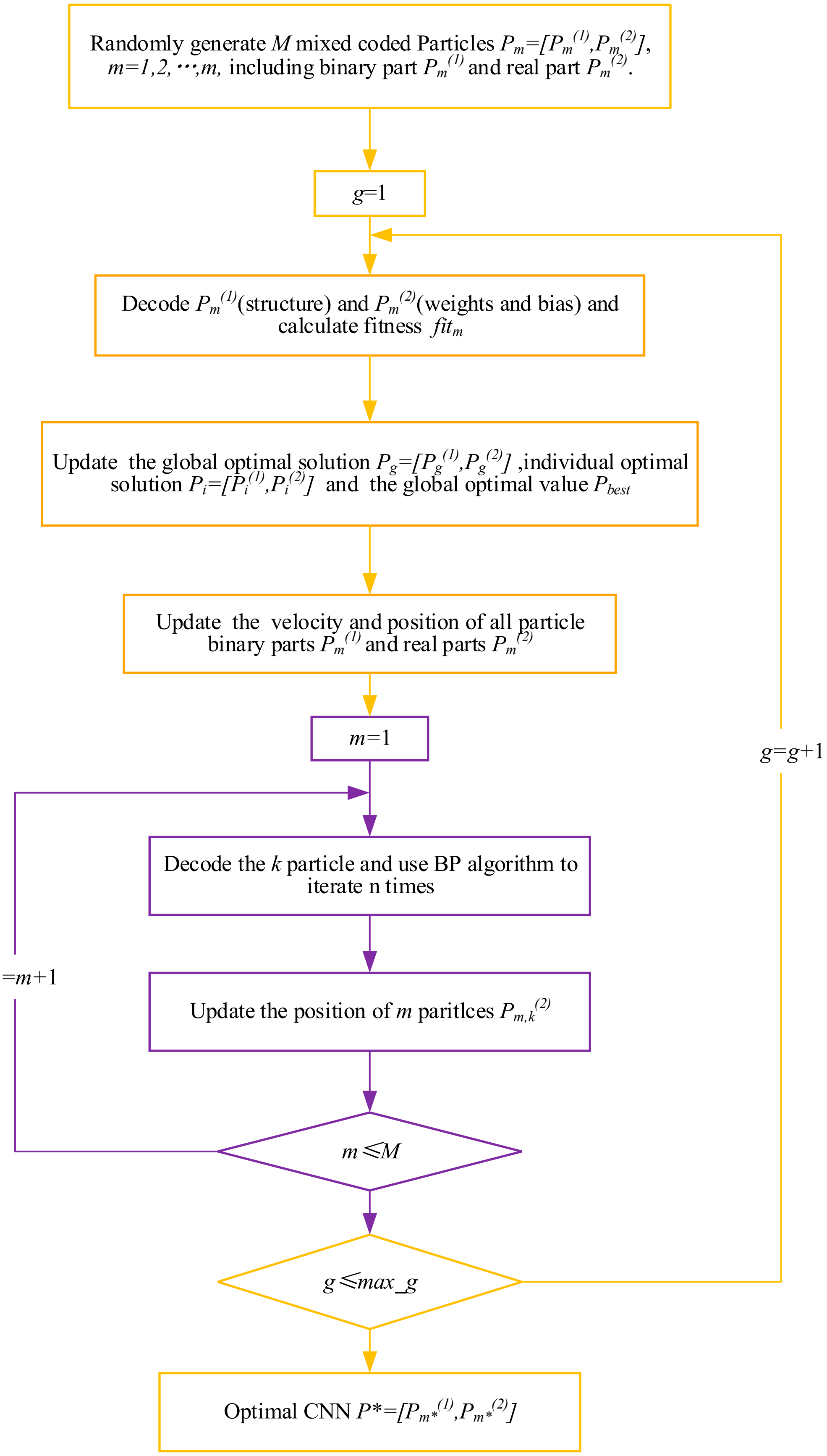

Randomly generate M mixed coded Particles $P_m=[P_m^{(1)},P_m^{(2)}]$, $m=1,2,\cdots,m$, including binary part $P_m^{(1)}$ and real part $P_m^{(2)}$.
g=1
Decode $P_m^{(1)}$(structure) and $P_m^{(2)}$(weights and bias) and calculate fitness fit_m
Update the global optimal solution $P_g=[P_g^{(1)},P_g^{(2)}]$,individual optimal solution $P_i=[P_i^{(1)},P_i^{(2)}]$ and the global optimal value P_{best}
Update the velocity and position of all particle binary parts $P_m^{(1)}$ and real parts $P_m^{(2)}$
m=1
Decode the k particle and use BP algorithm to iterate n times
Update the position of m paritlces $P_{m,k}^{(2)}$
$m \leqslant M$
m=m+1
$g \leqslant max\_g$
g=g+1
Optimal CNN $P^*=[P_{m^*}^{(1)},P_{m^*}^{(2)}]$

Figure 5. GA–PSO–CNN flowchart.

Step 1: Initialize the inertia factor, the learning factor, the mutation probability (α) of the binary part and the real part, the maximum velocity of the particle ($v_{\max}$), and the maximum number of iterations (max_g). According to the coding scheme, the hyperparameters use binary coding to control the number of convolutional layers, the kernel size, and the number of filters; the connection weights use real number coding. There is a one-to-one correspondence between the connection weights of the network nodes. Randomly generate M initial particles.

Step 2: Initialization: $g = 1$.

Step 3: Calculate the fitness function value of each particle according to Equation (7).

Step 4: Update the current optimal solution $P_g = [P_g^{(1)}, P_g^{(2)}]$, the individual optimal solution $P_i = [P_i^{(1)}, P_i^{(2)}]$, and the global optimal value P_{best}.

Step 5: Update the particle velocities using Equation (8). The position of the binary part of the particle is updated according to Equations (9) and (10), and the position of the real part of the particle is updated according to Equation (11). The mutation operation of the GA algorithm is added in the process of updating the binary part.

$$v_m(t+1) = \omega v_m(t) + c_1 r_1 (p_{best} - x_m) + c_2 r_2 (g_{best} - x_m) \tag{8}$$

$$s(v_m) = \frac{1}{1 + \exp(-v_m)} \tag{9}$$

$$x_m = \begin{cases} x_m,\ if\ rand() < s(v_m) \\ 1 - x_m,\ otherwise \end{cases} \tag{10}$$

$$x_{m,t,j} = \begin{cases} x_{m,t,j} + \rho_{[-0.5,0.5]} u_j & \text{if } x_{m,t,j} = g_j \text{ or } \rho \in [0, C_r) \\ g_j + \rho_{[-0.5,0.5]} u_j & \text{if } x_{m,t,j} \neq g_j \text{ or } \rho \in [C_r, C_g) \\ x_{m,t,j} + \rho_{[-0.5,0.5]} (x_{m,t,j} - g_j) & \text{if } x_{m,t,j} \neq g_j \text{ or } \rho \in [C_g, 1] \end{cases} \tag{11}$$

$$u_j = \frac{L_j - U_j}{2 \times N_{Var}} \tag{12}$$

where V_m represents the velocity of particle m, r_1 and r_2 are random values in the interval [0, 1]; p_{best} is the individual optimal position reached by the particle; g_{best} is the global optimal position; s is the probability of changing the position of the particle; $x_{m,t,j}$ represents the value of the jth site of particle m in the jth iteration; ρ_I is the random number generated within the interval I; g_j represents the value of the jth site of the global optimal particle; C_r and C_g are given constants; u_j is speed parameter; U_j and L_j are the upper and lower bounds of the variable, respectively; N_{Var} is the number of variables.

Step 6: $m = 1$.

Step 7: Perform n BP iterations on particle m. Update the value of the corresponding position in the real part of particle m according to the connection weight after iteration.

Step 8: If $m \leq M$, then $m = m + 1$. Go to Step 7. If not, go to Step 9.

Step 9: If $g \leq$ max_g, then $g = g + 1$. Go to Step 3. If not, go to Step 10.

Step 10: The algorithm is finished, and the optimal parameter combination $P^* = [P_{m^*}^{(1)}, P_{m^*}^{(2)}]$ of the CNN is obtained.

4. Experimental Study

4.1. Data Description and Preprocessing

A wind farm in Ningxia provided the study with real data of their operation in 2012 to test the performance of the model. The wind farm was located at the junction of Lingwu County of Yinchuan City and Yanchi County of Wuzhong City, with an average altitude of 1350 m. The vegetation at the wind farm was sparse; the terrain was open, flat, and slightly undulating. The wind direction was stable, and the wind energy was concentrated and rich. The wind farm area was approximately 111 km^2, equipped with a 200 kV wind farm booster station, a main transformer with a 150 MVA capacity, and approximately 200 sets of single-unit wind turbines with a capacity of 1500 kW. The total construction capacity was 300 MW, which will be completed in 4 phases. As the output power of a wind farm was affected mainly by the surrounding environment, this study considered wind speed, wind direction, air pressure, temperature, density at hub height, and historical wind power as input features. The variables are presented in Table 1.

Table 1. Data features used in the experiment.

Feature	Description	Unit
Wind speed	Measured by a digital anemometer outside the wind farm	m/s
Wind direction	The deviation angle between the wind direction and the center line room	°
Temperature	The temperature near the wind farm	K
Air pressure	The air pressure near the wind farm	Pa
Density at hub height	The Density at hub height measured by a density measuring instrument	kg/m^3
Power	Power generation capacity before the 15 min	MW

The data used in this study was from approximately 45 wind turbines in the first phase of the wind farm project, in 2 parts. The first part was the actual output wind power from January 1 to 31 December 2012. The second part was the meteorological wind speed, wind direction, air temperature, pressure, and density at hub height data from the wind farm in the same period. The sampling interval was 15 min; there were a total of 35,040 groups. There were a few missing data values. Lagrange interpolation was used to complete the data.

The data set was divided into 4 seasons: winter (December–February), spring (March–May), summer (June–August), and autumn (September–November). The first 70% of the data were used as the training set; the remaining 30% was the test set.

4.2. Time-Order Character

There is a time correlation in wind power data. In this section, the time-order character was established according to the correlation and used as input to predict the output of the next time point. The sliding window method was introduced to predict the output at each time point and compare it with the actual value. The wind speed, wind direction, atmospheric pressure, temperature, hub height density, and historical wind power output at the previous 4 time points were selected to form an input matrix x_1 to predict the wind power output power y_1 at the next time point as shown in Figure 6. Similarly, all characteristic variable values from the second to the fifth time points were selected to form an input matrix x_2 to predict the wind power output y_2 at the sixth time point, etc.

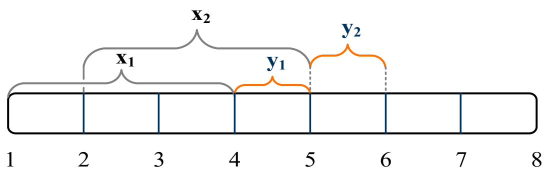
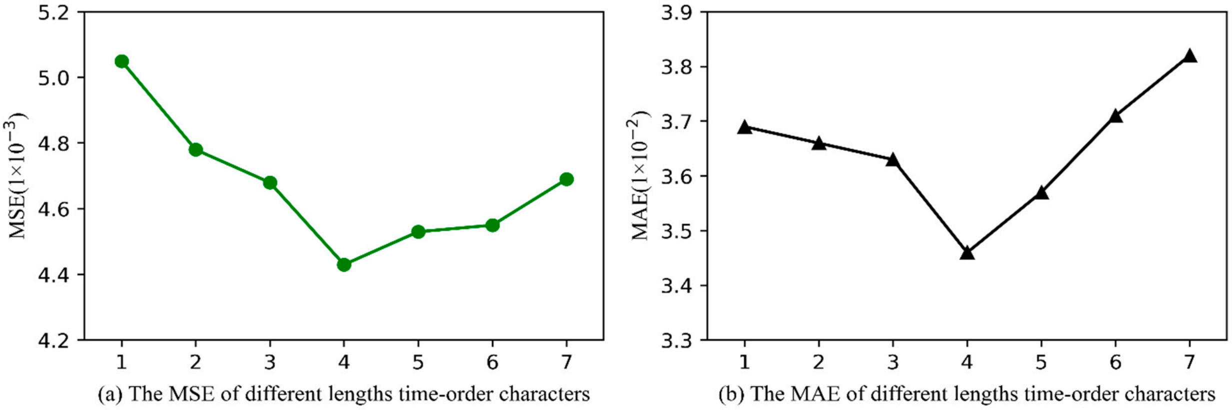


Figure 6. Structure of time-order character.

By constructing time-order characters, the CNN can better learn the inherent features in wind power data. However, when the time-order characters are too long, the model extracts too many redundant features, which reduces the prediction accuracy of the model and increases the time cost of training the model. Thus, time-order characters of different lengths were used as input to forecast the output of wind power. Figure 7 shows the MSE and MAE obtained by the CNN prediction model with time-order characters of different lengths used as input. The horizontal axis represents the length of the time-order characters; the vertical axis represents the error value. In Figure 7, the first half of the error curve showed a downward trend, and the second half showed an upward trend. When the time-order character length was 4, MSE and MAE simultaneously reached the minimum value. Thus, a time-order character length of 4 was chosen in this study.



(a) The MSE of different lengths time-order characters

(b) The MAE of different lengths time-order characters

Figure 7. Error curve of time-order characters with different lengths. (**a**,**b**) are the MSE and MAE of different lengths time-order characters, respectively.

4.3. Performance Testing

Table 2 the research was conducted using Python3.5 and Tensorflow1.4.0, on a personal computer with an i5-10300H CPU for calculations. To demonstrate the superior performance of the new CNN framework, 3 performance indicators were introduced to evaluate the prediction ability of the model: mean absolute error (MAE), mean square error (MSE) and mean absolute percentage error (MAPE).

Table 2. Parameter values for different models.

Model	Parameter	Description	Value
Single-CNN	c	The number of convolution layers	2
	k1 k2	The length of the convolution window	4/5
	f1 f2	The convolution kernel number of convolution layers	5/5
	epochs	The maximum number of iterations in the network	200
PSO-CNN	n	The number of particles in a particle swarm	20
	epochs1	The maximum number of iterations of PSO	20
	epochs2	The number of BP runs per iteration	10
	r	The ratio of BP in each generation of particles	0.3
ISSO-CNN	n	The number of particles in a particle swarm	20
	epochs1	The maximum number of iterations of PSO	20
	epochs2	The number of BP runs per iteration	10
	Cr Cv Cg	The parameters of the iterative formula	0.35/0.45/0.2
	r	The ratio of BP in each generation of particles	0.3
CHACNN	kernel	The interval of convolution kernel size	[2, 6]
	filters	The interval of filters in the convolution layers	[2, 8]
	alpha	The interval of learning rate in the network	[0.05, 2]
	epochs1	The maximum number of iterations of the algorithm	20
	epochs2	The interval of maximum BP iterations	[100, 250]
GA-PSO-CNN	n	The number of particles in a particle swarm	20
	epochs1	The maximum number of iterations of GA-PSO	20
	epochs2	The number of BP runs per iteration	50
	c1	The number of convolution layers	[1, 2]
	k1, k2	The interval of convolution kernel size	[2, 6]
	f1, f2	The interval of filters in the convolution layers	[2, 8]
	weights	The weights of convolutional neural networks	[−1, 1]
	bias	The bias of convolutional neural networks	[−1, 1]

4.4. Analysis of Results

To further verify the performance of the proposed model, each model was compared with GA–PSO–CNN, using MAE, MSE, and MAPE to measure its prediction accuracy. To ensure the validity of the evaluation criteria, the input of all models in the experiment was the same. As the experimental results have a certain degree of randomness, each model was trained 50 times to prevent accidental results in a single operation. The average of the error indicators for 50 operations was calculated for comparison to ensure the fairness of the results. Figure 8 shows the distribution of each performance index obtained from 50 independent operations of each model. It was observed that in each season, the GA–PSO–CNN model had a higher prediction accuracy and a smaller fluctuation interval in each performance index than the comparison models, which verified the excellent prediction performance and robustness of this model.

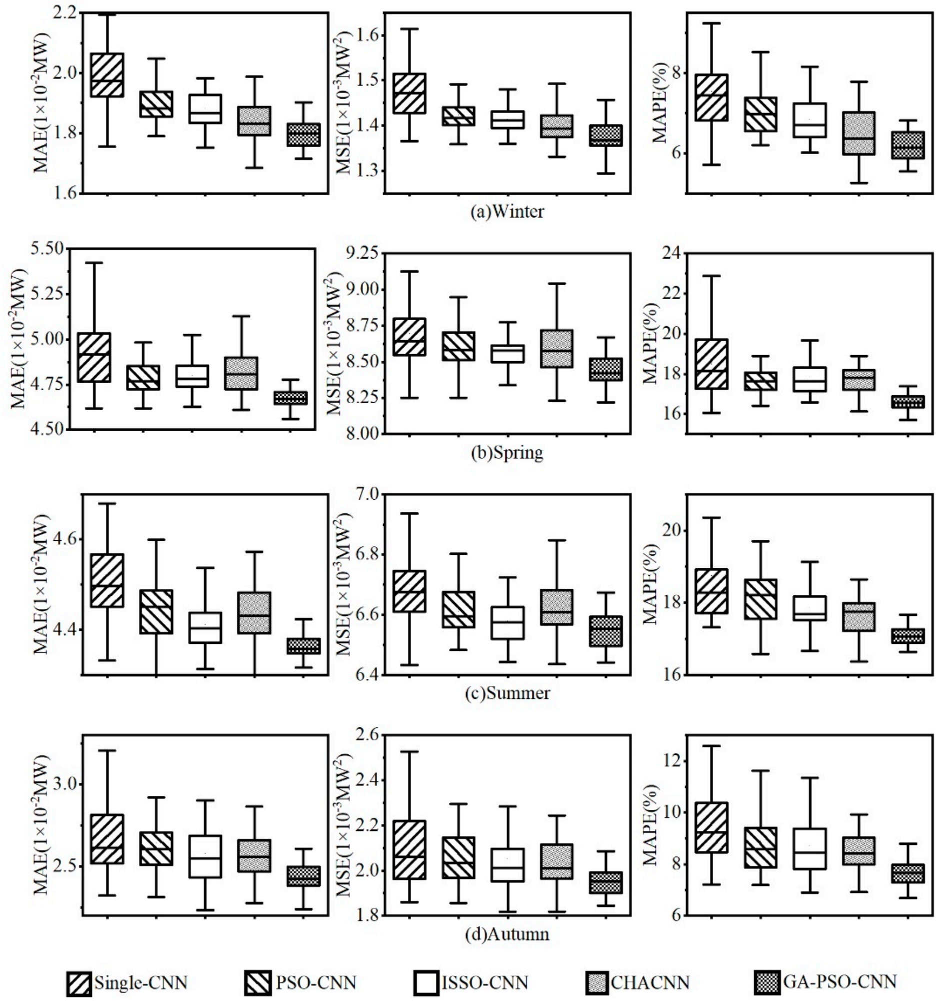


Figure 8. Performance metrics of models presented in box and whisker plots. (**a**–**d**) are the MAE, MSE and MAPE of the models in winter, spring, summer and autumn respectively.

Table 3 shows the average MAE, MSE, and MAPE values for different models in each season; the optimal value of each index is indicated in bold. The GA–PSO–CNN model demonstrates good wind power prediction performance in different seasons. In terms of MAE, MSE, and MAPE, the GA–PSO–CNN model had the best performance in all seasons, indicating that it was superior to the Single-CNN, PSO-CNN, ISSO-CNN, and CHACNN models in predicting ultra-short-term wind power. Compared with the benchmark model (Single-CNN), the GA–PSO–CNN model had the most obvious improvement in MAE, MSE, and MAPE performance indexes in different seasons, demonstrating its superior performance in ultra-short-term wind power forecasting.

Table 3. Wind power forecast error for each model in different seasons.

Season	Error	GA−PSO−CNN	Single−CNN	PSO−CNN	ISSO−CNN	CHACNN
Winter	MAE (MW)	1.80×10^{-2}	1.99×10^{-2} (−9.55%)	1.90×10^{-2} (−5.26%)	1.88×10^{-2} (−4.26%)	1.84×10^{-2} (−2.17%)
	MSE (MW2)	1.37×10^{-3}	1.48×10^{-3} (−7.43%)	1.42×10^{-3} (−3.52%)	1.41×10^{-3} (−2.84%)	1.40×10^{-3} (−2.14%)
	MAPE (%)	6.17	7.54 (−18.17%)	7.07 (−12.73%)	6.84 (−9.80%)	6.49 (−4.93%)
Spring	MAE (MW)	4.67×10^{-2}	4.95×10^{-2} (−5.66%)	4.79×10^{-2} (−2.51%)	4.80×10^{-2} (−2.71%)	4.82×10^{-2} (−3.11%)
	MSE (MW2)	8.45×10^{-3}	8.70×10^{-3} (−2.87%)	8.60×10^{-3} (−1.74%)	8.57×10^{-3} (−1.40%)	8.61×10^{-3} (−1.86%)
	MAPE (%)	16.61	18.5 (−10.22%)	17.72 (−6.26%)	17.87 (−7.05%)	17.7 (−6.16%)
Summer	MAE (MW)	4.36×10^{-2}	4.54×10^{-2} (−3.96%)	4.45×10^{-2} (−2.02%)	4.41×10^{-2} (−1.13%)	4.43×10^{-2} (−1.58%)
	MSE (MW2)	6.55×10^{-3}	6.70×10^{-3} (−2.24%)	6.61×10^{-3} (−0.91%)	6.58×10^{-3} (−0.46%)	6.64×10^{-3} (−1.36%)
	MAPE (%)	17.1	18.75 (−8.80%)	18.23 (−6.20%)	17.85 (−4.20%)	17.68 (−3.28%)
Autumn	MAE (MW)	2.44×10^{-2}	2.67×10^{-2} (−8.61%)	2.64×10^{-2} (−7.58%)	2.58×10^{-2} (−5.43%)	2.53×10^{-2} (−3.56%)
	MSE (MW2)	1.96×10^{-3}	2.13×10^{-3} (−7.98%)	2.08×10^{-3} (−5.77%)	2.05×10^{-3} (−4.39%)	2.04×10^{-3} (−3.92%)
	MAPE (%)	7.7	9.54 (−19.29%)	8.89 (−13.39%)	8.71 (−11.60%)	8.53 (−9.73%)
Average	MAE (MW)	3.32×10^{-2}	3.54×10^{-2} (−6.21%)	3.45×10^{-2} (−3.77%)	3.42×10^{-2} (−2.92%)	3.41×10^{-2} (−2.64%)
	MSE (MW2)	4.58×10^{-3}	4.75×10^{-3} (−3.58%)	4.68×10^{-3} (−2.14%)	4.65×10^{-3} (−1.51%)	4.67×10^{-3} (−1.93%)
	MAPE (%)	11.9	13.58 (−12.37%)	12.98 (−8.32%)	12.82 (−7.18%)	12.6 (−5.56%)

Due to the large climatic differences in different seasons, 3 days were randomly selected from different seasons to draw the error box plot between the predicted power of the GA–PSO–CNN model and the actual wind power, as shown in Figure 9. It was observed that in autumn and winter, the uncertainty of wind power prediction was relatively low; the error between the predicted value and the actual value was generally maintained below 10%. However, the uncertainty of wind power forecasting was greater in summer; the error was controlled within 20%. The climate conditions in summer were more complex and rainy and were not conducive to wind power prediction.

To further analyze the fitting effect of GA–PSO–CNN on wind power in each season, 1 day was selected from each season to draw a comparison diagram between the predicted value of GA–PSO–CNN and the actual value, as shown in Figure 10. GA–PSO–CNN exhibits good prediction performance in all seasons, and the prediction results were close to the actual values. When the climate conditions were complex, and the wind speed changed greatly and rapidly (as in summer), the error indexes of the prediction models were relatively high; the GA–PSO–CNN model produced the smallest error. In Figure 10, even with several wind power changes in a short period of time in summer, the GA–PSO–CNN model extracts relevant features. The prediction curve was close to the actual value curve, indicating a good prediction.

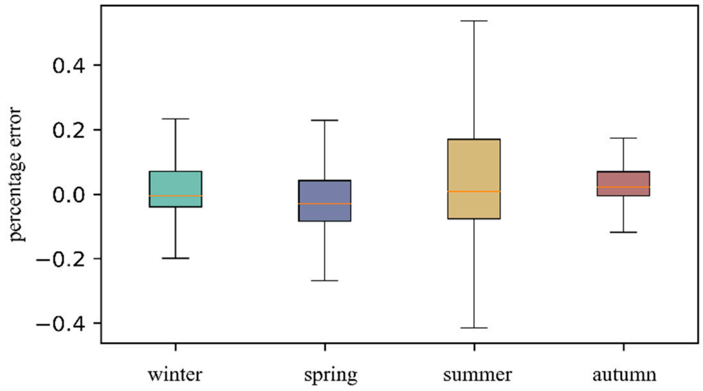
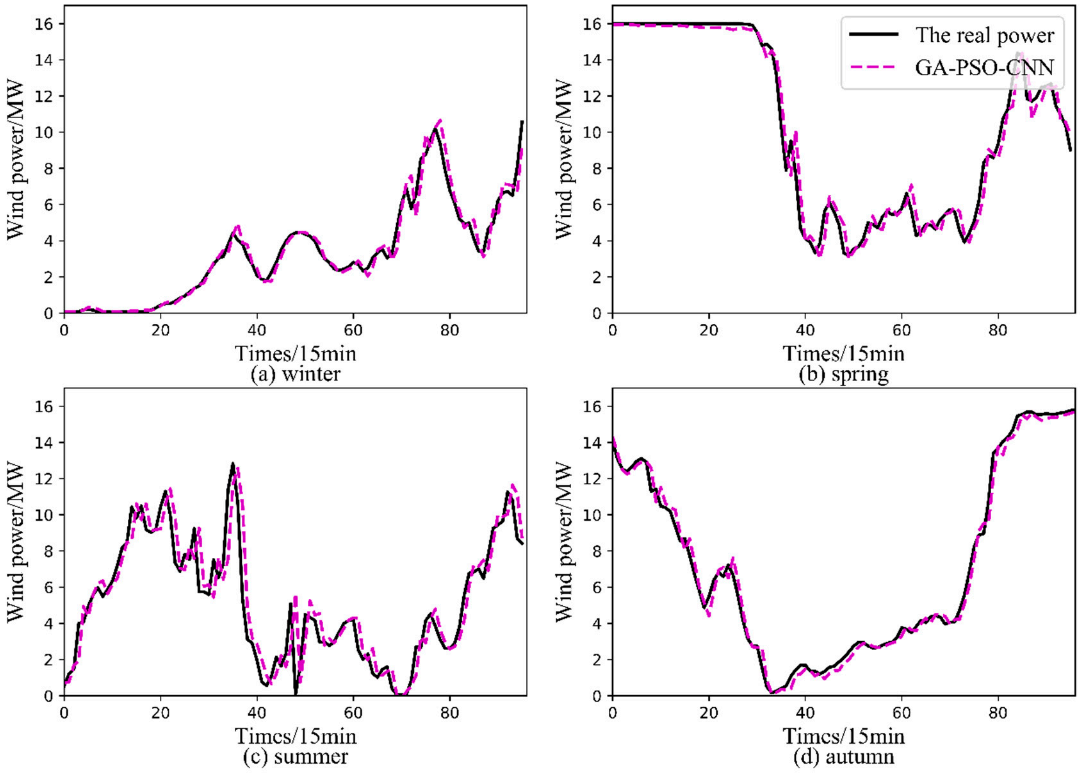


Figure 9. GA–PSO–CNN forecast error in different seasons.



(a) winter
(b) spring
(c) summer
(d) autumn

Figure 10. Wind power fitting curves for GA–PSO–CNN model in different seasons. (**a**–**d**) are wind power fitting curves for the model in winter, spring, summer and autumn respectively.

The formation of wind was closely related to air pressure; the weather conditions vary greatly in the morning and evening, as does the wind force. To further verify the wind power prediction accuracy improvements of the GA–PSO–CNN model, a month was randomly selected from the data set. The day was divided into day (8:00–22:00) and night (22:00–8:00 the next day) for prediction. Table 4 shows the prediction errors of each model in different time periods.

Table 4. Prediction error of each model in different time periods.

Time	Error	GA-PSO-CNN	Single-CNN	PSO-CNN	ISSO-CNN	CHACNN
Daytime (8:00–22:00)	MAE (MW)	1.77×10^{-2}	2.16×10^{-2} (−18.06%)	1.91×10^{-2} (−7.33%)	1.92×10^{-2} (−7.81%)	1.85×10^{-2} (−4.32%)
	MSE (MW^2)	1.54×10^{-3}	1.65×10^{-3} (−6.67%)	1.58×10^{-3} (−2.53%)	1.63×10^{-3} (−5.52%)	1.60×10^{-3} (−3.75%)
	MAPE (%)	6.23	9.91 (−37.13%)	7.39 (−15.70%)	7.32 (−14.89%)	6.79 (−8.25%)
Nighttime (22:00–8:00 next day)	MAE (MW)	1.69×10^{-2}	1.97×10^{-2} (−14.21%)	1.80×10^{-2} (−6.11%)	1.79×10^{-2} (−5.59%)	1.71×10^{-2} (−1.17%)
	MSE (MW^2)	1.05×10^{-3}	1.18×10^{-3} (−11.02%)	1.09×10^{-3} (−3.67%)	1.07×10^{-3} (−1.87%)	1.12×10^{-3} (−6.25%)
	MAPE (%)	4.89	6.99 (−30.04%)	5.51 (−11.25%)	5.58 (−12.37%)	4.84 (+2.30%)

During the day, the average power of the wind farm was 6.16 MW every 15 min; at night, the average power was 7.88 MW. As observed in Table 4, the influence of each disturbance factor was smaller at night, which was more conducive to wind power prediction. The prediction error of each model was significantly smaller at night than in the day. The GA–PSO–CNN model was only slightly inferior to the CHACNN model in MAPE index at night (2.30% difference); in most other cases, the GA–PSO–CNN model achieved the best results. The GA–PSO–CNN model was superior to Single-CNN, PSO-CNN, ISSO-CNN, and CHACNN models in wind power prediction in different time periods.

To study the wind power prediction performance of the model in different weather conditions, based on the local weather forecast, 3 days were selected in each season (sunny, cloudy, and rainy days) to compare the performance of the models. Table 5 shows the performance evaluation results for each model. The forecast accuracy on rainy days is lower than on sunny days for all models in all seasons. On rainy days, the meteorological conditions were chaotic, and there were many interference factors. Wind power was sensitive to weather conditions; accurate prediction was difficult on rainy days. Improving the prediction accuracy of wind power on rainy days has been a research focus in recent years. In Table 5, it is observed that the GA–PSO–CNN model exhibits the best performance of all prediction models, with the lowest MAE, MSE, and MAPE on most days. The error range of the GA–PSO–CNN model was the smallest of the models, even on cloudy and rainy days. The results further demonstrate the stability and effectiveness of the GA–PSO–CNN model.

The GA–PSO–CNN model was run 50 times independently on the data set to decode the optimal particles, and the size and number of convolution kernels in each layer were calculated, as shown in Table 6. Compared with the structural parameters selected by manual parameter tuning, the GA–PSO–CNN model has a simplified network structure. For the average value in 50 runs, the size and number of convolution kernels in each layer decreased by 5–18.4%, indicating that the GA–PSO–CNN model has a stronger ability to resist overfitting and better generalization.

The GA–PSO–CNN model can accurately predict wind power in different seasons, time periods, weather, and other conditions. Compared with other models, the GA–PSO–CNN model has improved prediction accuracy and robustness. This model can be applied to ultra-short-term wind power prediction in different scenarios.

Table 5. Wind power prediction errors of each model on a certain day.

Season	Type of Weather	Error	GA-PSO-CNN	Single-CNN	PSO-CNN	ISSO-CNN	CHACNN
Winter	Sunny (day 1)	MAE (MW)	1.05×10^{-2}	1.18×10^{-2} (−11.02%)	1.11×10^{-2} (−5.41%)	1.17×10^{-2} (−10.26%)	1.18×10^{-2} (−11.02%)
		MSE (MW^2)	3.70×10^{-4}	4.28×10^{-4} (−13.55%)	4.29×10^{-4} (−13.75%)	4.12×10^{-4} (−10.19%)	4.24×10^{-4} (−12.74%)
		MAPE (%)	1.49	1.63 (−8.59%)	1.54 (−3.25%)	1.59 (−6.29%)	1.63 (−8.59%)
	Cloudy (day 2)	MAE (MW)	1.13×10^{-2}	1.52×10^{-2} (−25.66%)	1.34×10^{-2} (−15.67%)	1.42×10^{-2} (−20.42%)	1.18×10^{-2} (−4.24%)
		MSE (MW^2)	4.60×10^{-4}	5.56×10^{-4} (−17.27%)	5.47×10^{-4} (−15.90%)	4.75×10^{-4} (−3.16%)	4.67×10^{-4} (−1.50%)
		MAPE (%)	4.03	6.36 (−36.64%)	5.39 (−25.23%)	5.60 (−28.04%)	3.89 (+3.60%)
	Rainy (day 3)	MAE (MW)	1.30×10^{-2}	1.50×10^{-2} (−13.33%)	1.45×10^{-2} (−10.34%)	1.36×10^{-2} (−4.41%)	1.42×10^{-2} (−8.45%)
		MSE (MW^2)	4.30×10^{-4}	4.48×10^{-4} (−4.02%)	4.58×10^{-4} (−6.11%)	4.26×10^{-4} (+0.94%)	4.82×10^{-4} (−10.79%)
		MAPE (%)	7.12	9.21 (−22.69%)	9.00 (−20.89%)	7.39 (−3.65%)	7.63 (−6.68%)
Spring	Sunny (day 4)	MAE (MW)	2.84×10^{-2}	3.52×10^{-2} (−19.32%)	3.31×10^{-2} (−14.20%)	3.05×10^{-2} (−6.89%)	3.15×10^{-2} (−9.84%)
		MSE (MW^2)	2.65×10^{-3}	3.42×10^{-3} (−22.51%)	3.08×10^{-3} (−13.96%)	3.17×10^{-3} (−16.40%)	3.24×10^{-3} (−18.21%)
		MAPE (%)	8.04	11.30 (−28.85%)	9.03 (−10.96%)	8.58 (−6.29%)	9.35 (−14.01%)
	Cloudy (day 5)	MAE (MW)	4.63×10^{-2}	5.16×10^{-2} (−10.27%)	5.05×10^{-2} (−8.32%)	5.07×10^{-2} (−8.68%)	4.77×10^{-2} (−2.94%)
		MSE (MW^2)	4.68×10^{-3}	5.93×10^{-3} (−21.08%)	5.70×10^{-3} (−17.89%)	5.55×10^{-3} (−15.68%)	4.99×10^{-3} (−6.21%)
		MAPE (%)	11.78	13.26 (−11.16%)	12.66 (−6.95%)	12.90 (−8.68%)	12.63 (−6.73%)
	Rainy (day 6)	MAE(MW)	8.74×10^{-2}	8.90×10^{-2} (−1.80%)	8.77×10^{-2} (−0.34%)	8.46×10^{-2} (+3.31%)	8.54×10^{-2} (+2.34%)
		MSE (MW^2)	1.92×10^{-2}	2.17×10^{-2} (−11.52%)	2.01×10^{-2} (−4.48%)	1.99×10^{-2} (−3.52%)	1.97×10^{-2} (−2.54%)
		MAPE (%)	19.09	21.69 (−11.99%)	21.10 (−9.53%)	20.52 (−6.97%)	19.48 (−2.00%)
Summer	Sunny (day 7)	MAE (MW)	4.33×10^{-2}	4.88×10^{-2} (−11.27%)	4.71×10^{-2} (−8.07%)	4.66×10^{-2} (−7.08%)	4.81×10^{-2} (−9.98%)
		MSE (MW^2)	4.21×10^{-3}	5.37×10^{-3} (−21.60%)	4.95×10^{-3} (−14.95%)	5.17×10^{-3} (−18.57%)	4.83×10^{-3} (−12.84%)
		MAPE (%)	6.28	7.51 (−16.38%)	6.73 (−6.69%)	6.64 (−5.42%)	6.61 (−4.99%)
	Cloudy (day 8)	MAE (MW)	4.01×10^{-2}	5.45×10^{-2} (−26.42%)	4.80×10^{-2} (−16.46%)	5.02×10^{-2} (−20.12%)	5.11×10^{-2} (−21.53%)
		MSE (MW^2)	7.80×10^{-3}	1.30×10^{-2} (−40.00%)	1.08×10^{-2} (−27.78%)	1.14×10^{-2} (−31.58%)	9.17×10^{-3} (−14.94%)
		MAPE (%)	12.04	17.57 (−31.47%)	16.07 (−25.08%)	15.68 (−23.21%)	14.19 (−15.15%)
	Rainy (day 9)	MAE (MW)	5.52×10^{-2}	6.21×10^{-2} (−11.11%)	5.69×10^{-2} (−2.99%)	5.83×10^{-2} (−5.32%)	5.70×10^{-2} (−3.16%)
		MSE (MW^2)	6.59×10^{-3}	7.04×10^{-3} (−6.39%)	6.83×10^{-3} (−3.51%)	6.79×10^{-3} (−2.95%)	6.72×10^{-3} (−1.93%)
		MAPE (%)	22.53	24.69 (−8.75%)	23.12 (−2.55%)	23.64 (−4.70%)	22.01 (+2.36%)

Table 5. *Cont.*

Season	Type of Weather	Error	GA-PSO-CNN	Single-CNN	PSO-CNN	ISSO-CNN	CHACNN
Autumn	Sunny (day 10)	MAE (MW)	2.65×10^{-2}	3.95×10^{-2} (−32.91%)	3.30×10^{-2} (−19.70%)	3.89×10^{-2} (−31.88%)	3.51×10^{-2} (−24.50%)
		MSE (MW^2)	4.04×10^{-3}	6.24×10^{-3} (−35.26%)	5.11×10^{-3} (−20.94%)	5.55×10^{-3} (−27.21%)	5.06×10^{-3} (−20.16%)
		MAPE (%)	3.66	5.84 (−37.33%)	4.51 (−18.85%)	5.22 (−29.89%)	4.62 (−20.78%)
	Cloudy (day 11)	MAE (MW)	2.11×10^{-2}	2.37×10^{-2} (−10.97%)	2.21×10^{-2} (−4.52%)	2.27×10^{-2} (−7.05%)	2.20×10^{-2} (−4.09%)
		MSE (MW^2)	7.89×10^{-4}	1.19×10^{-3} (−33.70%)	9.94×10^{-4} (−20.62%)	1.02×10^{-3} (−22.65%)	8.79×10^{-4} (−10.24%)
		MAPE (%)	6.31	7.21 (−12.48%)	6.81 (−7.34%)	6.56 (−3.81%)	6.75 (−6.52%)
	Rainy (day 12)	MAE (MW)	4.56×10^{-2}	5.28×10^{-2} (−13.64%)	4.91×10^{-2} (−7.13%)	4.64×10^{-2} (−1.72%)	4.63×10^{-2} (−1.51%)
		MSE (MW^2)	7.31×10^{-3}	9.88×10^{-3} (−26.01%)	9.21×10^{-3} (−20.63%)	8.48×10^{-3} (−13.80%)	8.05×10^{-3} (−9.19%)
		MAPE (%)	12.62	14.69 (−14.09%)	13.28 (−4.97%)	12.27 (+2.85%)	13.04 (−3.22%)

Table 6. Optimal structural parameter distribution of 50 independent operations in GA–PSO–CNN model.

Hyperparameter	Average	The Value of Manual Tuning	Reduced Proportion
kernel_size1	3.8	4	5%
filters1	4.56	5	8.8%
kernel_size2	4.08	5	18.4%
filters2	4.52	5	9.6%

5. Conclusions

To solve the problem of wind power prediction caused by randomness and volatility, a hybrid GA-PSO-CNN prediction model is proposed in this study. The model uses CNN to extract and fit the high-dimensional features of wind farm data, integrates the intelligent algorithm GA and PSO, and optimizes the structure and parameters of CNN to obtain the optimal network structure and connection weight combination simultaneously.

The proposed GA-PSO-CNN model is applied to the power generation data of a wind farm in Ningxia. The experimental results demonstrate the improvements of the GA-PSO-CNN model in prediction accuracy and network structure simplicity. Compared with Single-CNN, PSO-CNN, ISSO-CNN, and CHACNN models, the GA–PSO–CNN model reduced the MAE, MSE, and MAPE by 1.13–9.55%, 0.46–7.98%, and 3.28–19.29%, respectively, with data sets from different seasons. The GA–PSO–CNN model has a more simplified network structure than with hyperparameters selected by manual tuning, reducing the size and number of convolution kernels by 5–18.4%.

Author Contributions: Conceptualization, J.L. and J.Y.; formal analysis, J.Y. and R.H.; data curation, J.L.; methodology, Q.S. and J.L.; software, Q.S. and R.H.; validation, J.Y. and Q.S.; writing—original draft preparation, Q.S. and J.L.; writing—review and editing, R.H. and J.Y.; visualization, Q.S. and R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant numbers: 71822403, 31961143006, and 71573236) and the Hubei Natural Science Foundation, China (Grant number: 2019CFA089).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Mahmoudan, A.; Samadof, P.; Hosseinzadeh, S.; Garcia, D.A. A multigeneration cascade system using ground-source energy with cold recovery: 3E analyses and multi-objective optimization. *Energy* **2021**, *233*, 121185. [CrossRef]
2. Makkiabadi, M.; Hoseinzadeh, S.; Mohammadi, M.; Nowdeh, S.A.; Bayati, S.; Jafaraghaei, U.; Mirkiai, S.M.; Assad, M.E.H. Energy Feasibility of Hybrid PV/Wind Systems with Electricity Generation Assessment under Iran Environment. *Appl. Sol. Energy* **2020**, *56*, 517–525. [CrossRef]
3. Hoseinzadeh, S.; Ghasemi, M.H.; Heyns, S. Application of hybrid systems in solution of low power generation at hot seasons for micro hydro systems. *Renew. Energy* **2020**, *160*, 323–332. [CrossRef]
4. Hoseinzadeh, S.; Sohani, A.; Heyns, S. Comprehensive analysis of the effect of air injection on the wake development of an airfoil. *Ocean Eng.* **2021**, *220*, 108455. [CrossRef]
5. Petroleum, B. Statistics. 2020. Available online: https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy.html (accessed on 17 June 2020).
6. Hoseinzadeh, S.; Bahrami, A.; Mirhosseini, S.M.; Sohani, A.; Heyns, S. A detailed experimental airfoil performance investigation using an equipped wind tunnel. *Flow Meas. Instrum.* **2020**, *72*, 101717. [CrossRef]
7. Lv, W.; Ma, J.; Chen, J.; Wu, Y. The development status and bottleneck of wind power industry. *Renew. Energy Resour.* **2018**, *36*, 1214–1218. [CrossRef]
8. Xu, G.; Ye, Z.; Xie, H. The Current State and Future Development of Wind Turbine Technology. *Strateg. Study Chin. Acad. Eng.* **2018**, *20*, 44–50. [CrossRef]
9. Ayodele, T.R.; Jimoh, A.; Munda, J.L.; Agee, J.J.W. Challenges of grid integration of wind power on power system grid integrity: A review. *Int. J. Energy Res.* **2020**, *3*, 618–626.
10. Hou, M. Research status of ultra-short term wind power forecasting. *Electron. Technol. Softw. Eng.* **2019**, *19*, 210–212.
11. Damousis, I.G.; Alexiadis, M.C.; Theocharis, J.B.; Dokopoulos, P.S. A fuzzy model for wind speed prediction and power generation in wind parks using spatial correlation. *IEEE Trans. Energy Convers.* **2004**, *19*, 352–361. [CrossRef]
12. Zhang, W.; Qu, Z.; Zhang, K.; Mao, W.; Ma, Y.; Fan, X. A combined model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting. *Energ. Convers. Manag.* **2017**, *136*, 439–451. [CrossRef]
13. Li, L.-L.; Zhao, X.; Tseng, M.-L.; Tan, R.R. Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm. *J. Clean. Prod.* **2020**, *242*, 118447. [CrossRef]
14. Yatiyana, E.; Rajakaruna, S.; Ghosh, A. Wind Speed and Direction Forecasting for Wind Power Generation Using ARIMA Model. In Proceedings of the 2017 Australasian Universities Power Engineering Conference (AUPEC), Melbourne, Australia, 19–22 November 2017; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA; pp. 1–6. [CrossRef]
15. Aasim; Singh, S.N.; Mohapatra, A. Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting. *Renew. Energy* **2019**, *136*, 758–768. [CrossRef]
16. Li, L.; Liu, Y.-Q.; Yang, Y.-P.; Han, S.; Wang, Y.-M. A physical approach of the short-term wind power prediction based on CFD pre-calculated flow fields. *J. Hydrodyn. Ser. B* **2013**, *25*, 56–61. [CrossRef]
17. Cassola, F.; Burlando, M. Wind speed and wind energy forecast through Kalman filtering of Numerical Weather Prediction model output. *Appl. Energy* **2012**, *99*, 154–166. [CrossRef]
18. Wang, Z.; Zhang, J.; Zhang, Y.; Huang, C.; Wang, L. Short-Term Wind Speed Forecasting Based on Information of Neighboring Wind Farms. *IEEE Access* **2020**, *8*, 16760–16770. [CrossRef]
19. Chen, K.; Yu, J. Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach. *Appl. Energy* **2014**, *113*, 690–705. [CrossRef]
20. Milligan, M.; Schwartz, M.N.; Wan, Y.-H. Statistical Wind Power Forecasting for US Wind Farms. In Proceedings of the 17th Conference on Probability and Statistics in the Atmospheric Sciences, 2004/American Meteorological Society Annual Meeting, Seattle, WA, USA, 11–15 January 2004; National Renewable Energy Lab.: Golden, CO, USA, 2004.
21. Zhang, F.; Li, P.-C.; Gao, L.; Liu, Y.-Q.; Ren, X.-Y. Application of autoregressive dynamic adaptive (ARDA) model in real-time wind power forecasting. *Renew. Energy* **2021**, *169*, 129–143. [CrossRef]
22. Liu, F.; Li, R.; Dreglea, A. Wind Speed and Power Ultra Short-Term Robust Forecasting Based on Takagi–Sugeno Fuzzy Model. *Energies* **2019**, *12*, 3551. [CrossRef]
23. Liu, Z.; Jiang, P.; Zhang, L.; Niu, X. A combined forecasting model for time series: Application to short-term wind speed forecasting. *Appl. Energy* **2020**, *259*, 114137. [CrossRef]
24. Wang, S.; Wu, K.; Zhao, Q.; Wang, S.; Feng, L.; Zheng, Z.; Wang, G. Multienergy Load Forecasting for Regional Integrated Energy Systems Considering Multienergy Coupling of Variation Characteristic Curves. *Energy* **2021**, *9*. [CrossRef]
25. Han, S.; Li, J.; Liu, Y. Tabu search algorithm optimized ANN model for wind power prediction with NWP. *Energy Procedia* **2011**, *12*, 733–740. [CrossRef]

26. Han, S.; Yang, Y.; Liu, Y. The Comparison of BP Network and RBF Network in Wind Power Prediction Application. In Proceedings of the 2007 Second International Conference on Bio-Inspired Computing: Theories and Applications, Zhengzhou, China, 14–17 September 2007; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2007; pp. 173–176. [CrossRef]
27. Carolin Mabel, M.; Fernandez, E. Analysis of wind power generation and prediction using ANN: A case study. *Renew. Energy* **2008**, *33*, 986–992. [CrossRef]
28. Wang, J.; Zhang, W.; Li, Y.; Wang, J.; Dang, Z. Forecasting wind speed using empirical mode decomposition and Elman neural network. *Appl. Soft. Comput.* **2014**, *23*, 452–459. [CrossRef]
29. Ramirez-Rosado, I.J.; Fernandez-Jimenez, L.A.; Monteiro, C.; Sousa, J.; Bessa, R. Comparison of two new short-term wind-power forecasting systems. *Renew. Energy* **2009**, *34*, 1848–1854. [CrossRef]
30. Kaoshe, Z.; Zhao, L. Short-term wind power forecasting based on RBF-BP neural network. *Renew. Energy* **2014**, *32*, 1346–1351. [CrossRef]
31. Xu, L.; Mao, J. Short-Term Wind Power Forecasting Based on Elman Neural Network With Particle Swarm Optimization. In Proceedings of the 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, 28–30 May 2016; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2016; pp. 2678–2681. [CrossRef]
32. Hong, Y.-Y.; Rioflorido, C.L.P.P. A hybrid deep learning-based neural network for 24-h ahead wind power forecasting. *Appl. Energy* **2019**, *250*, 530–539. [CrossRef]
33. Du, P.; Wang, J.; Yang, W.; Niu, T. Multi-step ahead forecasting in electrical power system using a hybrid forecasting system. *Renew. Energy* **2018**, *122*, 533–550. [CrossRef]
34. Jiang, P.; Yang, H.; Heng, J. A hybrid forecasting system based on fuzzy time series and multi-objective optimization for wind speed forecasting. *Appl. Energy* **2019**, *235*, 786–801. [CrossRef]
35. Li, Y.; Shi, H.; Han, F.; Duan, Z.; Liu, H. Smart wind speed forecasting approach using various boosting algorithms, big multi-step forecasting strategy. *Renew. Energy* **2019**, *135*, 540–553. [CrossRef]
36. Li, Y.; Li, X.; Jia, X.; Ma, R.; Hui, D. Monitoring and Control for Hundreds Megawatt Scale Battery Energy Storage Station Based on Multi-Agent: Methodology and System Design. In Proceedings of the 2018 IEEE International Conference of Safety Produce Informatization (IICSPI), Chongqing, China, 10–12 December 2018; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2018; pp. 765–769. [CrossRef]
37. Li, X.; Wang, S. A review on energy management, operation control and application methods for grid battery energy storage systems. *CSEE Int. J. Electr. Power* **2019**, 1–15. [CrossRef]
38. Solas, M.; Cepeda, N.; Viegas, J.L. Convolutional Neural Network for Short-Term Wind Power Forecasting. In Proceedings of the 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), Bucharest, Romania, 29 September–2 October 2019; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2019; pp. 1–5. [CrossRef]
39. Wang, H.-Z.; Li, G.-Q.; Wang, G.-B.; Peng, J.-C.; Jiang, H.; Liu, Y.-T. Deep learning based ensemble approach for probabilistic wind power forecasting. *Appl. Energy* **2017**, *188*, 56–70. [CrossRef]
40. Liu, T.; Huang, Z.; Tian, L.; Zhu, Y.; Wang, H.; Feng, S. Enhancing Wind Turbine Power Forecast via Convolutional Neural Network. *Electronics* **2021**, *10*, 261. [CrossRef]
41. Ju, Y.; Sun, G.; Chen, Q.; Zhang, M.; Zhu, H.; Rehman, M.U. A Model Combining Convolutional Neural Network and LightGBM Algorithm for Ultra-Short-Term Wind Power Forecasting. *IEEE Access* **2019**, *7*, 28309–28318. [CrossRef]
42. Liu, H.; Mi, X.; Li, Y. Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network. *Energ. Convers. Manag.* **2018**, *166*, 120–131. [CrossRef]
43. Yildiz, C.; Acikgoz, H.; Korkmaz, D.; Budak, U. An improved residual-based convolutional neural network for very short-term wind power forecasting. *Energy Convers. Manag.* **2021**, *228*, 113731. [CrossRef]
44. Zhang, S.; Chen, Y.; Xiao, J.; Zhang, W.; Feng, R. Hybrid wind speed forecasting model based on multivariate data secondary decomposition approach and deep learning algorithm with attention mechanism. *Renew. Energy* **2021**, *174*, 688–704. [CrossRef]
45. Kuang, H.; Guo, Q.; Li, S.; Zhong, H. Short-Term Wind Power Forecasting Model Based on Multi-Feature Extraction and CNN-LSTM. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Shenzhen, China, 15–17 January 2021; IOP Publishing: Bristol, UK, 2021; Volume 702, p. 012019. [CrossRef]
46. Bochinski, E.; Senst, T.; Sikora, T. Hyper-Parameter Optimization for Convolutional Neural Network Committees Based on Evolutionary Algorithms. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2017; pp. 3924–3928. [CrossRef]
47. Oullette, R.; Browne, M.; Hirasawa, K. Genetic Algorithm Optimization of a Convolutional Neural Network for Autonomous Crack Detection. In Proceedings of the 2004 Congress on Evolutionary Computation, Portland, OR, USA, 19–23 June 2004; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2004; pp. 516–521. [CrossRef]
48. Fedorovici, L.-O.; Precup, R.-E.; Dragan, F.; Purcaru, C. Evolutionary Optimization-Based Training of Convolutional Neural Networks for OCR Applications. In Proceedings of the 2013 17th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 11–13 October 2013; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2013; pp. 207–212. [CrossRef]
49. Albeahdili, H.M.; Han, T.; Islam, N.E. Hybrid algorithm for the optimization of training convolutional neural network. *Int. J. Adv. Comput. Sci. Appl.* **2015**, *6*, 79–85.

50. Huang, C.-L.; Shih, Y.-C.; Lai, C.-M.; Chung, V.Y.Y.; Zhu, W.-B.; Yeh, W.-C.; He, X. Optimization of a Convolutional Neural Network Using a Hybrid Algorithm. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2019; pp. 1–8. [CrossRef]
51. Chen, G.; Shan, J.; Li, D.Y.; Wang, C.; Li, C.; Zhou, Z.; Wang, X.; Li, Z.; Hao, J.J. Research on Wind Power Prediction Method Based on Convolutional Neural Network and Genetic Algorithm. In Proceedings of the 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia), Chengdu, China, 21–24 May 2019; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2019; pp. 3573–3578. [CrossRef]
52. Dong, N.; Chang, J.-F.; Wu, A.-G.; Gao, Z.-K. A novel convolutional neural network framework based solar irradiance prediction method. *Int. J. Elec. Power* **2020**, *114*, 105411. [CrossRef]
53. Yang, M.; Wang, S. A Review of Wind Power Forecasting & Prediction. In Proceedings of the 2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), Beijing, China, 16–20 October 2016; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2016; pp. 1–7. [CrossRef]
54. Wang, R.; Chen, B.; Liu, B.; Lin, H.; Chen, W. The Application Research of Deep Learning Based Method for Short-term Wind Speed Forecasting. In Proceedings of the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 20–22 December 2019; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2019; pp. 2378–2381. [CrossRef]
55. Fan, H.; Zhang, X.; Mei, S.; Zhang, J. A Markov Regime Switching Model for Ultra-Short-Term Wind Power Prediction Based on Toeplitz Inverse Covariance Clustering. *Front. Energy Res.* **2021**, *9*, 638797. [CrossRef]
56. Zhang, Y.; Zhou, S.; Zhang, Z.; Yan, L.; Liu, L. Design of an Ultra-Short-Term Wind Power Forecasting Model Combined with CNN and LSTM Networks. In *Recent Developments in Intelligent Computing, Communication and Devices, Proceedings of the International Conference on Intelligent Computing, Communication & Devices (ICCD) 2019, Xi'an, China, 22–24 November 2019*; Springer: Singapore, 2021; Volume 1185, pp. 141–145. [CrossRef]
57. Fan, H.; Zhang, X.; Mei, S.; Chen, K.; Chen, X. M2GSNet: Multi-Modal Multi-Task Graph Spatiotemporal Network for Ultra-Short-Term Wind Farm Cluster Power Prediction. *Appl. Sci.* **2020**, *10*, 7915. [CrossRef]
58. Alkabbani, H.; Ahmadian, A.; Zhu, Q.; Elkamel, A. Machine Learning and Metaheuristic Methods for Renewable Power Forecasting: A Recent Review. *Front. Chem. Eng.* **2021**, *3*, 1–21. [CrossRef]
59. Dong, X.; Qian, L.; Huang, L. Short-Term Load Forecasting in Smart Grid: A Combined CNN and K-Means Clustering Approach. In Proceedings of the 2017 IEEE International Conference on Big Data & Smart Computing, Jeju, Korea, 13–16 February 2017; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2017; pp. 2375–9356. [CrossRef]
60. Huang, C.J.; Kuo, P.H. A Deep CNN-LSTM Model for Particulate Matter (PM2.5) Forecasting in Smart Cities. *Sensors* **2018**, *18*, 2220. [CrossRef] [PubMed]
61. Wang, S.; Sun, Y.; Zhai, S.; Hou, D.; Wang, P.; Wu, X. Ultra-Short-Term Wind Power Forecasting Based on Deep Belief Network. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2019; pp. 7479–7483. [CrossRef]
62. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; pp. 326–366.
63. Yang, R.; Singh, S.K.; Tavakkoli, M.; Amiri, N.; Yang, Y.; Karami, M.A.; Rai, R. CNN-LSTM deep learning architecture for computer vision-based modal frequency detection. *Mech. Syst. Signal Process.* **2020**, *144*, 106885. [CrossRef]
64. Yin, W.; Kann, K.; Yu, M.; Schütze, H. Comparative study of CNN and RNN for natural language processing. *arXiv* **2017**, arXiv:1702.01923.
65. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 1995; pp. 1942–1948. [CrossRef]
66. Trelea, I.C. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Inf. Process. Lett.* **2003**, *85*, 317–325. [CrossRef]
67. Shi, X.H.; Liang, Y.C.; Lee, H.P.; Lu, C.; Wang, L.M. An improved GA and a novel PSO-GA-based hybrid algorithm. *Inf. Process. Lett.* **2005**, *93*, 255–261. [CrossRef]
68. Shiwei, Y.; Yiming, W.; Kejun, Z. Hybrid Optimization Algorithm Based on Particle Swarm Optimization and Genetic Algorithm. *J. Syst. Eng. Electron* **2011**, *33*, 1647–1652. [CrossRef]
69. Yeh, W.-C. A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems. *Expert Syst. Appl.* **2009**, *36*, 9192–9200. [CrossRef]
70. Yeh, W.-C. An improved simplified swarm optimization. *Knowl. Based Syst.* **2015**, *82*, 60–69. [CrossRef]
71. Willmott, C.J.; Matsuura, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.* **2005**, *30*, 79–82. [CrossRef]
72. Frenzel, J.F. Genetic algorithms. *IEEE Potentials* **1993**, *12*, 21–24. [CrossRef]
73. Kennedy, J.; Eberhart, R.C. A Discrete Binary Version of the Particle Swarm Algorithm. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 1997; Volume5, pp. 4104–4108. [CrossRef]