Electrical Power and Energy Systems 141 (2022) 108143

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

An advanced short-term wind power forecasting framework based on the optimized deep neural network models

Seyed Mohammad Jafar Jalali [a], Sajad Ahmadian [b], Mahdi Khodayar [c], Abbas Khosravi [a], Miadreza Shafie-khah [d], Saeid Nahavandi [a], João P.S. Catalão [e,*]

[a] Institute for Intelligent Systems Research and Innovation, (IISRI), Deakin University, Geelong, Australia
[b] Faculty of Information Technology, Kermanshah University of Technology, Kermanshah, Iran
[c] Department of Computer Science - University of Tulsa, USA
[d] School of Technology and Innovations, University of Vaasa, Vaasa, Finland
[e] Faculty of Engineering of University of Porto and INESC TEC, Porto, Portugal

ARTICLE INFO

Keywords:
Deep neural networks
Evolutionary computation
Neuroevolution
Optimization
Wind power forecasting

ABSTRACT

With the continued growth of wind power penetration into conventional power grid systems, wind power forecasting plays an increasingly competitive role in organizing and deploying electrical and energy systems. The wind power time series, though, often present non-linear and non-stationary characteristics, allowing them quite challenging to estimate precisely. The aim of this paper is in proposing a novel hybrid model named Evol-CNN in order to predict the short-term wind power at 10-min interval up to 3-hr based on deep convolutional neural network (CNN) and evolutionary search optimizer. Specifically, we develop an improved version of Grey Wolf Optimization (GWO) algorithm by incorporating two effective modifications in its original structure. The proposed GWO algorithm is more effective than the original version due to performing in a faster way and the ability to escape from local optima. The proposed GWO algorithm is utilized to find the optimal values of hyperparameters for deep CNN model. Moreover, the optimal CNN model is employed to predict wind power time series. The main advantage of the proposed Evol-CNN model is to enhance the capability of time series forecasting models in obtaining more accurate predictions. Several forecasting benchmarks are compared with the Evol-CNN model to address its effectiveness. The simulation results indicate that the Evol-CNN has a significant advantage over the competitive benchmarks and also, has the minimum error regarding of 10-min, 1-hr and 3-hr ahead forecasting.

1. Introduction

In recent years, wind energy has gained remarkable attention as a clean source of electricity that addresses crucial environmental concerns [1–4]. The stability and reliability of energy production and the reduction in greenhouse gas emission are significant challenges recently emerged in the domain of power engineering [2,5–8]. The accurate prediction of wind power, which is considered as a highly varying time series with a stochastic and intermittent nature, plays a key role in overcoming such issues [1,9]. Even though wind power generated by a wind turbo depends heavily on atmospheric climatic conditions, the accurate prediction of wind power results in improved wind energy predictions. Consequently, the recent literature presents a broad variety of time series forecasting algorithms for the prediction of wind power time series. The nature of wind data is stochastic and chaotic, which means that predicting wind power with linear models is a very challenging task [10]. Furthermore, the length of the prediction horizon correlates negatively with the accuracy of the forecasting algorithm [9]. Ultrashort- term wind forecasting relates to forecasting of wind data within a few minutes to one hour ahead. This operation is primarily aimed at clearing the electricity sector, grid operations in real time and regulatory activities [11]. Short-term predictions are generally for a duration from one hour to several hours ahead. This type of forecasting is typically used for unit engagement and operational safety in the energy industry [11].

The wind power forecasting methodologies presented in recent technical literature can be categorized into four classes:
(1) The persistent model (PR) assumes that the future values of wind measurements have similar values as the most recent historical measurement. The smoothness assumption in the model leads to a simple method with the lowest computational resources required; however,

* Corresponding author.
E-mail address: catalao@fe.up.pt (J.P.S. Catalão).

https://doi.org/10.1016/j.ijepes.2022.108143
Received 11 October 2021; Received in revised form 6 February 2022; Accepted 18 March 2022
Available online 6 April 2022

Nomenclature

w_{ik}^{l-1}	Kernel of the ith neuron at layer $l-1$ towards the kth neuron of the layer l
α	Alpha wolf
β	Beta wolf
δ	Delta wolf
Δ_k^l	Delta of the kth neuron at layer l
Γ	Standard gamma function
$\hat{y}_i$	Vector of predicted wind power data
ω	Omega wolf
$\vec{X}_p$	Position vectors of the prey
$\vec{X}$	Position vectors of the grey wolf
b_k^l	Scalar bias of the kth neuron at layer l
b_j	Total number of the item type j
$conv1Dz(.,.)$	Fully convolutional operation in 1-D space using the zero padding
l	Current convolution layer
$l-1$	Previous convolution layer
lb	Lower bounds of the search space
$Levy()$	Levy flight function
$rev(.)$	Array reversing
s_i^{l-1}	Output of the ith neuron of layer $l-1$
ub	Upper bounds of the search space
x_k^l	kth feature of input
x_{ij}	Integer values of the X_i position in jth dimension
y_k^l	Intermediate output of neuron from the input
y_i	Vector of observed wind power data
$y_{(t)}$	Actual wind power for the time step t
y_{ij}	Transformed real number of the jth dimension of individual i
AEMO	Australian Energy Market Operator
AI	Artificial intelligence
ANN	Artificial neural network
AR	Auto regressive
ARIMA	Auto regressive integrated moving average
ARMA	Auto regressive moving average
B_s	Batch size
BaNN	Bagging neural network
BP	Back propagation
CNN	Convolutional neural network
D_r	Dropout rate
DE	Differential evolution
Evol-CNN	Proposed evolutionary CNN framework
FFNN	Feed-forward neural network
FP	Forward propagation
GWO	grey wolf optimizer
IGWO	Improved version of GWO
K_s	Kernel size
L_r	Learning rate
LSTM	Long short-term memory
M_r	Momentum rate
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MI	Mutual information
MP_s	Maxpooling size
MSE	Mean square error
N_c	Number of convolutional layers
N_e	Number of epochs
N_f	Number of filters
PR	Persistent model
PSO	Particle swarm optimization
RBFNN	Radial basis function neural network
ReLU	Rectified linear unit
RMSE	Root mean square error
SAE	Stacked auto-encoder
SGD	Stochastic Gradient Descent
SVR	Support vector regression

the accuracy of this model is significantly declined as the prediction horizon is expanded [11].

(2) The Physical models work on the basis of numerical weather prediction (NWP) by considering different meteorological parameters such as temperature, pressure, and obstacles. NWP leads to accurate predictions for long-term wind forecasting tasks, and utilize in problems corresponding to large-scale predictions in very wide areas. The major disadvantage of this method is the large time and memory complexity that is addressed by using supercomputers. Thus, this method cannot be employed for real-time predictions with limited computational resources [12].

(3) Statistical methods learn the mathematical relationship between different variables corresponding to real-time wind data samples. This category of approaches include the auto regressive (AR), auto regressive moving average (ARMA), auto regressive integrated moving average (ARIMA), Bayesian approach, as well as the grey predictions. In this domain, the authors of [13] developed a novel version of ARIMA for the short-term prediction of wind data that eases the decision making in wind sectors. Also, the research work in [14] provides a novel signal decomposition technique based on ARIMA to capture the general trend of wind speed datasets. Furthermore, a robust spline regression model is proposed in [15] which is optimized by the variational Bayesian method [16]. In this class of approaches, an infinite Markov switching AR is also presented by [17] as a non-parametric Bayesian framework with flexible posterior distribution for accurate wind forecasting in large-scale datasets.

(4) Artificial intelligence (AI) methodologies have shown their great potential for solving numerous types of real world applications [18–24]. AI algorithms such as artificial neural networks, support vector regression [25], transfer learning [26] and fuzzy systems led to novel wind prediction algorithms in very recent studies [27]. Artificial neural networks (ANNs) are widely used as nonlinear models that extract powerful relationships between the input wind measurements and the future wind speed values. In this group, ANNs are successfully applied to the prediction of different weather time series with various time scales [28]. Feed-forward ANN, recurrent ANN, radial basis function (RBF) ANN, and adaptive wavelet ANN are recently proposed for wind speed and wind power forecasting applications.

While these methodologies provide a highly nonlinear regression model, they cannot handle the large variations in the wind time series due to the lack of computational layers. Also, determining the optimal values of ANNs parameters such as their weights and biases is a challenging task in this research area [29]. As a result, recent studies are focused on deep neural networks that consist of large number of latent layers to extract wind power data in a supervised and unsupervised manner [30].

Among recent deep learning models, stacked autoencoders [31] are presented to develop an accurate multi-scale prediction method that learns heterogeneous wind features in a real-time fashion. In [32], a deep belief network is applied that can learn the most significant unsupervised wind features in a probabilistic manner. Also, the authors of [11] developed an interval version of this model that can address the uncertainties and noise in wind measurements. Long Short-Term Memory network [33] is a recurrent version of deep ANNs that applies a large number of temporal latent layers to effectively learn powerful temporal characteristics of the wind power data. In [34], an ensemble algorithm based on deep CNN and wavelet transform is proposed for wind power forecasting using a real wind farm dataset from China. The research work presented in [35] proposes an improved empirical mode decomposition combined with bagging neural network (BaNN), K-means clustering method, and shark smell optimization (SSO) algorithm to predict wind power data. In [36], a hybrid deep CNN with a radial basis function neural network (RBFNN) and double Gaussian function (DGF) approaches are applied to 24 hr-ahead short-term wind power forecasting. In [37], a strong combination of two-dimensional CNN and improved wavelet transform trained by improved version of particle swarm optimization (PSO) is proposed to forecast wind power for short and long-term forecasting horizons. [38] applies a differential evolution (DE) algorithm to optimize the hyperparameters of deep long short-term memory (LSTM) neural networks for a time series forecasting problem. In another work [39], a forecasting algorithm based on variational-mode decomposition long short-term memory (VMD-LSTM) is proposed in order to improve the accuracy of multi-step wind power forecasting.

In most of the previous works, the architecture of deep learning algorithms is manually designed, which is a time consuming and difficult task to do [40–42]. In order to overcome these issues, advanced deep neuroevolution search methods have been introduced to design this procedure automatically and efficiently [43,44]. These methods have been successfully implemented and used in several real-world problems such as computer vision, fraud detection and medical domains [40]. Deep neuroevolution is defined as the process of optimizing the architecture of deep learning networks by evolutionary methods in order to achieve highest accuracy and obtain the optimal architectures for DNNs [40].

In this paper, a novel deep neuroevolution method is presented to employ convolution operation for extracting powerful temporal features from the wind power signals. Our model is optimized by a new deep neuroevolution algorithm for short-term wind power prediction. In contrast to classic deep models, we introduce a modified version of grey wolf optimizer (GWO) with CNNs to overcome the local minimum issues in the optimization of the hyperparameters and obtain the best architectures with highest accuracy. Besides, we employ the mutual information (MI) feature extraction strategy in order to obtain the optimal input features for our proposed deep learning model. Our proposed forecasting algorithm is applied on both ultrashort-term and short-term forecasting horizons. In summary, the main contributions of this study can be summarized in four main categories:

- An improved version of GWO algorithm is proposed as a novel evolutionary optimization approach to strengthen the search space capability and reduce the possibility of trapping into local optima.
- An efficient forecasting method is developed based on the improved GWO algorithm and CNN deep learning model for wind power signals to obtain the highest prediction accuracy. To the best of our knowledge, this is the first attempt to solve wind power prediction problem by a deep neuroevolution algorithm. To this end, the proposed method considers nine critical hyperparameters of deep CNNs to be optimized by the improved GWO algorithm.
- An effective mutual information feature extraction strategy is applied to increase the accuracy of the wind power forecasting procedure and gain the optimal input features for such a highly volatile signal.
- Our simulations show the superiority of the proposed method over the state-of-the-art and recently published works in terms of 10 min up to 3 h ahead predictions for two case studies.

The rest of the paper is organized as follows: In Section 2, the theories and components of the proposed method are presented. Section 3 denotes to the numerical analysis for wind power data and finally, the paper is concluded in Section 4.

2. Proposed method

In this section, we develop an integration of deep neural network and evolutionary search strategy to propose a novel wind power forecasting method called Evol-CNN. To this end, the available time series data for wind power in the past is used as the input of deep convolutional neural network for forecasting wind power in the future. There are several hyperparameters used in deep CNN architecture that their values impact on the performance of deep CNN. In other words, the accuracy of deep CNN for forecasting wind power can be improved by determining the optimal values of its hyperparameters. Therefore, we develop an improved version of grey wolf optimization algorithm as an evolutionary search strategy to obtain the optimal values of hyperparameters for the deep CNN. In the following subsections, we first discuss the representation of the solutions in the evolutionary optimization algorithm. Then, the fitness function used in the proposed method is introduced in details. Finally, we discuss our novel search strategy for the Evol-CNN algorithm.

2.1. Representation of solutions

In the proposed method, we adopt one-dimensional convolutional neural network since the dimension of the data used for time series forecasting problems has generally one dimension. Thus, we apply 1D-CNN for forecasting the unknown values of wind power data. 1D-CNN has several hyperparameters and the aim of using improved GWO algorithm is to obtain optimal values for these hyperparameters. To this end, we optimize nine critical hyperparameters by Evol-CNN. Therefore, we need to define each solution in the population space of improved GWO algorithm as a vector of nine values corresponding to the considered hyperparameters. On the other hand, the basic search strategy of GWO algorithm is mainly applied for optimization problems with continuous domain of individuals, while the hyperparameters values of a CNN should mostly be discrete values. Therefore, we use an encoding transformation function to map each real number vector, which transforms the position of an individual in the continuous space to a new integer vector for an individual in the discrete space as follows [45]:

$$Q_{ij} = \lfloor b_j * \frac{x_{ij} - lb}{ub - lb} + 0.5 \rfloor, j = 1, \dots, n \tag{1}$$

where x_{ij} represents the real values of the X_i position in the jth dimension and Q_{ij} denotes to the transformed integer number of the jth dimension of individual i. b_j denotes to the total number of the item type j and the lower and upper bounds of the search space are represented by lb and ub, respectively.

The overall schema of representation of solutions in the population space of IGWO algorithm for the Evol-CNN is shown in Fig. 1.

2.2. Calculation of fitness function

Convolutional neural networks (CNNs) are a type of deep neural networks which generally processes complex and large datasets. Despite the fact that CNNs were effectively used in a wide range of real-world problems, quite few researches have confirmed CNN for wind power forecasting. Mostly the CNN technique involve three main layers: convolutional layer(s), pooling layer(s), and fully-connected layer. The convolutional layer uses an algebraic theory known as ‘‘convolution’’ to

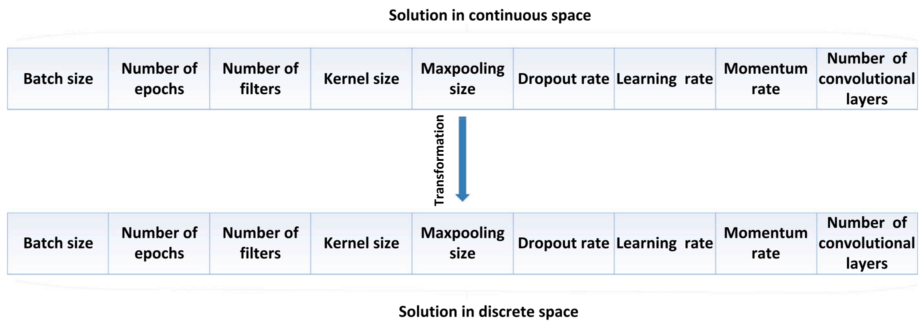

Fig. 1. Representation of each solution in population space of IGWO algorithm for Evol-CNN.

retrieve feature points inside the raw data, whereas the pooling layer is utilized to decrease the dimension of the input data. At last, a fully-connected layer at the end of the CNN forecasts the points based on the retrieved unique features. Also every convolutional layer is designed to derive patterns from the response component (wind power) as shown below:

$$y_{ik}^{k} = f\left(\left(W^{k} * h\right)_{ij} + b_{k}\right) \tag{2}$$

where f denotes to the activation function. Here, W^k is the kernel weight and * is the convolutional layer. We also use Rectified linear unit (Relu) as the activation function in the present work.

In the optimization process of the Evol-CNN, the mean square error (MSE) is used to evaluate the performance of performed CNN for each solution in the search space which is considered as the fitness function. Therefore, the fitness function used in the optimization process can be calculated as follows:

$$E_p = MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

where y_i is a vector of observed wind power data points being predicted and $\hat{y}_i$ represents the vector of n predictions generated from a sample of n wind power data points.

We use dropout and learning rate techniques in the fully-connected layer as a form of regularization to prevent over-fitting during training. The nonlinear transformation of input data has been conducted via the processes of convolution, pooling, and the fully-connected layers. In addition to these configurations, we use momentum rate which is a powerful strategy helping to improve both training speed and accuracy. Thus, the distinguishing features of input data for prediction have been learnt in this procedure. Finally, training procedure is performed by SGD method.

The aim of our novel Evol-CNN algorithm is in obtaining the optimal values of CNN hyperparameters leading to improve the performance of wind power forecasting. To this end, we need to consider the values set of CNN hyperparameters as a vector representing the solutions in the search space. Then, the proposed optimization approach can be applied to find the optimal values of CNN hyperparameters by exploring the search space. On the other hand, in the optimization process, we need to define a fitness function to evaluate the quality of each solution. To this end, we apply CNN model on the values of training samples and the prediction error value obtained by CNN is considered as the fitness function. Suppose that the historical wind power values for M time steps are represented by a vector as follows:

$$\vec{y} = (y_{(0)}, y_{(1)}, \dots, y_{(M-1)}) \tag{4}$$

where $y_{(t)}$ represents the actual wind power for the time step t. Then, we utilize CNN model to predict the wind power values for the next N time steps. The predicted values of wind power for the next time steps are expressed as follows:

$$\vec{\hat{y}} = (\hat{y}_{(M)}, \hat{y}_{(M+1)}, \dots, \hat{y}_{(M+N-1)}) \tag{5}$$

where $\hat{y}_{(t)}$ denotes to the predicted wind power values for the time step t. After forecasting the wind power data using the CNN model, its error is considered as the fitness value of the corresponding solution. To this end, the MSE metric is used to calculate the error of CNN as the fitness value using Eq. (3). In the proposed model, the number of convolutional layers is automatically obtained by the used optimization strategy. Also, one max pooling layer followed by a dense fully connected layer are used for constructing the backbone of CNN architectures.

2.3. Search strategy

In the Evol-CNN method, we design an improved version of grey wolf optimization algorithm as an evolutionary search strategy to obtain the optimal values of hyperparameters for CNN model. GWO algorithm is a swarm evolutionary meta-heuristic inspired by encircling and hunting behaviour of the grey wolves in nature [46]. We apply each individual in GWO algorithm to configure a CNN based on the obtained values of hyperparameters. Then, the wind power values in training data can be predicted by the configured CNN.

The main optimization process of Evol-CNN starts with an initialization step where a number of individuals are initialized with random values representing the hyperparameters values of CNN model. Therefore, the number of dimensions in each individual vector is equal to the number of hyperparameters optimized by the improved GWO algorithm. Each individual represents a solution containing the values of hyperparameters for CNN model. After the initialization step, the search procedure of improved GWO algorithm continues with obtaining new generations of the first population and repeating this step by a number of predefined iterations to find the optimal solution corresponding to the optimal values of hyperparameters.

When designing the improved GWO algorithm, we consider the best solutions as the alpha (α) wolves. The second and third best solutions, respectively, are called beta (β) and delta (δ) wolves. It is assumed that the rest of the candidate solutions are omega (ω) wolves. Hunting behaviour (optimization procedure) is driven by α, β, and δ. These three wolves accompany the other wolves. The encircling behaviour is mathematically modelled using the following equations:

$$\vec{D} = |\vec{C}.\vec{X}_p(t) - \vec{X}(t)| \tag{6}$$

$$\vec{X}(t+1) = |\vec{X}_p(t) - A.\vec{D}| \tag{7}$$

where t denotes the current iteration, A and C indicate coefficients, and $\vec{X}_p$ and $\vec{X}$ represent the position vectors of a prey and a grey wolf, respectively.

The coefficients A and C are determined as follows:

$$A = 2a.\vec{r}_1 - a \tag{8}$$

$$C = 2.\vec{r}_2 \tag{9}$$

where the components of a are decreased linearly from 2 to 0 during the iterations and the vectors of r_1 and r_2 have random values in the [0, 1] interval.

In order to model the hunting behaviour of the grey wolves mathematically, we presume that α (the best candidate solution), β, and δ have better awareness of the possible location of the prey. Accordingly, the first three best solutions that have been obtained so far have to be saved and the other search agents (including the omega wolves) need to change their positions based on the current location of the best search agents. In this respect, the distances between any other wolves (including Omegas) and these three best wolves are determined by the following equations in the decreasing order of their fitness:

$$\begin{aligned} \vec{D}_\alpha &= |C_1.\vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta &= |C_2.\vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta &= |C_3.\vec{X}_\delta - \vec{X}| \end{aligned} \tag{10}$$

Using the Eqs. (6) and (7), these distances are applied to provide the new position of wolf $\vec{X}(t+1)$.

$$\begin{aligned} \vec{X}_1 &= \vec{X}_\alpha - A_1.\vec{D}_\alpha \\ \vec{X}_2 &= \vec{X}_\beta - A_2.\vec{D}_\beta \\ \vec{X}_3 &= \vec{X}_\delta - A_3.\vec{D}_\delta \end{aligned} \tag{11}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{12}$$

The best solution or prey is found by repetitively deploying the two encircling and hunting operators.

In a nutshell, the GWO starts with a random grey wolf population. The mechanism of searching is principally driven by α, β and δ. They diverge in looking for prey when $|A| > 1$ and converge in attacking prey when $|A| < 1$. Eventually, if the stop criterion is met, the optimal solution (prey) is obtained.

Although GWO is a robust optimizer that has demonstrated outstanding efficiency across a number of optimization algorithms, this paper offers a two-phase modification technique that boosts the efficiency of this algorithm stronger than before. The proposed modifications of GWO algorithm are described in the following.

- *First modification*:

Based on the greedy selection (GS) procedure of DE algorithm, we employ the idea of "survival of the fittest" with the probability p. As per this technique, new dominant positions within each generation proceed to be improved for the next generations. Moreover, the worse positions are ignored. The formula for this operator is as following:

$$\vec{x}(t+1) = \begin{cases} \vec{x}(t) & f\left(\vec{x}_{new}(t)\right) > f(\vec{x}(t)) \text{ and } r_{new} < p \\ \vec{x}_{new}(t) & \text{otherwise} \end{cases} \tag{13}$$

where $f(X(t))$ represents the last position fitness, r_{new} and p denote to the random values into the $(0,1)$ range, and $X_{new}(t)$ represents the new position obtained by Eq. (12). In each iteration, the value of p in Eq. (13) is characterized into the $[0,1]$ range randomly. The search abilities are enhanced by the combination of GS into GWO since each leader wolf gets the opportunity to stay alive and afterwards share their observed information with other hunters in the next phases of the search procedure.

- *Second modification*:

In GWO, the parameter A is utilized to monitor the step size of the search agents which tends to decrease linearly with iterations. In this modification phase, we use the strong functionality of the levy flight strategy in order to tune the parameter A. This modification improves the potential exploration and exploitation of GWO, continuously. Assume the ∞ parameter indicates to the step size as following:

$$\infty \oplus \text{Levy}(\beta) \sim 0.01 \frac{p}{|q|^{1/\beta}} \left(X_i^k - X_{\text{best}}^k\right) \tag{14}$$

where the values of p and q are defined by:

$$p \sim N\left(0, \phi_u^2\right), q \sim N\left(0, \phi_v^2\right) \tag{15}$$

$$\phi_u = \left[\frac{\Gamma(1+\beta) \times \sin(\pi \times \beta/2)}{\Gamma[(1+\beta)/2] \times \beta}\right]^{1/\beta}, \phi_v = 1 \tag{16}$$

where Γ symbolizes the standard gamma function in interval $[0,2]$. We modify the parameter A using Eq. (17) as follows:

$$\mathbf{A} = Levy(X) * \mathbf{u} \tag{17}$$

where X represents the position of wolves and u is a random value between [0, 1] range. These concepts are used to improve the global exploration as well as local exploitation capacity of conventional technology and to deepen the searching advantages of GWO.

These two-stage modifications greatly improve the local exploitation and global exploration of GWO. We name this new algorithm as improved GWO (IGWO). The flowchart of the proposed IGWO is shown in Fig. 2. Besides, a pseudo-code of our advanced wind power forecasting framework has been provided in Algorithm 1.

Algorithm 1 Pseudo-code of the proposed wind power forecasting model

```
Input: pop_size (population size) and n (maximum number of iterations).
Output: Predicted wind power.
Begin algorithm:
Split dataset into two sets including training set Tr and test set Te;
Initialize the grey wolf population X_i (i = 1, 2, ..., pop_size);
Initialize parameter α, A and C;
for (each solution X_i in the grey wolf population) do
    Set a CNN model based on the values of solution X_i as the hyperparameters;
    Calculate the fitness of solution X_i using Eq. (3) as the MSE error of CNN model
obtained based on the training set Tr;
end for
Let X_α be the best solution;
Let X_β be the second best solution;
Let X_δ be the third best solution;
Apply GS strategy;
while (number of iterations < n) do
    for each solution X_i in the grey wolf population do
        Update the position of X_i using Eq. (11);
        Set a CNN model based on the values of solution X_i as the hyperparameters;
        Calculate the fitness of solution X_i using Eq. (3) as the MSE error of CNN model
obtained based on the training set Tr;
    end for
    Update α, A by Levy flight operator and C;
    Update X_α, X_β and X_δ
    Increase the number of iterations by 1;
end while
Set a CNN model based on the values of solution X_α as the hyperparameters;
Predict the wind power data in the test set Te using the CNN model;
End algorithm
```

Due to the selection of best possible sets of hyperparameters and architectures, the strategy of training CNNs is considered to be a complex and difficult problem with an uncertain search space. Moreover, in the IGWO, the stability between exploration and exploitation phases is successful, which can be quite effective in addressing complex challenges such as CNN training. Thus, we can obtain the best solution containing the optimal values of CNN hyperparameters after performing IGWO algorithm. It should be noted that, the evolutionary search strategy is applied based on the training data to configure the CNN model with best optimal hyperparameter values. Then, the configured CNN model is applied on test data to predict unknown values of wind power data points. The overall procedure of the Evol-CNN is conceptually presented in Fig. 3.

3. Experimental results and discussions

3.1. Wind power data

We evaluate the novel Evol-CNN algorithm on 10 min intervals of wind power data provided by the Australian Energy Market Operator (AEMO) for the whole year of 2010 from an existing wind farm in Australia [47]. In this work, the data of Woolnorth wind farm located in northwest of Tasmania is taken into consideration. This wind farm consists of 62 turbines with 140 megawatt (MW) nominal capacity. It is important to note that the wind site of Woolnorth is one of the most challenging situations for wind power forecasting in Australia

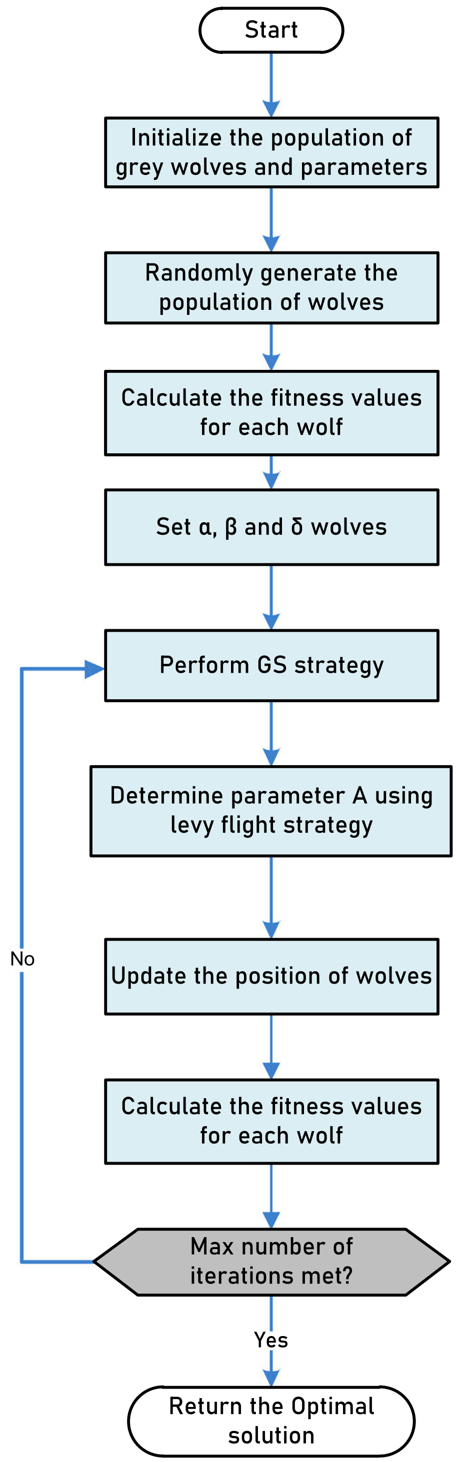

Fig. 2. The flowchart of the proposed IGWO.

because of its location on the edge of a cliff facing the Southern Ocean [47]. Similar to [10], we divided the used dataset based on different seasons to show the performance of wind forecasting models in different weather conditions. This scenario makes it possible to show the sensitivity of compared models to different seasons. Then, for each season, 75 percent of the data is used for training, and the remaining is utilized for testing. The training set is used to train the models and the test set is used to evaluate the performance of the compared models in terms of different evaluation metrics.

Table 1
The IGWO parameters.

Parameter	Value
a	[2→0]
Population size	20
Number of iteration	20
Number of runs	10

Table 2
List of CNN hyperparameter symbols and their values.

Symbol	Value
B_s	[10, 20, …, 100]
N_e	[1, 300]
N_f	[1, 300]
K_s	[1, 25]
MP_s	[1, 15]
D_r	[0.2, 0.25, …, 0.65]
L_r	[0.001, 0.006, …, 0.1]
M_r	[0.05, 0.1, …, 0.95]
N_c	[1, 2, …, 5]

To obtain a better representation of the input features values and improve the forecasting performance, we employ mutual information (MI) strategy. By considering $p(t)$ as the value of wind power for time t, we measure the MI of $p(t-l+1)$ and $p(t+1)$ assuming l regarded as the time-lag of wind power time series values. We measure the MI based on the lags from $l=1$ to $l=100$. We pick the time lags with MI values higher than a threshold $\tau=0.4$ to be considered as the selected input sets for making a better correlation of wind power time series data which leads to generate the time-lags from $l=1$ to $l=29$. Let us assume we are currently at the time t and the future time horizon's wind power value will be predicted. Thus, based on this inference, our selected input set is considered as $29+28=57$ dimensional set $\{p(t-28), \Delta p(t-27), p(t-27), \dots, p(t)\}$ with the sequential difference $\Delta p(t)=p(t)-p(t-1)$ in the wind power dataset.

3.2. Initialization setups for evol-CNN

Selecting values for the initial parameters in evolutionary algorithms to train deep neural networks plays an important role in the performance of these types of networks, and IGWO is no exception from this rule. We chose the initialized values of the IGWO algorithm based on the recommendations in aforementioned works [46]. These values are as shown in Table 1.

On the other hand, it is necessary to determine the architecture of deep CNNs before their training, which is associated by selecting the hyperparameters aligned with each network layer. In this study, we optimize nine hyperparameters with Evol-CNN framework including batch size (B_s), number of epochs (N_e), number of filters (N_f), kernel size (K_s), maxpooling size (MP_s), dropout rate (D_r), learning rate (L_r), momentum rate (M_r), and number of convolutional layers (N_c). Based on the previous literature [48], these hyperparameters are the most important hyperparameters which have significantly impact on the performance of CNN training. Table 2 shows the hyperparameters and their ranges which are used for the experiments in this work. These values are selected based on the suggestions from literature and trial and error for not resulting in over-fitting. The other hyperparameters used in CNN training are the activation function type which has been initialized by powerful ReLU function, the optimizer is considered by SGD, and pooling type is considered with maxpooling. Also, the performance of different forecasting models in this paper is evaluated using three well-known evaluation metrics including the root mean square error (RMSE), mean absolute percentage error (MAPE) and mean absolute error (MAE).

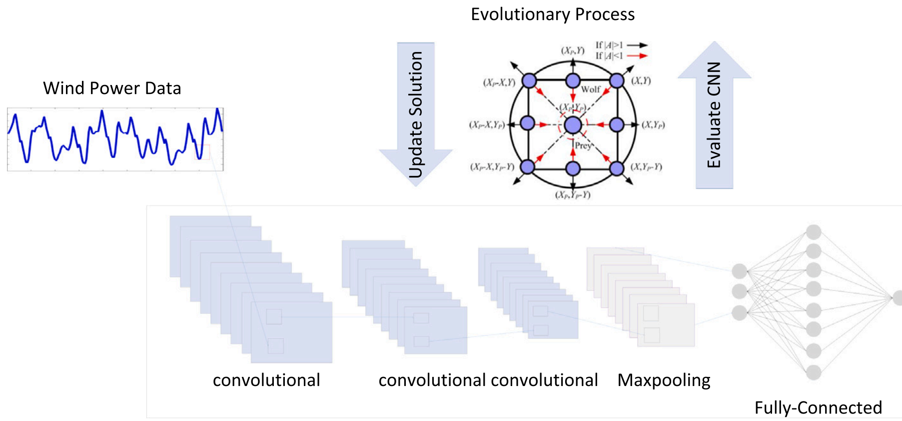

Fig. 3. An overview of the proposed wind power forecasting model.

Table 3
Error values of forecasting methods for spring season using different time horizons.

Model	Time step								
	10 min			1 h			3 h		
	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE
PR	0.1312	295.617	0.0972	0.1554	325.794	0.1022	0.2019	403.908	0.1328
AR	0.1108	256.223	0.0833	0.1389	221.026	0.0954	0.1961	349.241	0.09963
ARMA	0.1076	189.343	0.0782	0.1323	178.454	0.0902	0.1936	317.565	0.09577
ARIMA	0.0996	144.668	0.0754	0.1261	153.778	0.0855	0.1927	279.118	0.09042
SVR	0.0711	82.227	0.0415	0.1178	107.156	0.0734	0.1923	236.442	0.08166
CNN	0.0481	21.939	0.0277	0.0922	63.116	0.0581	0.1724	204.811	0.06343
FFNN	0.0673	37.513	0.0361	0.1124	97.212	0.0693	0.1881	225.666	0.07242
LSTM	0.0426	18.413	0.0254	0.0919	55.707	0.0577	0.1533	184.932	0.06122
DE-LSTM	0.0413	18.255	0.0247	0.0901	55.265	0.0531	0.1514	182.677	0.05896
SAE	0.0396	17.173	0.0236	0.0862	52.656	0.0472	0.1449	172.559	0.05361
DeepHybrid	0.0382	16.688	0.0222	0.0834	48.054	0.0456	0.1412	167.201	0.05032
GWO-CNN	0.0361	15.545	0.0205	0.0794	42.335	0.0424	0.1397	164.302	0.04754
Evol-CNN	**0.0346**	**13.103**	**0.0177**	**0.0752**	**34.229**	**0.0313**	**0.1364**	**161.559**	**0.0403**

Table 4
Error values of forecasting methods for summer season using different time horizons.

Model	Time step								
	10 min			1 h			3 h		
	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE
PR	0.1354	508.831	0.0415	0.1559	279.494	0.0526	0.3814	305.667	0.0855
AR	0.0938	315.992	0.0369	0.1433	178.101	0.0445	0.3177	242.881	0.0694
ARMA	0.0903	256.166	0.0361	0.1356	135.772	0.0427	0.3086	192.442	0.0668
ARIMA	0.0881	114.883	0.0346	0.1282	91.552	0.0418	0.2981	162.982	0.0627
SVR	0.0742	42.372	0.0298	0.1223	76.636	0.0397	0.2591	145.872	0.0589
FFNN	0.0632	36.856	0.0288	0.1074	69.284	0.0392	0.2451	141.208	0.0575
CNN	0.0618	34.217	0.0281	0.1052	62.646	0.0388	0.2313	135.662	0.0556
LSTM	0.0593	31.193	0.0278	0.1045	56.123	0.0385	0.2051	129.362	0.0542
DE-LSTM	0.0575	30.227	0.0272	0.1023	53.676	0.0379	0.2012	126.018	0.0527
SAE	0.0532	27.883	0.0265	0.1008	51.099	0.0376	0.1972	124.026	0.0514
DeepHybrid	0.0511	26.109	0.0258	0.0986	49.898	0.0371	0.1943	120.433	0.0498
GWO-CNN	0.0496	24.656	0.0251	0.0971	47.222	0.0365	0.1897	117.404	0.0485
Evol-CNN	**0.0472**	**21.267**	**0.0232**	**0.0953**	**43.227**	**0.0315**	**0.1866**	**113.228**	**0.0462**

3.3. Simulation results

In this section, we compare the performance of our proposed Evol-CNN method with the classical baselines for short-term wind power forecasting models including persistence (PR) algorithm [9], auto-regressive (AR), auto regressive moving average (ARMA), and auto-regressive integrated moving average (ARIMA). In addition, the single and hybrid methods in recent literature are compared with the proposed model. A single model approach applies a single regression architecture to undertake the prediction task. In order to demonstrate the impact of deep feature learning on wind data regression problems, we compare the Evol-CNN with shallow deep ANN-based methods, including feed-forward neural network (FFNN), long short-term memory (LSTM), and convolutional neural network (CNN). Besides, support vector regression (SVR) [49] is chosen as another powerful supervised learning benchmark used for regression tasks in the literature.

On the other hand, hybrid algorithms use multiple methods of wind feature extraction to improve the accuracy of prediction tasks. In this work, we compare the proposed Evol-CNN model with the recently proposed hybrid differential evolution-LSTM (DE-LSTM) [50] algorithm that employs DE to optimize the LSTM hyperparameters, as well as deep stacked auto-encoder (SAE) [9] that learns rough patterns from the input wind data. Also, a combination of standard version of GWO with deep CNN has been provided in order to show the searching capability of our IGWO model. In addition, we compare our proposed Evol-CNN

Table 5
Error values of forecasting methods for autumn season using different time horizons.

Model	Time step								
	10 min			1 h			3 h		
	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE
PR	0.1029	111.586	0.0455	0.1188	113.868	0.0446	0.1819	166.859	0.0519
AR	0.9652	74.402	0.0405	0.1098	85.922	0.0408	0.1682	145.911	0.0462
ARMA	0.0933	65.881	0.0392	0.1094	78.912	0.0397	0.1635	141.667	0.0451
ARIMA	0.0892	58.922	0.0371	0.1089	73.443	0.0389	0.1592	138.509	0.0438
SVR	0.0826	34.832	0.0309	0.1055	52.052	0.0342	0.1476	127.669	0.0415
FFNN	0.0622	19.914	0.0289	0.0991	61.948	0.0329	0.1445	120.446	0.0397
CNN	0.0615	16.332	0.0283	0.0936	60.407	0.0324	0.1408	117.494	0.0391
LSTM	0.0598	14.601	0.0278	0.0944	53.245	0.0321	0.1373	116.221	0.0378
DE-LSTM	0.0573	13.651	0.0268	0.0913	51.806	0.0317	0.1355	115.109	0.0371
SAE	0.0521	11.282	0.0262	0.0882	49.202	0.0311	0.1317	111.865	0.0356
DeepHybrid	0.0493	10.769	0.0256	0.0853	46.099	0.0306	0.1284	107.257	0.0331
GWO-CNN	0.0472	10.121	0.0241	0.0821	44.788	0.0296	0.1266	104.556	0.0326
Evol-CNN	**0.0445**	**9.343**	**0.0211**	**0.0794**	**39.545**	**0.0278**	**0.1238**	**102.433**	**0.0315**

Table 6
Error values of forecasting methods for winter season using different time horizons.

Model	Time step								
	10 min			1 h			3 h		
	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE
PR	0.1289	125.9188	0.0358	0.1549	215.661	0.0347	0.2977	357.326	0.0496
AR	0.1124	97.2224	0.0309	0.1481	178.545	0.0325	0.2561	289.156	0.0455
ARMA	0.1053	88.1532	0.0292	0.1419	148.919	0.0317	0.2451	251.727	0.0434
ARIMA	0.0956	79.4322	0.0278	0.1356	124.771	0.0308	0.2238	238.115	0.0416
SVR	0.0847	50.1192	0.0251	0.1243	99.0987	0.0288	0.2142	192.663	0.0391
FFNN	0.0809	25.6676	0.0235	0.1285	90.3269	0.0275	0.2055	188.919	0.0362
CNN	0.0758	22.5572	0.0228	0.0121	67.8111	0.0271	0.1863	183.434	0.0346
LSTM	0.0583	18.9991	0.0224	0.1159	66.9182	0.0266	0.1791	178.934	0.0338
DE-LSTM	0.0561	17.2118	0.0218	0.1134	44.3099	0.0259	0.1761	173.244	0.0332
SAE	0.0535	15.8782	0.0215	0.1113	39.5664	0.0254	0.1722	169.092	0.0327
DeepHybrid	0.0493	12.2999	0.0208	0.1073	34.3373	0.0251	0.1693	163.455	0.0321
GWO-CNN	0.0476	11.6673	0.0204	0.1089	30.8982	0.0259	0.1674	159.915	0.0309
Evol-CNN	**0.0435**	**10.1983**	**0.0198**	**0.1022**	**27.2891**	**0.0246**	**0.1635**	**154.676**	**0.0302**

Table 7
The best CNN architectures found by Evol-CNN.

Dataset	Horizon	RMSE	Hyperparameters								
			N_f	K_s	N_e	B_s	MP_s	D_r	L_r	M_r	N_c
Spring	10 min	0.031	40	1	30	60	2	0.25	0.011	0.05	1
	1 h	0.071	70	2	20	40	3	0.25	0.046	0.1	3
	3 h	0.131	30	3	70	30	2	0.35	0.006	0.05	1
Summer	10 min	0.041	80	1	30	50	2	0.4	0.026	0.2	1
	1 h	0.092	70	2	40	70	1	0.2	0.031	0.05	2
	3 h	0.179	50	3	20	60	2	0.15	0.016	0.3	3
Autumn	10 min	0.038	55	1	30	30	3	0.35	0.041	0.1	3
	1 h	0.072	20	2	20	30	6	0.25	0.031	0.15	3
	3 h	0.114	20	3	40	20	2	0.45	0.021	0.25	1
Winter	10 min	0.043	35	1	20	30	1	0.25	0.026	0.1	2
	1 h	0.098	50	1	20	40	4	0.2	0.006	0.3	2
	3 h	0.156	30	2	30	30	5	0.15	0.011	0.2	1

with the hybrid algorithm proposed in [11] named as DeepHybrid. The design of this algorithm is based on deep belief network (DBF) and fuzzy type II inference system (FT2IS) for the supervised regression of future wind values.

In order to have a fair comparison for choosing the best configurations for hyperparameters of deep ANNs, the learnable hyperparameters including dropout rate (D_r), learning rate (L_r) and momentum rate (M_r) are taken into consideration. For CNN, LSTM and FFNN models, D_r is considered with values corresponding to 0.3, 0.25 and 0.3, respectively. Also, L_r is equal to 0.006, 0.021 and 0.36 for CNN, LSTM and FFNN models, respectively. Finally, M_r is assigned to values equal to 0.05, 0.3 and 0.2 for CNN, LSTM and FFNN models, respectively. These values have been chosen based on the trial and error through a grid search strategy. For other algorithms, the optimal values of parameters reported in their corresponding papers are used in the experiments. The number of runs and number of iterations for all baseline models are considered the same as our proposed Evol-CNN model. All of the algorithms are implemented using Python 3.7 on a GPU of NVIDIA GTX 1080 Ti with the Intel Core i7 CPU and 32 GB RAM.

Tables 3–6 show the average of RMSE and MAPE of the different methods to determine 10 min, 1 h and 3 h forecasting ahead of wind power data points for different seasons. The RMSE and MAPE generated from different algorithms for spring dataset are tabulated in Table 3, showing that the RMSE and MAPE generated by the Evol-CNN in each prediction step (from 10 min to 3 h forecasting horizon) carry out the lowest values. In Table 4, the Evol-CNN algorithm has higher prediction accuracy in comparison with other twelve benchmarks for summer season. Moreover, the PR and SVR perform weaker than neural network family algorithms. This happens since irregularity and the linearity of wind power data is very high and these two methods are not able

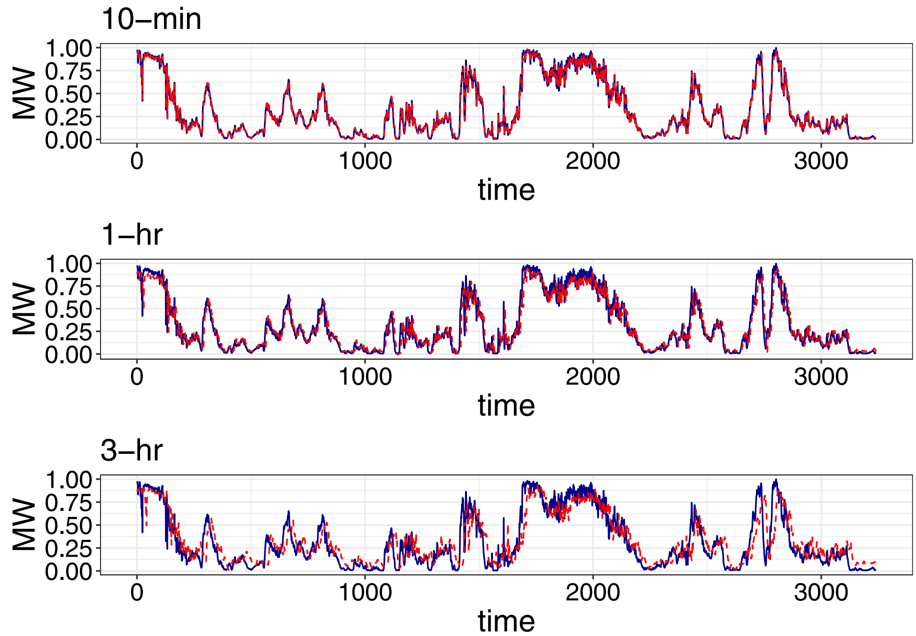

Fig. 4. Actual vs predicted values of Evol-CNN.

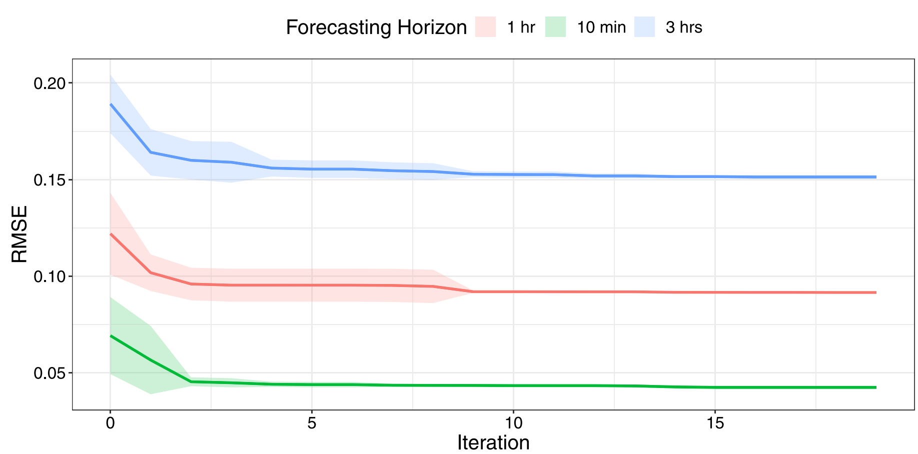

Fig. 5. Convergence curves of proposed Evol-CNN algorithm.

to compete with ANN algorithms. Among deep ANNs, DeepHybrid algorithm outperforms other deep ANN frameworks. However, the Evol-CNN already has a higher modelling capability of wind power forecasting for this case.

As can be seen in Table 5, for ultrashort-term predictions, the PR model has reasonably good performance, however yields poor results as time steps rise. With longer horizons, SVR and ANN methods have significantly smaller values of RMSE and MAPE compared to PR. LSTM improves the RMSE and MAPE for 10 min, 1 h and 3 h compared to two other NN methods such as CNN and FFNN. DE-LSTM framework still outperforms better than LSTM in all seasons because differential mutation operator stabilizes the search space of the LSTM algorithm and increases its accuracy. However, compared to deep ANNs, DeepHybrid outperforms better than the other ones. Among all methods, Evol-CNN has the best forecasting performance for different horizons. The result of RMSEs and MAPEs generated of winter season by the twelve benchmark methods are tabulated in Table 6, indicating that the Evol-CNN in three different time horizons outperforms other benchmarks. This is primarily due to the extraction of more substantive features through CNN representation and also to the robustness of extracted features resulting from the optimization process in IGWO.

In Table 7, the best architectures found with lowest RMSE by Evol-CNN for three different time steps of four seasons are represented. The overall conclusion drawn from this table is that the values chosen by the Evol-CNN are approximately not computationally high. For example, to select the proper values for N_f, the algorithm chooses numbers that range from 20 to 80, which are almost far from the end of N_f interval equal to 300. Thus, it is deduced that for network training, the Evol-CNN chooses normal values with lower computational costs.

In order to intuitively present the performance of the Evol-CNN algorithm, the test dataset of wind power time-series for spring season and their predicted values is shown in Fig. 4. In this figure, the blue and red lines indicate the actual and predicted wind power data points, respectively. For predicting the next 10 min interval, the two lines almost overlap, meaning that the predicted values are close to the actual real data points. Nonetheless, as the horizon steps increase, the performance for predicting the next 1 h and 3 h decreases. This is also rational, since it is more difficult to predict the 1 h and 3 h ahead wind power forecasting than the 10 min prediction.

Fig. 5 illustrates the convergence curves of Evol-CNN algorithm using 10 independent runs for spring dataset. According to this figure, as forecasting horizon goes up, the prediction error increases. Moreover,

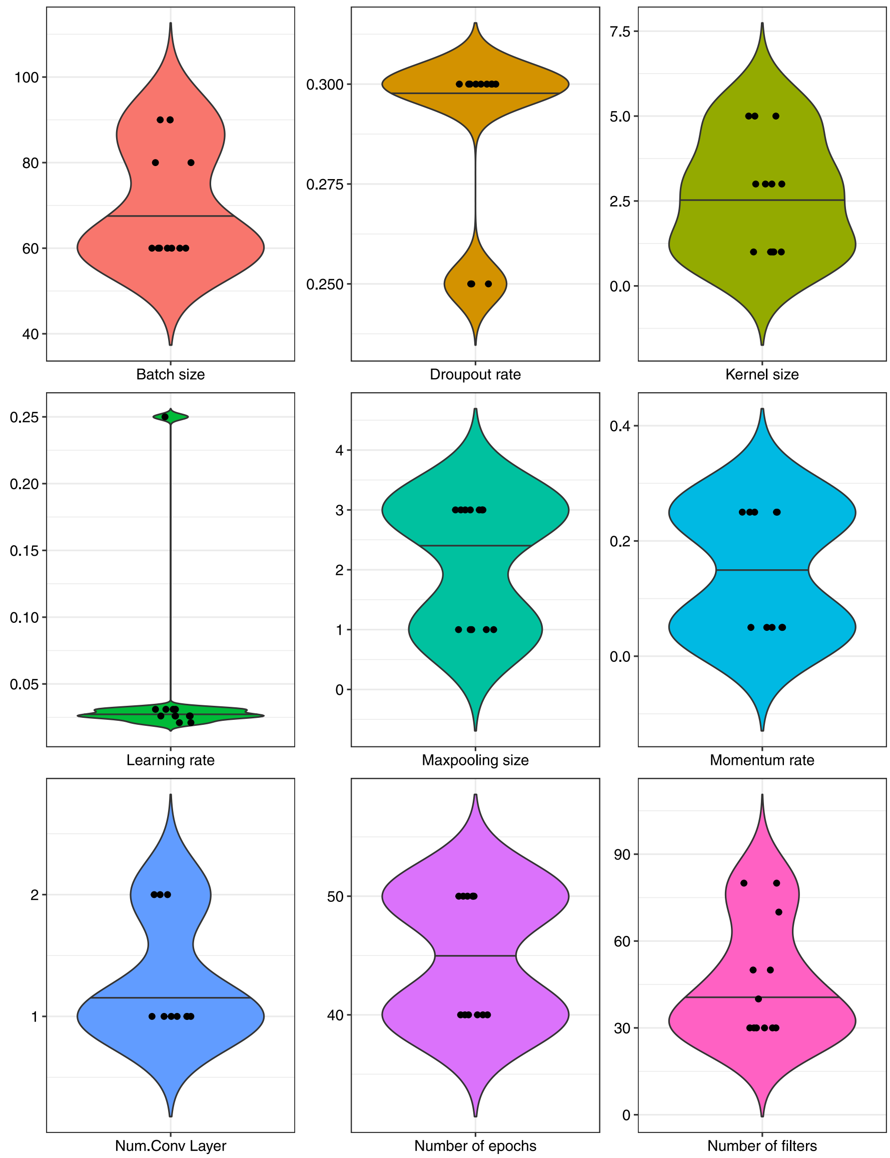

Fig. 6. Violin plots of hyperparameters generated by Evol-CNN for 10 min interval.

it is much easier to converge for a forecasting horizon of 10 min ahead compared to 1 h and 3 h ahead. Finally, for all forecasting horizons, the optimization process converges properly toward the end of iterations.

Figs. 6–8 shows the violin plots of nine optimized hyperparameters for three different horizons of spring season. This figure is important since it can share valuable information about the selection of the main hyperparameter values for CNN architectural design procedure. For instance, to select the initialization values for dropout rate, Fig. 8 shows that the appropriate values for this hyperparameter fall into the value of around 0.3. On the other hand, the value of 0.6 is not suggested for CNN training since it does not contain large amounts of dropout values during 10 times of Evol-CNN running. Such an interpretation applies to other hyperparameters of Figs. 6–8 as well.

In order to determine statistically the significance of the differences between the performance of the Evol-CNN and other benchmarks, the T-test statistic technique is conducted. This test is carried out on the basis of the Evol-CNN results at 5% significance level and degree of freedom equal to 3 against each of the other benchmarks. Table 8 lists the obtained p values performed by T-test. By investigating the obtained p values in this table, it can be seen that the null hypothesis (significant difference) at 5% significance level is rejected in all cases. Therefore, we can conclude that the proposed EvolCNN model is significantly better than other compared models in three horizons of each season dataset.

4. Conclusion

This paper presents a novel algorithm called Evol-CNN which is a combination of deep CNNs and improved version of GWO algorithm for wind power forecasting. The aim of this algorithm is in optimization of the CNN hyperparameters in a discrete space for improving the accuracy of wind power forecasting. We also use the MI strategy for obtaining the optimal features for our proposed model. In order to demonstrate the effectiveness of the Evol-CNN, the performance of this algorithm is compared with twelve forecasting benchmarks on an Australian wind farm dataset for three different horizons. Considering different short-horizon time steps for different scenarios, Evol-CNN showed relatively better performance than other benchmarks in terms of RMSE and MAPE evaluation metrics.

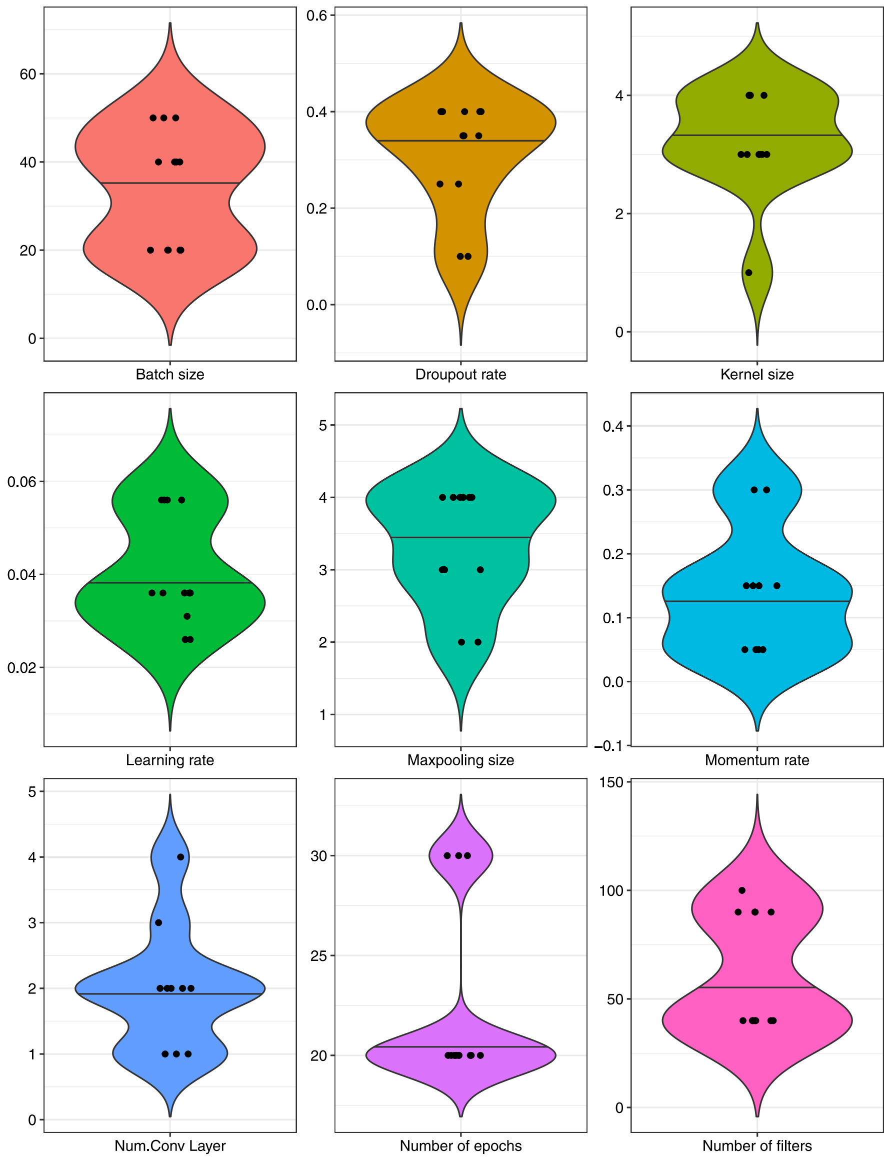

Fig. 7. Violin plots of hyperparameters generated by Evol-CNN for 1 h interval.

Table 8
p values of T-test for Evol-CNN forecasting results vs other models.

Season	Horizon	PR	AR	ARMA	ARIMA	SVR	FFNN	CNN	LSTM	DE-LSTM	SAE	DeepHybrid	GWO-CNN
Spring	10 min	9.81E−07	5.92E−06	5.53E−06	4.91E−06	1.66E−05	1.58E−05	1.54E−04	3.61E−03	8.14E−04	1.90E−03	1.16E−02	1.03E−02
	1 h	7.98E−07	8.97E−06	8.41E−06	7.93E−06	6.15E−06	5.22E−06	5.13E−05	1.18E−05	1.03E−05	1.12E−04	1.09E−04	7.66E−03
	3 h	3.27E−06	2.77E−06	2.41E−06	1.93E−06	8.87E−07	5.28E−07	7.78E−06	6.12E−05	3.18E−04	2.35E−03	1.61E−03	1.32E−03
Summer	10 min	3.98E−07	3.51E−07	3.21E−07	2.81E−07	1.09E−06	2.23E−04	1.91E−04	1.69E−04	1.15E−03	5.99E−03	1.63E−02	1.23E−02
	1 h	1.31E−06	1.26E−06	1.02E−06	1.12E−06	1.61E−06	4.59E−05	4.26E−04	4.11E−04	6.25E−04	1.24E−03	9.83E−04	1.11E−03
	3 h	1.45E−06	1.21E−06	1.02E−06	9.11E−05	1.16E−04	3.72E−07	2.71E−05	1.65E−04	1.54E−04	4.57E−05	3.43E−03	1.78E−03
autumn	10 min	7.69E−07	6.44E−07	6.11E−07	3.16E−07	2.73E−06	2.56E−06	2.90E−04	3.18E−04	4.99E−04	9.00E−05	3.01E−03	1.91E−03
	1 h	2.45E−05	1.22E−05	7.81E−04	3.21E−04	2.85E−06	6.15E−06	1.54E−04	7.51E−05	1.16E−04	4.19E−04	1.92E−04	1.25E−03
	3 h	9.59E−05	8.12E−05	7.77E−05	7.13E−05	5.33E−06	2.12E−05	1.85E−05	1.58E−04	3.95E−04	4.57E−05	2.09E−03	1.56E−03
Winter	10 min	4.57E−07	9.11E−06	6.56E−06	4.13E−06	2.93E−06	2.77E−06	2.02E−06	4.43E−05	2.63E−04	4.57E−05	5.33E−04	2.46E−04
	1 h	3.96E−06	1.90E−06	1.01E−06	9.72E−05	1.43E−04	6.12E−05	2.90E−04	9.27E−05	2.02E−03	2.18E−03	3.61E−03	2.25E−03
	3 h	1.67E−07	6.22E−06	2.32E−06	1.13E−06	2.33E−03	9.43E−06	3.98E−04	1.16E−04	5.21E−04	2.22E−03	6.83E−04	3.15E−04

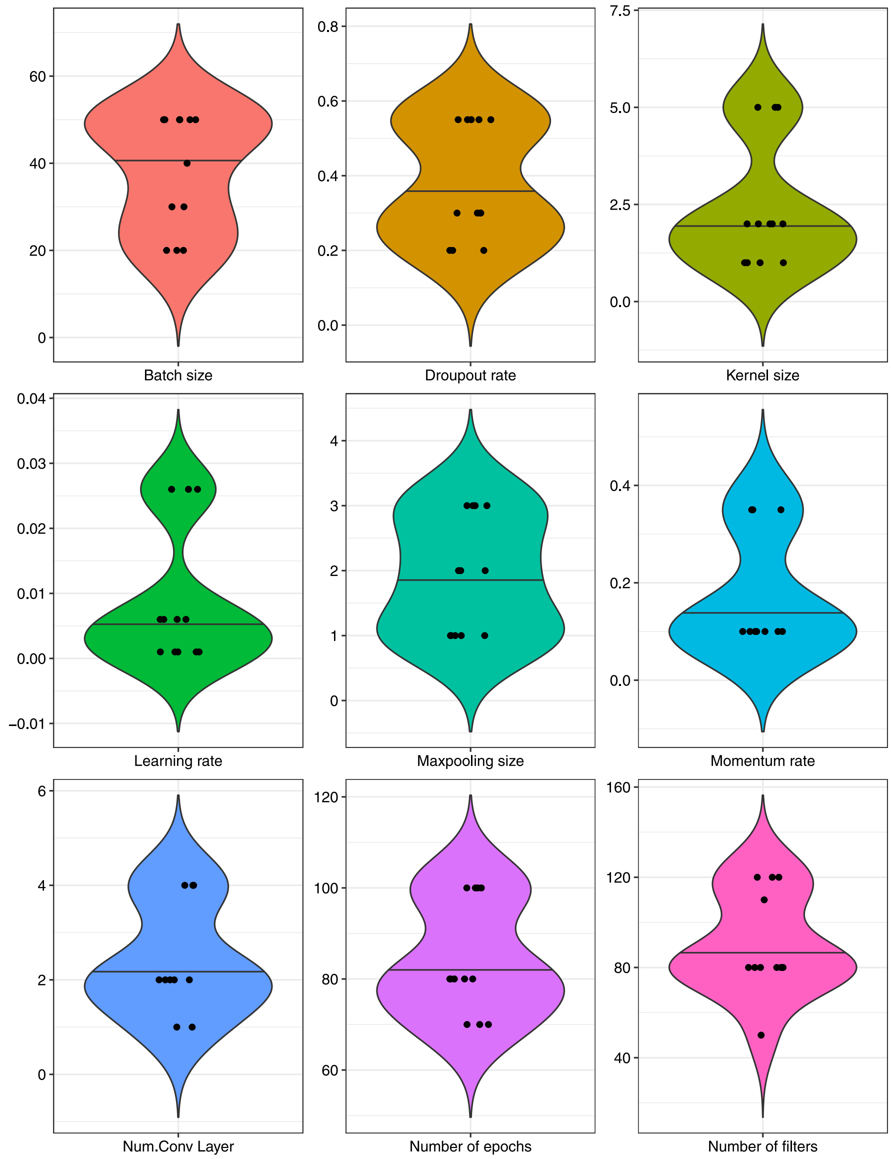

Fig. 8. Violin plots of hyperparameters generated by Evol-CNN for 3 h interval.

CRediT authorship contribution statement

Seyed Mohammad Jafar Jalali: Investigation, Visualization, Writing – original draft. **Sajad Ahmadian:** Methodology, Data curation. **Mahdi Khodayar:** Formal analysis. **Abbas Khosravi:** Supervision. **Miadreza Shafie-khah:** Conceptualization. **Saeid Nahavandi:** Validation. **João P.S. Catalão:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

J.P.S. Catalão acknowledges the support by FEDER funds through COMPETE 2020 and by Portuguese funds through FCT, under POCI-01-0145-FEDER-029803 (02/SAICT/2017).

References

[1] Luo X, Sun J, Wang L, Wang W, Zhao W, Wu J, et al. Short-term wind speed forecasting via stacked extreme learning machine with generalized correntropy. IEEE Trans Ind Inf 2018;14(11):4963–71.
[2] Jalali SMJ, Khodayar M, Ahmadian S, Noman MK, Khosravi A, Islam SMS, et al. A new uncertainty-aware deep neuroevolution model for quantifying tidal prediction. In: 2021 IEEE industry applications society annual meeting. IEEE; 2021, p. 1–6.
[3] Jalali SMJ, Khodayar M, Khosravi A, Osório GJ, Nahavandi S, Catalão JP. An advanced generative deep learning framework for probabilistic spatio-temporal wind power forecasting. In: 2021 IEEE international conference on environment and electrical engineering and 2021 IEEE industrial and commercial power systems Europe. IEEE; 2021, p. 1–6.
[4] Jalali SMJ, Ahmadian S, Khodayar M, Khosravi A, Ghasemi V, Shafie-khah M, et al. Towards novel deep neuroevolution models: Chaotic levy grasshopper optimization for short-term wind speed forecasting. Eng Comput 2021;1–25.
[5] Khodayar M, Khodayar ME, Jalali SMJ. Deep learning for pattern recognition of photovoltaic energy generation. Electr J 2021;34(1):106882.
[6] Jalali SMJ, Ahmadian S, Khosravi A, Shafie-khah M, Nahavandi S, Catalão JP. A novel evolutionary-based deep convolutional neural network model for intelligent load forecasting. IEEE Trans Ind Inf 2021;17(12):8243–53.
[7] Jalali SMJ, Ahmadian S, Kavousi-Fard A, Khosravi A, Nahavandi S. Automated deep CNN-LSTM architecture design for solar irradiance forecasting. IEEE Trans Syst Man Cybern A 2021;52(1):54–65.

[8] Saffari M, Khodayar M, Jalali SMJ, Shafie-khah M, Catalão JP. Deep convolutional graph rough variational auto-encoder for short-term photovoltaic power forecasting. In: 2021 international conference on smart energy systems and technologies. IEEE; 2021, p. 1–6.
[9] Khodayar M, Kaynak O, Khodayar ME. Rough deep neural architecture for short-term wind speed forecasting. IEEE Trans Ind Inf 2017;13(6):2770–9.
[10] Hill DC, McMillan D, Bell KR, Infield D. Application of auto-regressive models to UK wind speed data for power system impact studies. IEEE Trans Sustain Energy 2011;3(1):134–41.
[11] Khodayar M, Wang J, Manthouri M. Interval deep generative neural network for wind speed forecasting. IEEE Trans Smart Grid 2018;10(4):3974–89.
[12] Choi I-J, Park R-S, Lee J. Impacts of a newly-developed aerosol climatology on numerical weather prediction using a global atmospheric forecasting model. Atmos Environ 2019;197:77–91.
[13] do Nascimento Camelo H, Lucio PS, Junior JBVL, de Carvalho PCM, dos Santos DvG. Innovative hybrid models for forecasting time series applied in wind generation based on the combination of time series models with artificial neural networks. Energy 2018;151:347–57.
[14] Zhang J, Wei Y, Tan Z. An adaptive hybrid model for short term wind speed forecasting. Energy 2019;115615.
[15] Wang Y, Hu Q, Srinivasan D, Wang Z. Wind power curve modeling and wind power forecasting with inconsistent data. IEEE Trans Sustain Energy 2018;10(1):16–25.
[16] Liu Y, Qin H, Zhang Z, Pei S, Wang C, Yu X, et al. Ensemble spatiotemporal forecasting of solar irradiation using variational Bayesian convolutional gate recurrent unit network. Appl Energy 2019;253:113596.
[17] Xie W, Zhang P, Chen R, Zhou Z. A nonparametric Bayesian framework for short-term wind power probabilistic forecast. IEEE Trans Power Syst 2018;34(1):371–9.
[18] Ahmadian S, Moradi P, Akhlaghian F. An improved model of trust-aware recommender systems using reliability measurements. In: 2014 6th Conference on information and knowledge technology. IEEE; 2014, p. 98–103.
[19] Tahmasebi F, Meghdadi M, Ahmadian S, Valiallahi K. A hybrid recommendation system based on profile expansion technique to alleviate cold start problem. Multimedia Tools Appl 2021;80(2):2339–54.
[20] Ahmadian M, Ahmadi M, Ahmadian S, Jalali SMJ, Khosravi A, Nahavandi S. Integration of deep sparse autoencoder and particle swarm optimization to develop a recommender system. In: 2021 IEEE international conference on systems, man, and cybernetics. IEEE; 2021, p. 2524–30.
[21] Moradi P, Rezaimehr F, Ahmadian S, Jalili M. A trust-aware recommender algorithm based on users overlapping community structure. In: 2016 sixteenth international conference on advances in ICT for emerging regions. IEEE; 2016, p. 162–7.
[22] Hasani H, Jalali SMJ, Rezaei D, Maleki M. A data mining framework for classification of organisational performance based on rough set theory. Asian J Manag Sci Appl 2018;3(2):156–80.
[23] Jalali SMJ, Hedjam R, Khosravi A, Heidari AA, Mirjalili S, Nahavandi S. Autonomous robot navigation using moth-flame-based neuroevolution. In: Evolutionary machine learning techniques. Springer; 2020, p. 67–83.
[24] Jalali SMJ, Khosravi A, Kebria PM, Hedjam R, Nahavandi S. Autonomous robot navigation system using the evolutionary multi-verse optimizer algorithm. In: 2019 IEEE international conference on systems, man and cybernetics. IEEE; 2019, p. 1221–6.
[25] Kong X, Liu X, Shi R, Lee KY. Wind speed prediction using reduced support vector machines with feature selection. Neurocomputing 2015;169:449–56.
[26] Hu Q, Zhang R, Zhou Y. Transfer learning for short-term wind speed prediction with deep neural networks. Renew Energy 2016;85:83–95.
[27] Marugán AP, Márquez FPG, Perez JMP, Ruiz-Hernández D. A survey of artificial neural network in wind energy systems. Appl Energy 2018;228:1822–36.
[28] Qian Z, Pei Y, Zareipour H, Chen N. A review and discussion of decomposition-based hybrid models for wind energy forecasting applications. Appl Energy 2019;235:939–53.
[29] Ahmadian S, Khanteymoori AR. Training back propagation neural networks using asexual reproduction optimization. In: 7th conference on information and knowledge technology. IEEE; 2015, p. 1–6.
[30] Liu X, Zhang H, Kong X, Lee KY. Wind speed forecasting using deep neural network with feature selection. Neurocomputing 2020;397:393–403.
[31] Chen J, Zhu Q, Li H, Zhu L, Shi D, Li Y, et al. Learning heterogeneous features jointly: A deep end-to-end framework for multi-step short-term wind power prediction. IEEE Trans Sustain Energy 2019.
[32] Wang K, Qi X, Liu H, Song J. Deep belief network based k-means cluster approach for short-term wind power forecasting. Energy 2018;165:840–52.
[33] Liu H, Mi X, Li Y. Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, LSTM network and ELM. Energy Convers Manage 2018;159:54–64.
[34] 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.
[35] Abedinia O, Lotfi M, Bagheri M, Sobhani B, Shafie-khah M, Catalao JP. Improved EMD-based complex prediction model for wind power forecasting. IEEE Trans Sustain Energy 2020.
[36] Hong Y-Y, Rioflorido CLPP. A hybrid deep learning-based neural network for 24 h ahead wind power forecasting. Appl Energy 2019;250:530–9.
[37] Abedinia O, Bagheri M, Naderi MS, Ghadimi N. A new combinatory approach for wind power forecasting. IEEE Syst J 2020.
[38] Hu Y-L, Chen L. A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and differential evolution algorithm. Energy Convers Manage 2018;173:123–42.
[39] Han L, Zhang R, Wang X, Bao A, Jing H. Multi-step wind power forecast based on VMD-LSTM. IET Renew Power Gener 2019;13(10):1690–700.
[40] Stanley KO, Clune J, Lehman J, Miikkulainen R. Designing neural networks through neuroevolution. Nat Mach Intell 2019;1(1):24–35.
[41] Mousavirad SJ, Jalali SMJ, Ahmadian S, Khosravi A, Schaefer G, Nahavandi S. Neural network training using a biogeography-based learning strategy. In: International conference on neural information processing. Springer; 2020, p. 147–55.
[42] Ahmadian S, Jalali SMJ, Raziani S, Chalechale A. An efficient cardiovascular disease detection model based on multilayer perceptron and moth-flame optimization. Expert Syst 2021;e12914.
[43] Ahmadian S, Jalali SMJ, Islam SMS, Khosravi A, Fazli E, Nahavandi S. A novel deep neuroevolution-based image classification method to diagnose coronavirus disease (COVID-19). Comput Biol Med 2021;139:104994.
[44] Jalali SMJ, Ahmadian M, Ahmadian S, Khosravi A, Alazab M, Nahavandi S. An oppositional-Cauchy based GSK evolutionary algorithm with a novel deep ensemble reinforcement learning strategy for COVID-19 diagnosis. Appl Soft Comput 2021;111:107675.
[45] Li Z, He Y, Li H, Li Y, Guo X. A novel discrete grey wolf optimizer for solving the bounded Knapsack problem. In: International symposium on intelligence computation and applications. Springer; 2018, p. 101–14.
[46] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw 2014;69:46–61.
[47] Cutler N, Outhred H, MacGill I. Final report on UNSW project for AEMO to develop a prototype wind power forecasting tool for potential large rapid changes in wind power. The Centre for Energy and Environmental Markets; 2011.
[48] Sun Y, Xue B, Zhang M, Yen GG. Evolving deep convolutional neural networks for image classification. IEEE Trans Evol Comput 2019.
[49] Santamaría-Bonfil G, Reyes-Ballesteros A, Gershenson C. Wind speed forecasting for wind farms: A method based on support vector regression. Renew Energy 2016;85:790–809.
[50] Peng L, Liu S, Liu R, Wang L. Effective long short-term memory with differential evolution algorithm for electricity price prediction. Energy 2018;162:1301–14.