This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2021.3126272, IEEE Transactions on Industry Applications

A New Hybrid Deep Neural Architectural Search based Ensemble Reinforcement Learning Strategy for Wind Power Forecasting

Seyed Mohammad Jafar Jalali, *Student Member, IEEE,* Gerardo J. Osório, Sajad Ahmadian, Mohamed Lotfi, *Member, IEEE*, Vasco M. A. Campos, Miadreza Shafie-khah, *Senior Member, IEEE*, Abbas Khosravi, *Senior Member, IEEE,* and João P. S. Catalão, *Senior Member, IEEE*

Abstract—**Wind power instability and inconsistency involve the reliability of renewable power energy, the safety of the transmission system, the electrical grid stability and the rapid developments of energy market. The study on wind power forecasting is quite important at this stage in order to facilitate maximum wind energy growth as well as better efficiency of electrical power systems. In this work, we propose a novel hybrid data driven model based on the concepts of deep learning based convolutional-long short term memory (CLSTM), mutual information, evolutionary algorithm, neural architectural search procedure and ensemble-based deep reinforcement learning strategies. We name this hybrid model as DOCREL. In the first step, the mutual information extracts the most effective characteristics from raw wind power time series datasets. Secondly, we develop an improved version of the evolutionary whale optimization algorithm in order to effectively optimize the architecture of the deep CLSTM models by performing the neural architectural search procedure. At the end, our proposed deep reinforcement learning based ensemble algorithm integrates the optimized deep learning models to achieve the lowest possible wind power forecasting errors for two wind power datasets. In comparison with fourteen state of the art deep learning models, our proposed DOCREL algorithm represents an excellent performance seasonally for two different case studies.**

Index Terms—**Wind power forecasting, deep neural architectural search, advanced evolutionary algorithm, ensemble reinforcement learning strategy, hybrid model.**

I. Introduction

Renewable energy will be an incredibly prominent strategic power generation in the foreseeable future. The volume of renewable energy resources has thus risen significantly in recent years and the percentage of electricity generation in the overall electricity generation has certainly grown [1]. In the recent years, the wind power generation is considered as one of the major representatives of renewable energy. Wind power is intermittent, stochastic, and unstable, affecting the safety and reliability operation of the electricity grids as well as the efficiency of potential power suppliers [2], [3]. If the wind farm power can be predicted accurately, the effect of wind power on transmission system service will be substantially reduced, the operational costs of the electricity system will significantly be lowered and the robust foundation for power system will be effectively provided [4].

Wind power forecasting can be classified into ultra-short-term forecasting, short-term forecasting, medium-term forecasting and long-term forecasting from the view of time forecasting scale [5], [6]. Specifically, the ultra-short term forecasting uses mostly historical wind farm data and the forecasting scale of several hours and the short-term prediction uses tens of hours of a couple of days to forecast wind power output. The forecasting methods are primarily classified into physical, statistical, machine-learning based, and hybrid methods [7], [8]. In order to generate the reliable wind power forecasts, the physical approach is focused largely on the geographical climate, weather conditions (such as atmospheric pressure, precipitation, and temperature). Because of the high costs and the complexity of modelling, wind power cannot be accurately predicted by physical approaches. The statistical models are focused on analysis of the wind farm's historical data to determine the relationship between the wind power and certain variables or hidden rules, and apply such indicators to boost their prediction performance [9].

Recently, the machine learning methodologies have proven to be highly efficient with remarkable results for many real-world problems [10]–[16] such as wind power forecasting [17]. Several previous works involve the use of machine learning algorithms such as artificial neural network (ANN), support vector regression (SVR), Kalman filter and extreme learning machine (ELM) that have been successfully applied to forecast the wind power energy in several real-world datasets [18], [19]. On the other hand, the hybrid approaches are regarded as the efficient models for integrating two or more forecasting models. In [20], a multi-step wind forecasting model was presented by an ensemble fuzzy system forecasting model. The Kalman filter and the ANN model were implemented in [21] to address the wind nonlinear behavior and uncertainty, and the ARIMA model was introduced to increase their prediction performances. In [22], the wavelet packet transform and Least square support vector machine models are combined in order to reduce the non-linearity and instability of wind data characteristics. Thus, while the wind power forecasting methods are continuously improving, the more reliable forecasting algorithms still need to be achieved. Besides, as stated in [23], according to the complexity, uncertainty and randomness of wind power data, previous physical, statistical and intelligent models are not efficiently able to derive the depth features found in these non-linear data points.

S.M.J. Jalali, and A. Khosravi are with the Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Waurn Ponds, VIC 3216, Australia.

G J. Osório is with REMIT, Portucalense University Infante D. Henrique, 4200-075, Porto, Portugal. C-MAST, University of Beira Interior, 6200-358, Covilha, Portugal.

S. Ahmadian is with the Faculty of Information Technology, Kermanshah University of Technology, Kermanshah, Iran.

M. Lotfi is with the Faculty of Engineering of the University of Porto and INESC TEC Porto, Portugal.

Vasco M. A. Campos is with the Redes Energéticas Nacionais (REN) SGPS, S.A, 1700-177 Lisbon, Portugal.

M. Shafie-khah is with the School of Technology and Innovations, University of Vaasa, 65200 Vaasa, Finland.

J.P.S. Catalão is with the Faculty of Engineering of the University of Porto (FEUP) and INESC TEC, Porto 4200-465, Portugal (e-mail: catalao@fe.up.pt)

In recent years, deep learning technology as a member of the ANN family has received a large amount of success in the implementation of many forms of classification tasks including speech recognition and computer vision, and also has been widely utilized for several real-world regression tasks such as time series forecasting in energy domain [24]–[28]. Many previous studies have shown that the deep neural networks can boost the model performance in complex approximation function mechanisms and uncover the data complexity attributes with their powerful non-linear capacity by mapping operation. Driven by these accomplishments, the researchers have started working on the wind power and wind speed forecasting using the deep learning technologies. Furthermore, the most widely-known deep learning models in the literature for wind power forecasting are recognized by stacked auto-encoder, stacked denoising auto-encoder, deep belief network (DBN), long short-term memory (LSTM) and convolutional neural network (CNN) models. Among all these models, a large number of studies have revealed that the convolutional neural network and long short-term memory models have an excellent performance for time series-based wind power forecasting problems [29], [30]. In [31], a LightGBM and CNN-based ensemble learning method is proposed which achieves better accuracy than the compared wind power models. An efficient two stage-based strategy by employing wavelet packet transform (WPT) algorithm and a novel deep convolution neural network (EDCNN) model are hybridized to forecast wind power in [32]. In [33], a hybrid deep learning model for short-term bidirectional memory-convolutional neural network (BiLSTM-CNN) is proposed for short-term wind power forecasting. Firstly, the gray correlation analysis is used to pick inputs for the forecasting model. The proposed hybrid model then extracts multi-dimensional input features to forecast wind power from a temporal-spatial perspective, where the Bi-LSTM model is used to mine bi-directional temporal characteristics, while CNN's convolution and pooling operations are used to extract spatial characteristics from various input time series. The authors in [34] proposed a hybrid forecasting framework based on the powerful feed-forward CNN algorithm and the time-circulation neural long short-term memory network (LSTM) models to boost the accuracy of ultra-short-term forecasting for the wind power datasets. Furthermore, in [35], the authors proposed a novel residual-based CNN algorithm for very short-term wind power forecasting. This hybrid framework has a good forecasting performance when compared with state-of-the-art pre-trained networks. An efficient deep neural network model for short-term wind power forecasting which is employed by recurrent neural networks (RNNs) and infinite feature selection has been proposed by [36]. The simulation results of this model represent that it improves significantly the forecasting accuracy in different seasons of the year based on the data from the National Renewable Energy Laboratory (NREL). In [37], the authors introduced a new strategy based on isolated forest and deep learning neural networks to alleviate the damaging consequences of supervisory control and data acquisition-data outliers for wind power forecasting. The experimental results showed that the isolation forest improved the accuracy of training and testing in deep neural network models for anomaly detection in wind power prediction. In another work carried out by [38], the researchers developed a model named by DeepESN which is introduced based on deep learning technology into the echo state network (ESN). This novel strategy is employed for the problem of wind power forecasting and outperforms the existing well-known algorithms for two case studies. Besides, the researchers in [39] developed an efficient deep neural network framework based on the temporal convolutional networks (TCN) and orthogonal array tuning method for forecasting the generated power of the wind turbines in a wind farm. This model showed an excellent outcome in comparison with multi-step ahead deep learning models including LSTM and CNN-LSTM.

Generally, the deep neural network architectures play a crucial role in their performance which is a time-consuming and manual search process that explores a wide range of solutions [40]. The neural architecture search (NAS) methods search for the hyperparameters in a deep neural network architecture to find the most optimum architectures automatically and efficiently. The deep neural network architectures discovered by NAS methods have achieved the high state-of-the-art efficiency in computer vision among other tasks [41], [42] but few works have been implemented by NAS strategy in time series regression problems specially for wind power forecasting. As an example, the authors in [7], proposed a novel NAS strategy based on the rough theory for short-term wind power forecasting. However, their model was not efficient in discovering the wide range of hyperparameters for their proposed deep learning model. In another work proposed by [43], the authors used a random model based on the NAS strategy, however this model works based on the trial and error strategy and also is not intelligent and efficient enough for power consumption forecasting. Thus, introducing an intelligent model based on the NAS technology for the problem of wind power forecasting can be considered as an essential strategy with the least trial and error efforts. On the other hand, the previous studies show that the accuracy of an ensemble learning based-regression model is higher than a single regression model, but there is a certain degree of overlapping in the regression models [44]. It is therefore very noteworthy to investigate how the number of ensemble regression models can dynamically be reduced under the assumption that their performance is guaranteed in which we adopt the strength of the deep reinforcement learning (RL) algorithm in an ensemble manner.

Thus, in this paper, we attempt to propose an accurate hybrid algorithm based on three main stages: a) We extract the most efficient characteristics from raw wind power time series data as the input for the deep learning models, b) We perform the NAS technique by boosting the performance of WOA model based on a two-stage evolutionary strategy to efficiently optimize the architecture of deep convolutional-LSTM models for having the most excellent deep learning models without the procedure of trial and error for designing the deep learning models, and c) At the end, we design an ensemble deep reinforcement learning strategy to integrate the forecasting results obtained by the multiple optimized deep learning models to obtain the highest possible accuracy from the wind power time series datasets. It is worth noting that we consider our proposed method as a hybrid forecasting model due to combine different methodologies in its main procedure. Each of these methodologies has significant advantages in boosting the effectiveness of the proposed method. More specifically, the mutual information strategy is used to extract more efficient characteristics from the raw input data. This strategy is very helpful as the forecasting model can produce more accurate predictions through employing such efficient extracted characteristics. In addition, a combined deep neural network model is utilized by integrating the advantages of the convolutional neural network and long short term memory neural network. This leads to use the ability of convolutional layer in extracting latent features of input data, and also the ability of LSTM model in considering the temporal feature of input data. Also, an improved version of WOA model is developed based on the two effective operators to make a better balance between the exploration and exploitation phases of the WOA model. Then, this improved optimization algorithm is used to obtain the optimal values of the parameters of the combined deep neural network model. Finally, the deep reinforcement learning model is used to integrate the final results of the input forecasting models and obtain more accurate predictions.

The rest of this work is organized as follows: In Section II, the proposed hybrid deep learning model is explained in details. In Section III, the wind power datasets and the experimental setups for running the proposed model are generally described. The satisfying performance and remarks of the proposed method are represented in Section IV. Finally, the conclusions are drawn in Section V.

II. Methodology

In this section, we introduce our novel hybrid framework based on three phases as follows:

- Phase 1: In the first step, we utilize the mutual information (MI) strategy as a powerful input characteristic extraction to obtain the effective characteristics for the deep-optimized CLSTM models. Mutual information assesses a reduction of uncertainty regarding a random variable dependent on another variable's information [45]. In contrast to linear correlations, MI is more generally based on the fact that all information about variables, both linear and non-linear, is contained in this technique, and it is remarkably useful in assessing the same sort of relationship. MI is also easy to interpret, based on information theory and indifferent to size of data bases [45], [46].

Let's assume the wind power time series are represented by $V=\{v(1),v(2),\cdots,v(t-1),v(t),v(t+1),\cdots\}$ where at time t, the wind power is denoted by $v(t)$. As the nature of wind power data involves the continuous data, the wind power at the time $t+1$ relies on the previous wind power value $v(t{-}i{+}1)(i=1,\cdots,N)$, in which N is set to 100.

Therefore, the corresponding wind power at $v(t-i+1)$, is considered in the prediction of wind power at $v(t+1)$, where n is set to be 8760 samples with resolutions of half-hourly horizons. Thus, we construct the matrix of characteristics (indicated by C) as following:

$$C=\left[\begin{array}{lllll} C_1 & C_2 & C_3 & \cdots & C_{100} \end{array}\right]=\left[\begin{array}{ccc} v(t)_1 & \cdots & v(t-99)_1 \\ & & \\ v(t)_n & \cdots & v(t-99)_n \end{array}\right] \tag{1}$$

where $C_k(k=1\ldots100)$ matches to the k-th feature.

The MI among the future wind power $V(t+1)=\left[\begin{array}{lll} v(t+1)_1 & \cdots & v(t+1)_n \end{array}\right]^T$ and all feature vectors are specified by the following formula:

$$\begin{gathered} I(C_k;V(t+1)= \\ \iint p_{\text{joint }}\left(C_k(i),v(t+1)\right)\log\frac{p_{\text{joint }}\left(C_k(i),v(t+1)\right)}{p\left(C_k(i)\right)p(v(t+1))}dC_k(i)dv(t+1) \end{gathered} \tag{2}$$

where p represents the probability density for a single variable whereas p_{joint} denotes to the density of a joint probability between two variables. When two vectors are distinct from each other, the value for MI equals zero.

The MI can be determined with the below expression because of the complexity existed in Eq.(2):

$$I\left(C_k;V(t+1)\right)=H\left(C_k\right)+H(V(t+1))-H\left(C_k,V(t+1)\right) \tag{3}$$

where $H\left(C_k\right)$ and $H(V(t+1))$ are respectively, the entropy for vectors C_k and $v(t{+}1)$ while $H\left(C_k,V(t+1)\right)$ defines the joint entropy of the two vectors represented by:

$$H\left(C_k\right)=-\int p\left(C_k(i)\right)\log p\left(C_k(i)\right)dC_k(i) \tag{4}$$

$$H(v(t+1))=-\int p\left(v(t+1)\right)\log p\left(v(t+1)\right)dv(t+1) \tag{5}$$

$$\begin{gathered} H\left(C_k,v(t+1)\right)= \\ -\iint p_{\text{joint }}\left(C_k(i),v(t+1)\right)\log p_{\text{joint }}\left(C_k(i),v(t+1)\right) \\ dC_k(i)dv(t+1) \end{gathered} \tag{6}$$

Since quantifying the entropy is a challenging task to carry out, the MI estimation algorithms are the commonly used techniques in the literature. Besides, the MI calculation necessitates estimating marginal probability density functions as well as the joint probability density function. To accomplish this, both parametric and non-parametric techniques can be adopted. The density distribution is assumed to have a functional form in parametric density techniques. Nonetheless, for the vast majority of real-world data sets, the parametric form of the underlying density distribution is unknown. Non-parametric approaches for calculating the densities of unknown distributions present a consistent strategy. The most commonly and efficiently used non-parametric method for estimating the MI is histogram [47].

In comparison with the kernel model and k-nearest neighbor model [48], thanks to the simplicity and higher computational performance of the histogram model, in this work, we use histogram model [49] in order to determine the optimal MI for our wind power datasets. To be more specific, the histogram model avoids the bias accumulation issue happening in the time series forecasting problems and it efficiently exacts the number of characteristics to be chosen in the Gaussian scenario on a time series-based characteristic extraction problem.

Based on the experiments we performed, as $i>39$, the MI value among the predicted wind power $V(t+1)$ and the actual wind power $V(t-i+1)$ is minimal sufficient to be dismissed. Therefore, the MI returns the selected features as $x=\{v(t-38),\cdots,v(t-1),v(t)\}$ in order to consider the optimal input vectors for the optimized deep CLSTM models.

- Phase 2: *Neural Architectural Search Procedure:*

The problem of neural architecture search can be theoretically described here.

We name D for the space of all datasets, M for the space of the deep neural network model and A in quest of the architectural search space. Thus, using this interpretation, a search space architecture $\alpha\in A$ encodes all the properties needed for network training on a dataset, including selecting the optimization algorithm for model parameters as well as all hyperparameters and the regularization strategies used in deep neural network architecture.

The general deep learning algorithm Λ calculates the model $m_{\alpha,\theta}\in M_\alpha$ in a dataset d that splits into the training partition

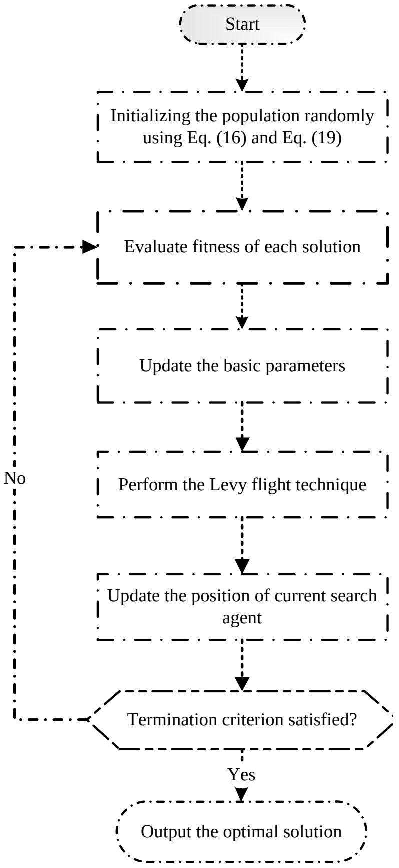
Start
Initializing the population randomly using Eq. (16) and Eq. (19)
Evaluate fitness of each solution
Update the basic parameters
Perform the Levy flight technique
Update the position of current search agent
No
Termination criterion satisfied?
Yes
Output the optimal solution

Fig. 1: Flowchart of the proposed BWOA.

d_{train} and the testing partition d_{test}. This model is calculated by minimizing the $\mathcal{L}$ loss function which has the $\mathcal{R}$ regularization term in respect of the training dataset which is:

$$\Lambda(\boldsymbol{\alpha}, d) = \underset{m_{\alpha,\theta} \in M_{\alpha}}{\arg\min} \mathcal{L}\left(m_{\boldsymbol{\alpha},\boldsymbol{\theta}}, d_{\text{train}}\right) + \mathcal{R}(\boldsymbol{\theta}) \tag{7}$$

The main responsibility of the neural architecture search is in searching the best architecture α^* that minimizes or maximizes the objective function $\mathcal{O}$ on the testing partition dataset $d_{\text{valid.}}$ indicated by:

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha} \in A}{\arg\min} \mathcal{O}\left(\Lambda\left(\boldsymbol{\alpha}, d_{\text{train}}\right), d_{\text{valid}}\right) = \underset{\boldsymbol{\alpha} \in A}{\arg\min} f(\boldsymbol{\alpha}) \tag{8}$$

As our aim in the problem of wind power forecasting is in minimizing the error of wind power datasets, the considered $\mathcal{O}$ is in minimization format. In some cases, the researchers consider $\mathcal{O}$ as the negative loss function $\mathcal{L}$.

Optimizing the f response function is a black-box problem of global optimization. Several approaches such as evolutionary algorithms, reinforcement learning and Bayesian optimization are proposed in the literature for neural architecture optimization. Among these methods, evolutionary algorithms are recognized as one of the most effective and promising optimization methods in deep neural architectural search for various real-world applications [26].

In this study, we utilize the fusion of both powerful deep CNN and LSTM neural network models named as CLSTM to design the baseline of deep neural network architectures for wind power time-series forecasting problem. The deep CLSTM architectural models have gained a great deal of research interest due to their excellent superiority in incorporating the efficiency of automated feature extraction on CNN and high potential in capturing long-term temporal dependency by LSTM algorithm. The convolutional layer designed in CLSTM separates the interconnections by maintaining the probabilistic and stochastic patterns forming the basis of the original time series. Thus, this deep learning model generates more precise feature interpretations to make time frame constraints more accurate towards the LSTM layers.

By providing more details, the data sequence is used for feature extraction firstly as the input for two consecutive convolutional layers. Those low-level features and correlations among parameters in the sense of sequential affects are thereby achieved through convolutional mechanisms of the filters with different properties, non-linear neuron activation and functional interpretation of max pooling operators. The feature map then is transferred to the LSTM layer, which gives a comprehensive analysis of three efficient gates in LSTM, i.e. forgetting input, input and output gates to acquire the complex properties. In particular, the forgetting gate discards unnecessary or repetitive information from the earlier cell states. The input gate extracts efficient new data from the input sequence. In addition, the cell state signals are extracted and transmitted by the output gate to the next state. In fact, the stored temporal data is considered for nonlinear mapping as an input to the dense layers. Finally, the information acquired is transformed into the output vector and the forecasting result values are successfully achieved.

The deep CLSTM neural network models have been practiced to effectively address a number of time series regression problems such as traffic forecasting [50], global horizontal irradiance forecasting [51], and residential energy consumption forecasting [52]. In this study, we develop the CLSTM skeleton architecture, on which the optimized arrangement of the hyperparameters is based on the evaluation of our proposed evolutionary algorithm.

The next step centers on the optimization of the deep CLSTM hyperparameters using our proposed evolutionary algorithm.

The original whale optimization algorithm introduced by Mirjalali and Lewis [53] is a powerful evolutionary algorithm which has shown significant progress for searching the optimal solutions in solving the real-world optimization problems. However, when the original WOA faces with high-dimensional problems such as neural architecture search, it has a slow convergence speed in finding the optimal solution and easily falls into local optima. To overcome these critical issues of WOA, we improve its standard version using two powerful optimization operators including Quasi-opposition and Lévy flight trajectory operators. We name our novel robust evolutionary model as the boosted whale optimization algorithm (BWOA).

First, we introduce the mechanism of WOA in summary. The WOA numerically patterned three activities of humpback whales throughout the hunting procedure i.e. prey encircling, bubble-net attacking and prey searching.

For the phase of encircling prey, humpback whales can well detect the location of prey and encircle it for capturing the prey. In order to mimic this mechanism, let's assume the current optimal solution is globally optimal or located near to it, the other search agents step more towards the current optimal solution and update their current position. The following formula explains this behavior:

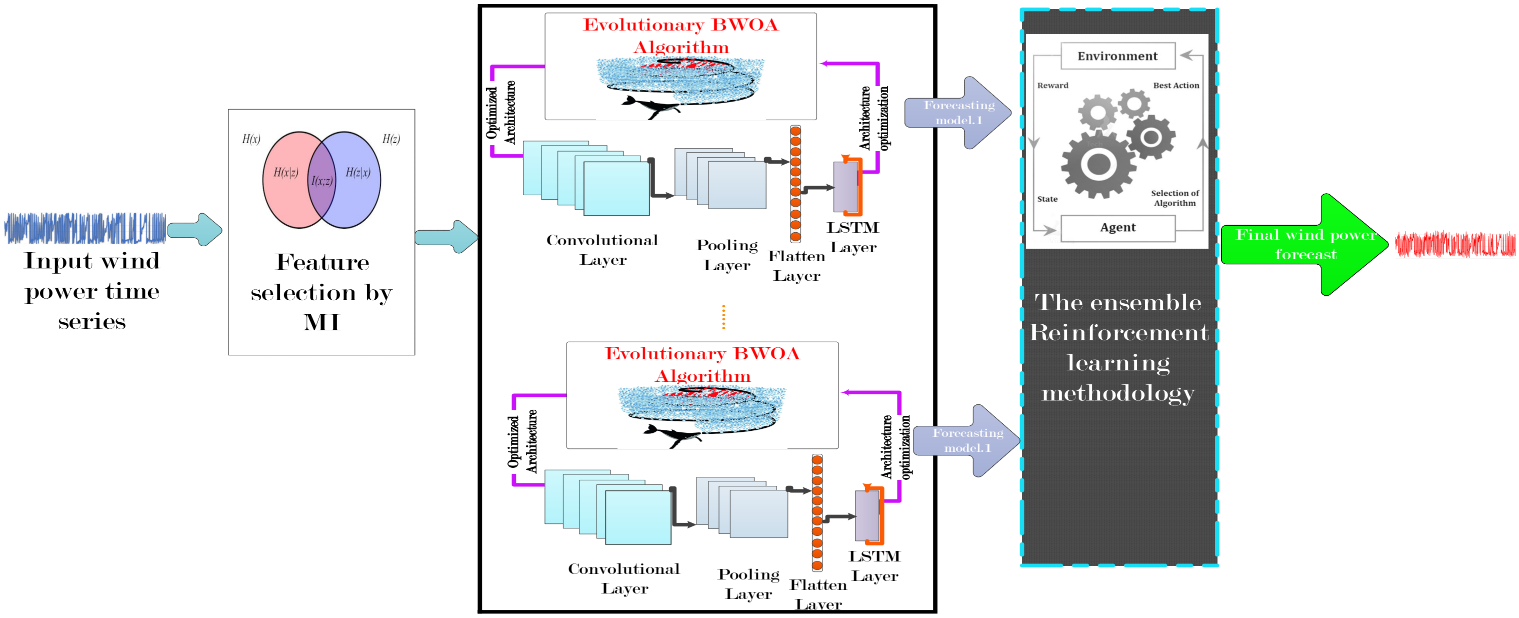

Fig. 2: A general overview of the proposed DOCREL framework.

$$\overrightarrow{D} = \left|\overrightarrow{C} \cdot \overrightarrow{X^*}(t) - \overrightarrow{X}(t)\right| \tag{9}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X^*}(t) - \overrightarrow{A}.\overrightarrow{D} \tag{10}$$

where $X(t)$ represents the position vector, X^* denotes to the best found solution in each iteration that is updated if a better solution has been found in the search space and t is the current iteration number. The $\overrightarrow{A}$ and $\overrightarrow{C}$ represent two coefficient vectors which their updating mechanisms are given by:

$$\overrightarrow{A} = 2\overrightarrow{a} \cdot \overrightarrow{r} - \overrightarrow{a} \tag{11}$$

$$\overrightarrow{C} = 2 \cdot \overrightarrow{r} \tag{12}$$

where $\overrightarrow{a}$ is decreased linearly over the course of iterations from 2 to 0, and r denotes to a random vector in [0, 1].

At bubble-net attacking phase, WOA allows spiral motion to model humpback whales attacking prey with bubble nets. The following is the formula for the mathematical modeling of this behaviour:

$$\overrightarrow{D'} = \left|\overrightarrow{X^*}(t) - \overrightarrow{X}(t)\right| \tag{13}$$

$$\overrightarrow{X'}(t+1) = \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) \tag{14}$$

where $\overrightarrow{D'}$ represents the distance from the ith search agent (whale) to the best found solution (prey), l is a random number in the interval of [-1,1], and b represents a constant value to define the logarithmic spiral shapes.

In searching for prey stage, humpback whales randomly look for the prey on their position judgment during the hunting procedure. The formula for numerical simulation of this action is given by following:

$$\overrightarrow{D} = \left|\overrightarrow{C} \cdot \overrightarrow{X_{rand}} - \overrightarrow{X}\right| \tag{15}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_{rand}} - \overrightarrow{A} \cdot \overrightarrow{D} \tag{16}$$

where $\overrightarrow{X_{rand}}$ represents a random position vector from the population.

In WOA, the $\overrightarrow{A}$ vector determines whether the algorithm performs the exploration (search for prey) or exploitation (encircling prey and bubble-net attacking) phases. If $\left|\overrightarrow{A}\right| > 1$, the algorithm performs exploration ability, otherwise the exploitation ability is activated. It should be noticed that the humpback whales swim all around the prey in a circle loop while going around the fish herds with a logarithmic conical movement. For simplification, we presume the updating position of the humpback whales with a 50 percent probability, that can be interpreted mathematically by:

$$\overrightarrow{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \overrightarrow{A} \cdot \overrightarrow{D} & if\ p < 0.5 \\ \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) & if\ p > 0.5 \end{cases} \tag{17}$$

where p represents a random value in [0, 1].

Most of the evolutionary algorithms including WOA incorporate stochastic operators to evolve during the iterative procedure resulting in a slow rate of convergence speed. Thus, in order to resolve this issue, we utilize the quasi-opposition strategy with the purpose of replacing random search for improving the search space effectively. The opposite point and the opposite number are two main concepts of this strategy. For opposite number, assume $x \in [a, b]$ is a real number which its opposite number x^o is given by:

$$x^o = a + b - x \tag{18}$$

Quasi-opposite point $(X^{qo} = x_1^{qo}, x_2^{qo}, \ldots, x_n^{qo})$ can be described as the point between the center and the opposite point of the search space by:

$$x_i^{qo} = \text{rand}\left(c_i, x_i^o\right) \tag{19}$$

where $c_i = \frac{a_i+b_i}{2}$. This concept allows dynamic jumping of evolutionary WOA in the phase of updating position, which can improve an excellent trade-off between the exploration and exploitation capabilities and ensure it does not collapse into local optimization.

Lévy flight is another powerful evolutionary operator that has the capability to maximize the diversification of search agents in evolutionary algorithms which guarantees in discovering the the search space effectively. This operator is expressed by the following mathematical formula:

$$\overrightarrow{X}(t+1) = \overrightarrow{X}(t) + \mu sign\left[rand - 1/2\right] \oplus Levy \tag{20}$$

$$Levy \sim u = t^{-\lambda}, 1 < \lambda \leq 3 \tag{21}$$

where $\overrightarrow{X}(t)$ represents the ith search agent at iteration t of the position vector $\overrightarrow{X}$, μ denotes to an uniform distribution random number, the $rand$ means a random variable in the [0, 1] interval, $\oplus$ represents the entrywise multiplication and λ denotes to a balanced distribution by the random phase length distribution s using the following formula:

$$s = \frac{\mu}{|\nu|^{1/\beta}} \tag{22}$$

where s means the Lévy flight step distance represented by $Levy(\lambda)$. λ in Eq. (21) dedicates to $\lambda = 1 + \beta$ formula in which $\beta = 1.5$. Both of $\mu = N\left(0, \sigma_\mu^2\right)$ and $\nu = N\left(0, \sigma_\nu^2\right)$ represent the normal distributions given by following formula:

$$\sigma_\mu = \left[\frac{\Gamma\left(1+\beta\right) \times \sin\left(\pi \times \beta/2\right)}{\Gamma\left(\left(\left(1+\beta/2\right)\right) \times \beta \times 2^{(\beta-1)/2}\right.}\right]^{1/\beta} \quad \text{and } \sigma_\nu = 1 \tag{23}$$

By using the powerful Quasi-opposition strategy and the Lévy flight trajectory operators for the basic version of WOA, the diversity of the population against slow convergence and potential enhancement to jump out of maximum local optimisation is increased efficiently. Besides, the exploration and exploitation phases of the WOA is balanced excellently. The schematic flowchart of the BWOA algorithm can be seen in Fig. 1

In the next step, we apply the proposed BWOA to evolve the deep CLSTM hyperparameters to reduce the forecasting error of wind power dataset. On the other hand, the novel BWOA addresses the solution representation and fitness function evaluation, as two major principal considerations utilized for any evolutionary optimization problem. Thus, all solutions in BOWA are determined by an eleven-dimensional vector in which each element represents one of the hyperparameters adopted in deep CLSTM models. These hyperparameters have the critical roles in designing the CLSTM architecture. As an example, max-pooling size, the number of CNN or LSTM layers as well as the learning rate are the hyperparameters used in CLSTM network that BOWA optimizes them efficiently.

As that solution space is repetitively searched by the BWOA strategy, we just need converting the optimum values acquired with their corresponding discrete hyperparameters. As a response, to transform each real value to an integer value, we model this converting procedure by the following equation:

$$y_{ij} = \lfloor g_j * \frac{x_{ij} - lb}{ub - lb} + 0.5 \rfloor, j = 1, ..., n \tag{24}$$

where g_j represents the maximum number of the item type j, x_{ij} denotes to the real number of jth dimension for the solution X_i, while the lower and upper bounds for the search space are defined by lb and ub, respectively.

The BWOA model generates a population of search agents in which each solution is a representative of eleven dimensional vector defining as each hyperparameter used in the evolutionary procedure of CLSTM architecture. Next, the new obtained solutions will update repeatedly using the Quasi-opposition strategy and the Lévy-flight trajectory operators to make an efficient balance between the exploration and exploitation phases as well as increasing more the convergence speed. Furthermore, the WOA meets the stopping condition resulting to the optimal CLSTM hyperparameter values as the best solution found by the proposed BWOA strategy.

For determining the efficiency of each solution, a fitness function has to be defined. Therefore, the input time series data is divided into training and testing portions. The training set is necessary for optimizing hyperparameters of CLSTM while the testing set assesses the efficiency of the final wind power forecasting model. Suppose $\vec{y}$ is a vector which can define the actual wind power time series data for the following M time steps:

$$\vec{y} = (y_{(0)}, y_{(1)}, ..., y_{(M-1)}) \tag{25}$$

where $y(t)$ represents the actual wind power values for the time step t. The combined BWOA-CLSTM model assumes to predict the wind power values in the following N time steps using the CLSTM model:

$$\vec{\hat{y}} = (\hat{y}_{(M)}, \hat{y}_{(M+1)}, ..., \hat{y}_{(M+N-1)}) \tag{26}$$

where the predicted values of the time step t is denoted by $\vec{y}_{(t)}$.

Algorithm 1 The pseudo-code of DOCREL framework for forecasting wind power data-points.

```
**Input**: $Max_{it}$ (Maximum iteration number), $P_s$ (Population size), $CL$ (Number of CLSTM models).
**Output**: The predicted values of wind power datasets.
**Begin algorithm**:
Separate the wind power data points into training $Tr$ and testing $Te$ sets;
**while** ($i < CL$) **do**
    Set $P_s$ solutions as the initial population in which each solution is a CLSTM model with random values of hyperparameters;
    Calculate fitness value of each solution using Eq.(27) as the MSE of related CLSTM model found by $Tr$ set;
    **while** ($t < Max_{iter}$) **do**
        **for** each search agent **do**
            Update $a$, $A$, $C$ and $p$;
            **if** (p<0.5) **then**
                **if** (|A|<1) **then**
                    Update the position of current search agent by quasi-opposition strategy based on Eq.(19);
                **else if** (|A|≥1) **then**
                    Update the position of the search agent by the Eq.(16);
                **end if**
            **else if** (p≥0.5) **then**
                Update the position of the search agent by the Eq.(14);
            **end if**
        **end for**
        Update the current search agent position using the Lévy flight trajectory;
        Calculate the fitness of all search agents and update $X^*$ if a better solution is found;
    **end while**
    Set CLSTM model $i$ according to the hyperparameters values of the best obtained solution;
**end while**
Perform deep Q-learning model to obtain the weights of CLSTM models as $[w_1, w_2, ..., w_i]$;
Ensemble the results of CLSTM models based on the obtained weights to forecast the wind power points in the test set $Te$;
Return the predicted wind power points as the output;
**End algorithm**
```

We represent the input vectors of the deep CLSTM neural network using the training data of Eq. (25), while the CLSTM predicts the wind power values from the next N time-steps given in Eq. (26). Thus, we assess the fitness values of each BWOA solution using the well-known mean square error (MSE) as the fitness function using the following formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2 \tag{27}$$

where y_i and y_i' represent the actual and forecasted wind power values obtained by the CLSTM model, respectively.

As the proposed BWOA-CLSTM looks for the solution with the lowest MSE values, this mechanism achieves the deep CLSTM neural networks having the best and highest performance of the testing set for the data points of wind power forecasting problem.

- Phase 3: By obtaining the best deep CLSTM neural network models using our novel evolutionary algorithm, the last step of our hybrid algorithm is to use the powerful deep reinforcement learning theory. Reinforcement learning is an online learning methodology that examines the optimal decision making procedure by means of trials and errors, and then produces an optimal outcome during interactions with the environment. Two most common reinforcement learning algorithms are Q-learning and SARSA algorithms, which have been successfully implemented in several real-world applications. Taking into account the strong convergence of the Q-learning algorithm, this technique is used in this work as an ensemble learning strategy for integrating the optimized deep CLSTM networks.

First, we formulate the concepts of state matrix S and the action matrix a. The state matrix represents the weight in the ensemble of the optimized deep CLSTM networks and the action matrix denotes to the weight estimation of the action matrix. These two basic concepts in Q-learning are given by the following formulas:

$$S = [w_1, w_2, ..., w_i] \; a = [\Delta w_1, \Delta w_2, ..., \Delta w_i] \tag{28}$$

where, w_i represents the weight of the ith optimized deep CLSTM model. Δw_i means the changing weights of the deep optimized models.

Next, based on the training sets of the optimized deep CLSTM models, we train the agent in which the agent carries out action a based on the greedy policy in compliance with the current state S.

$$a_m = \left\{ \begin{array}{l} \text{Based on maxQ (probability of } 1-\varepsilon) \\ \text{Random(probability of } \varepsilon \text{)} \end{array} \right. \quad \varepsilon \in (0,1) \tag{29}$$

In the third step, we should determine the error function L and the reward R function. We consider MSE as the error function and the reward function is calculated by:

$$R = \left\{ \begin{array}{l} +1 + L_m - L_{m+1} \left(L_{m+1} < L_m\right) \\ -1 + L_m - L_{m+1} \left(L_{m+1} > L_m\right) \end{array} \right. \tag{30}$$

In forth step, we should assess the evaluation function Q by updating the Q table by:

$$\begin{aligned} Q_{m+1}\left(S_m, a_m\right) &= Q_m\left(S_m, a_m\right) \\ &+ \gamma_m \left(R\left(S_m, a_m\right) + \lambda \max Q_m\left(S_{m+1}, a_{m+1}\right) - Q_m\left(S_m, a_m\right)\right) \end{aligned} \tag{31}$$

where γ represents the learning rate and the discount parameter is denoted by λ. We name this advanced framework as DOCLER (**D**eep **O**ptimized **C**onvolutional **L**stm based **E**nsemble **R**einforcement learning strategy). The major stages of the DOCLER model can be outlined in the pseudo code given by Algorithm 1.

TABLE I: List of the primary CLSTM architecture hyperparameters

Hyperparameters	Symbol
Number of convolutional layers	N_c
Number of filters in each convolutional layer	$fN_{i,i\in N_c}$
Kernel size in each convolutional layer	$kS_{i,i\in N_c}$
Number of pooling layers	N_p
Dropout	R_{dropout}
Batch size	S_{batch}
Learning rate	Lrate
Number of LSTM layers	N_l
Maximum epoch	Max_e
Neural unites in each LSTM layer	$uN_{i,i\in N_l}$
LSTM batch size	lS_{batch}

TABLE II: List of expressions and valuations of CLSTM hyperparameters

Expression	Valuation List
$N_c = \psi$	$\overline{N_c(\psi)}_{\psi\in[1,8]} \in [1,2,3,...]$
$fN_{1<i<N_c} = 2^{\psi+2}$	$\overline{fN(\psi)}_{\psi\in[1,6]} \in [8,16,32,\cdots]$
$kS_{1<i<N_c} = (2\times\psi)+1$	$\overline{kS(\psi)}_{\psi\in[1,5]} \in [3,5,7,9,11]$
$N_p = \chi$	$\overline{N_p(\chi)}_{\chi\in[1,3]} \in [1,2,3]$
$R_{\text{dropout}} = (\psi+3)\times 0.05$	$\overline{R_{\text{dropout}}}(\psi)_{\psi\in[1,8]} \in [0.2,0.25,\cdots]$
$S_{\text{batch}} = 10\times\psi$	$\overline{S_{\text{batch}}}(\gamma)_{\gamma E[1,10]} \in [10,20,\cdots]$
$L_{\text{rate}} = 0.001 + 0.005\times(\psi-1)$	$\overline{L_{\text{rate}}}(\psi)_{\psi\in[1,21]} \in [0.001,0.006,\cdots]$
$N_l = \eta$	$\overline{N_l(\eta)}_{\eta\in[1,3]} \in [1,2,3]$
Max_e	$\overline{Max_e}(\eta)_{\eta\in[1,500]} \in [1,2,...]$
$uN_{i,i\in N_l}$	$\overline{N_c(\psi)}_{\psi\in[1,300]} \in [1,2,3,...]$
$lS_{\text{batch}} = 10\times\eta$	$\overline{lS_{\text{batch}}}(\eta)_{\eta[1,3]} \in [10,20,30]$

III. Wind Power Datasets and Experimental Setups

The Sotavento wind farms (43.354377°N, 7.881213°W, 644m.a.s.l.) is employed in this paper as the utilized datasets for the evaluation of our proposed deep learning algorithm. The wind farm of Sotavento is situated in Galicia, Spain. This wind farm comprises of 24 wind turbines with a power generation of 17,56 MW. The total historical power generation data of this wind farm can be obtained from the website by considering the resolution of half-hour. The accompanying data can be collected from the Meteogalicia numerical weather prediction mechanism, depending on the situation of the Sotavento wind farm. Throughout this study, the total historical wind power data for both the years of 2019 and 2020 were collected for examining the DOCREL framework. Each wind power database comprises 17568 GHI time-series data points in an interval of half-hour. We first split the data based on four different seasons for both of the 2019 and 2020 datasets. Then, we consider 75% of each dataset to represent the training set and the remaining 25% to represent the testing set. As mentioned before, We use the mutual information strategy (MI) to choose the input characteristics for the optimized deep CNN models which results in 39 wind power input characteristics in order to train the deep learning models. For implementing the DOCREL framework, we use the advanced and intelligent libraries for deep learning, including Keras and TensorFlow with the hardware devices of 32 GB RAM, Intel Core i7 CPU, two GeForce GTX 1080 Ti GPUs as well the Ubuntu operation system. To execute this proposed methodology, we program it using Python language programming version 3.7. We configure the proposed BWOA as the part of CLSTM deep neural architecture search with the number of population and maximum iteration number equal to 40 and 30, respectively. The deep Q-learning algorithm is configured as maximum iteration of 40, learning rate of 0.95 and discount parameter equal to 0.5. It should be mentioned that we feed 10 optimized CLSTM models to the deep RL ensemble strategy. In

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2021.3126272, IEEE Transactions on Industry Applications

TABLE III: The experimental findings of four seasons for the Spanish dataset gathered in the year 2019

Category	Model name	Seasons							
		Spring		Summer		Autumn		Winter	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Deep learning models	LSTM	0.084883	0.062442	0.086973	0.063527	0.086104	0.062947	0.084418	0.061981
	Bi-LSTM	0.084452	0.062052	0.086732	0.063286	0.085775	0.062714	0.084313	0.061666
	CNN-BiLSTM	0.084306	0.061886	0.086342	0.063096	0.085673	0.062504	0.083991	0.061554
	EnsemLSTM	0.083773	0.061353	0.086176	0.06283	0.085407	0.062278	0.083529	0.061092
	hybridDeep	0.083172	0.060752	0.085643	0.062297	0.084784	0.061715	0.082993	0.060556
	WD-LSTM	0.082882	0.060462	0.085042	0.061696	0.084273	0.061313	0.082639	0.060202
	SIRAE	0.082613	0.060398	0.084752	0.061506	0.084083	0.061123	0.082317	0.05988
Hyperparameter searching models	BWOA-CNN-BiLSTM	0.082551	0.060131	0.084421	0.060985	0.083562	0.060602	0.082127	0.059565
	BWOA-LSTM	0.082328	0.059908	0.084198	0.060652	0.08312	0.06016	0.081439	0.059002
	Bayesian-CNN-BiLSTM	0.081461	0.059052	0.083342	0.060096	0.082673	0.059713	0.081058	0.058621
RL-based deep learning models	Bayesian-CLSTM-RL	0.080472	0.058461	0.082751	0.059405	0.081982	0.059135	0.080428	0.057768
	GA-CLSTM-RL	0.080376	0.057956	0.082246	0.0589	0.081545	0.058491	0.080092	0.057655
	PSO-CLSTM-RL	0.079572	0.057152	0.081442	0.057987	0.080564	0.057604	0.079004	0.056979
	WOA-CLSTM-RL	0.078983	0.056284	0.080574	0.0575	0.080005	0.056837	0.078025	0.055588
	Proposed DOCREL	0.07534	0.05292	0.07721	0.053864	0.076441	0.053481	0.074871	0.052434

TABLE IV: The experimental findings of four seasons for the Spanish dataset gathered in the year 2020

Category	Model name	Seasons							
		Spring		Summer		Autumn		Winter	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Deep learning models	LSTM	0.083271	0.059627	0.08367	0.06112	0.081994	0.058256	0.080535	0.056168
	Bi-LSTM	0.083018	0.059374	0.083448	0.060921	0.081795	0.058209	0.080488	0.055939
	CNN-BiLSTM	0.082906	0.059262	0.083238	0.060777	0.081651	0.058066	0.080444	0.055826
	EnsemLSTM	0.082502	0.058698	0.082671	0.060211	0.081085	0.057499	0.079635	0.055026
	hybridDeep	0.082018	0.058374	0.082293	0.059833	0.080786	0.057057	0.079435	0.054826
	WD-LSTM	0.081554	0.05771	0.081683	0.059223	0.080219	0.056633	0.079012	0.054434
	SIRAE	0.081232	0.057588	0.081437	0.059044	0.07994	0.056298	0.078677	0.054068
Hyperparameter searching models	BWOA-CNN-BiLSTM	0.080917	0.057273	0.081247	0.058786	0.07976	0.056175	0.078553	0.053944
	BWOA-LSTM	0.080353	0.056628	0.080761	0.0583	0.079175	0.055454	0.077656	0.052825
	Bayesian-CNN-BiLSTM	0.0799	0.056256	0.08023	0.057968	0.078674	0.055088	0.077466	0.052857
RL-based deep learning models	Bayesian-CLSTM-RL	0.079119	0.055419	0.079529	0.057068	0.077942	0.054615	0.076867	0.052325
	GA-CLSTM-RL	0.078918	0.055274	0.079248	0.056787	0.077797	0.054211	0.07659	0.051981
	PSO-CLSTM-RL	0.078331	0.054687	0.078661	0.056332	0.077206	0.05362	0.075834	0.050855
	WOA-CLSTM-RL	0.076939	0.05323	0.077259	0.054901	0.075684	0.052093	0.074736	0.049925
	Proposed DOCREL	0.073786	0.050142	0.074116	0.051655	0.072529	0.048943	0.071322	0.046713

order to have a fair comparison, the proposed DOCLER model and the other competitive deep learning algorithms execute with 10 independent runs.

Besides, the information regarding the CLSTM hyperparameters and their corresponding ranges are represented in Tables I and II which are optimized in the process of neural architecture search using the BWOA model. In addition to these evolved hyperparameters, the Relu as the activation function and Adam as the optimizer are selected during the training procedure.

The root mean square error (RMSE) and mean absolute error (MAE) as the frequently used performance indicators in the scholarly literature are considered to determine the accuracy of the deep learning forecasting algorithms. The formulas for these two evaluation metrics are as following:

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i' - y_i)^2} \tag{32}$$

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i' - y_i| \tag{33}$$

where y_i' denotes the predicted value of y_i, where n denotes the quantity of data points in the testing test.

In order to show how strong the performance of our proposed algorithm is, we compare it with the latest deep learning algorithms used for wind power forecasting in the literature. These state of the art deep learning models include LSTM, Bi-LSTM, CNN-BiLSTM, EnsemLSTM [54], hybridDeep [55], WD-LSTM [56] and stacked by independent recurrent autoencoder (SIRAE) [57], which have shown acceptable performance for wind power forecasting in the literature. We also test the effect of powerful hyperparameter searching models including BWOA-CNN-BiLSTM, BWOA-LSTM, and Bayesian-CNN-BiLSTM for the deep learning models and also the effect of different searching models for RL-ensemble based CLSTM algorithms including Bayesian, genetic algorithm (GA), particle swarm optimization (PSO), and the standard version of WOA algorithm.

IV. Experimental Results

In this section, we seasonally examine the performance of our proposed algorithm against 14 powerful deep learning algorithms on the 2019 and 2020 datasets. We also divide the experiments into (1) comparison of deep learning models, i.e., LSTM, Bi-LSTM, CNN-BiLSTM, EnsemLSTM, hybridDeep, WD-LSTM, and SIRAE (2) comparison of hyperparameter searching algorithms including BWOA-CNN-BiLSTM, BWOA-LSTM

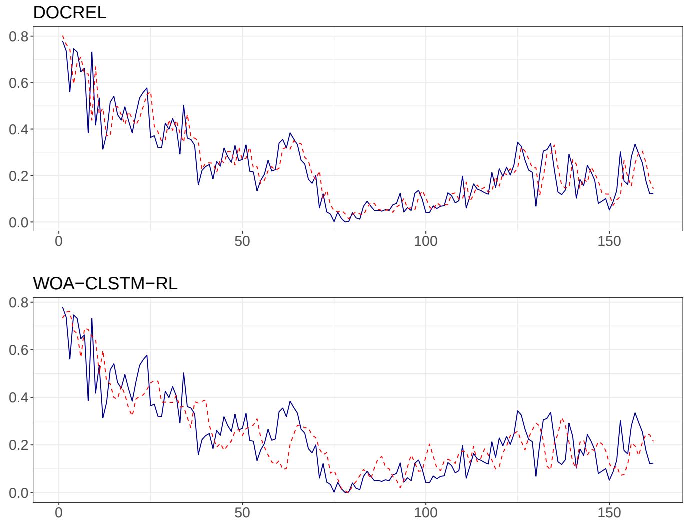

Fig. 3: Actual (indicated by blue) vs predicted (indicated by red) data points for the year 2019 dataset.

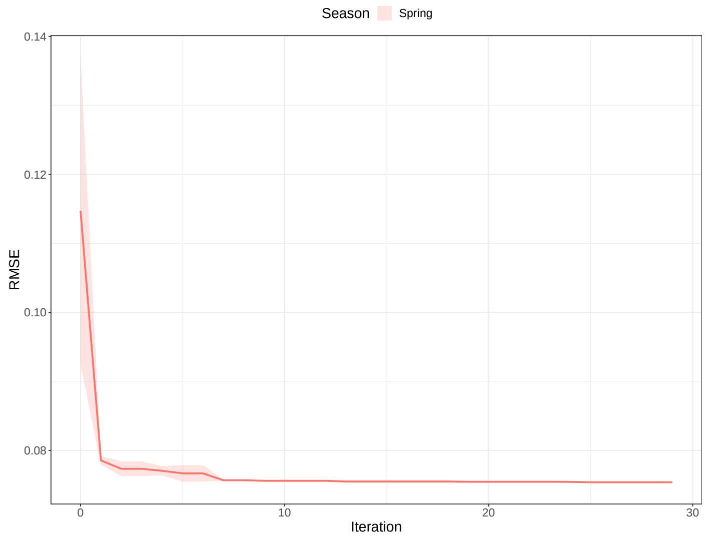

Fig. 5: The convergence curve of the DOCREL model for the collected wind farm of the year 2019.

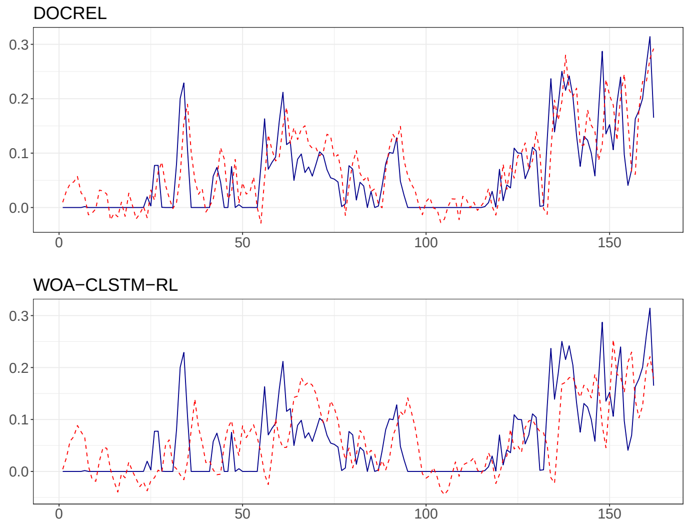

Fig. 4: Actual (indicated by blue) vs predicted (indicated by red) data points for the year 2020 dataset.

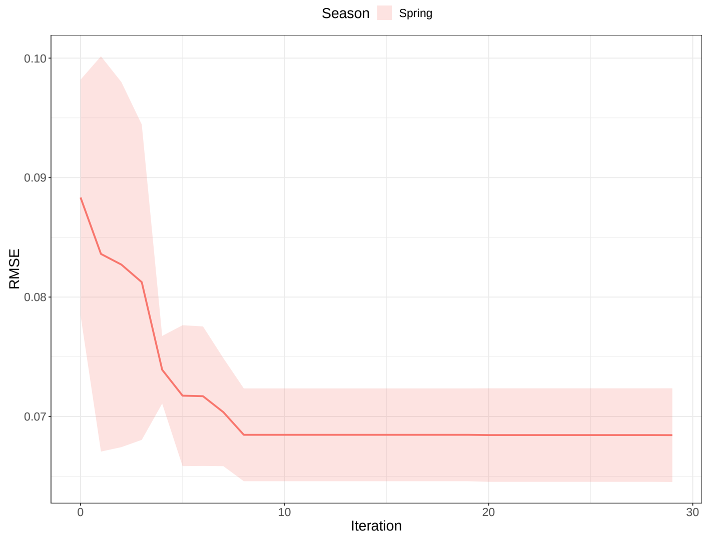

Fig. 6: The convergence curve of the DOCREL model for the collected wind farm of the year 2020.

and Bayesian-CNN-BiLSTM, (3) comparison of reinforcement learning-based ensemble algorithms. Tables III and IV show the performance of our proposed DOCREL algorithm and other benchmarked methods based on two evaluation criteria, RMSE and MAE for four different seasons of the years of 2019 and 2020. As can be seen from these tables, the proposed DOCREL model for four seasons of the years of 2019 and 2020 has the lowest error values among the different benchmarked algorithms for the wind power datasets. A closer look at these tables reveals that the closest follower to our proposed algorithm is the WOA-CLSTM-RL (the standard version of WOA), which for the various seasons of 2019 and 2020, has the lowest RMSE and MAE error rates among all benchmarked algorithms for two wind power datasets. Figs. 3 and 4 demonstrate the actual and predicted wind power time series datasets for our proposed DOCREL model versus the second best model (WOA-CLSTM-RL) for the spring season of the years of 2019 and 2020. As can be seen from these figures, the real and predicted wind power time series datasets are well superimposed by DOCREL algorithm, which shows the high ability of our proposed model in seasonally forecasting of two different wind power datasets. The WOA-CLSTM-RL as the most competitive algorithm compared with DOCREL can not efficiently meet the mapping of the actual and predicted points for both the 2019 and 2020 wind power datasets.

Furthermore, the convergence curves for two wind power datasets of the spring season by our proposed algorithm is displayed on 30 iterations based on the fitness function in Figs. 5 and 6. As can be seen from both of these figures, the proposed DOCREL algorithm converges excellently and efficiently for two different Spanish wind power datasets.

In order to represent how effective the proposed DOCREL framework is in selecting the initial hyperparameters in the optimization process, we show all eleven CLSTM hyperparameters used by the proposed evolutionary algorithm in Figs. 7 and 8 using the violin plots. The dots in these figures represent each observation of the initialized value for each hyperparameter. Based on the results shown in these two figures, the proposed framework in most hyperparameters has selected values close to

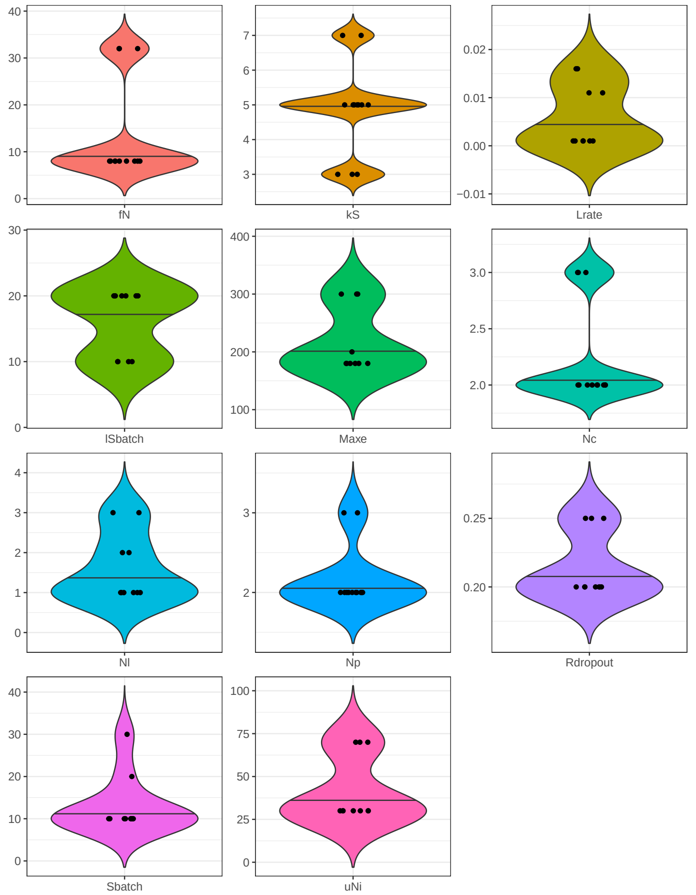

Fig. 7: The violin plots of eleven evolved hyperparameters for the year 2019.

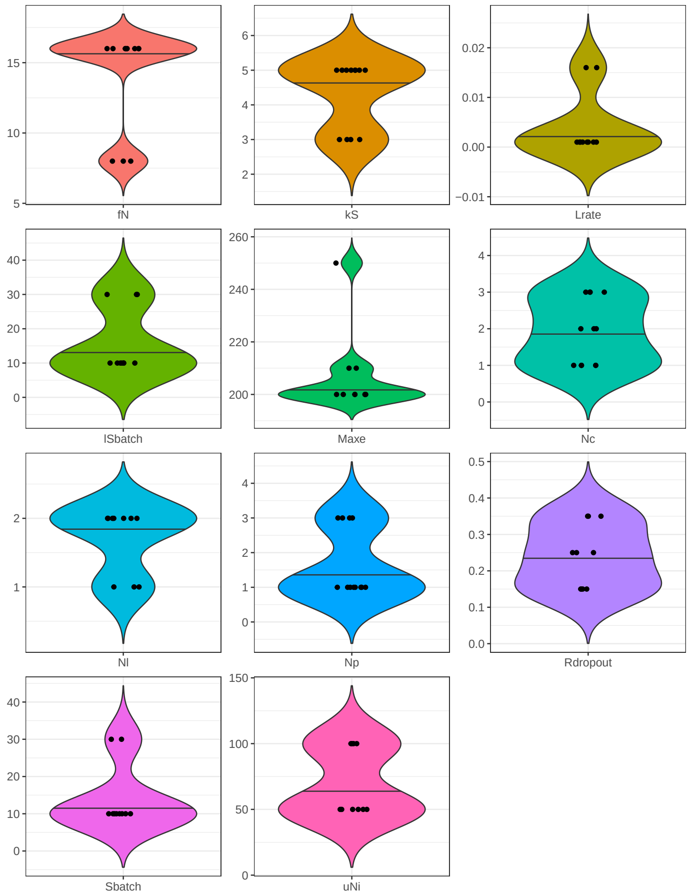

Fig. 8: The violin plots of eleven evolved hyperparameters for the year 2020.

the selected minimum intervals shown in Table II, which shows the low computational ability of this algorithm in selecting CLSTM hyperparameters during the evolutionary procedure. For instance, by a closer look at the N_c hyperparameter for the year 2019, we figure out that most of the N_c values selected by DOCREL are with values equal to 2.

TABLE V: Run-time analysis for the proposed DOCREL model and other different benchmarks

Model	Time (in seconds)			Num of parameters
	Optimization time	Training time	Testing time	
LSTM	-	354	84	404K
Bi-LSTM	-	371	98	442K
CNN-BiLSTM	-	375	103	469K
EnsemLSTM	-	389	114	492K
hybridDeep	-	383	108	483K
WD-LSTM	-	363	87	421K
SIRAE	-	369	94	435K
BWOA-CNN-BiLSTM	831	211	56	349K
BWOA-LSTM	792	204	52	331K
Baysian-CNN-BiLSTM	938	322	81	402K
Baysian-CLSTM-RL	916	305	73	394K
GA-CLSTM-RL	896	276	67	385K
PSO-CLSTM-RL	874	251	65	362K
WOA-CLSTM-RL	843	226	61	354K
Proposed	778	191	47	328K

In terms of statistics for demonstrating the strength of the DOCREL algorithm compared to the benchmarked deep learning algorithms, we show all these methods for the two Spanish wind power databases for the years of 2019 and 2020, using the box plot tool. As shown in Fig. 9 and 10, the box plots of our proposed method has the least amount of errors as well as the most compact box in terms of comparison with other deep learning methods. It is worth noting that these results are obtained based on 10 independent runs.

In order to show the time spent for the proposed algorithm and other benchmarked algorithms, we show the three times used to run the algorithms including optimization, training and testing times as well as the the number of parameters (i.e., the number of weights in the connected layers) which can be seen in Table V. Also, it should be noted that the optimization time is specific to categories of hyperparameter searching models as well as RL-based deep learning models because they use optimization elements in their architectures. Table V shows that our proposed DOCREL algorithm has the lowest consumption time based on the seconds among the algorithms that use the optimization element for their architecture. DOCREL also has the least amount of consumption time in terms of testing and training times in comparison to all of the benchmarked algorithms. Furthermore, with having the number of parameters equal to 328K, our proposed model considers less parameters in comparison to other deep-learning algorithms for wind power forecasting.

V. Conclusion

We introduced a deep learning-based hybrid framework in this work by developing the three phases involving MI strategy as the efficient characteristic extraction strategy, the optimized deep CLSTM based on the modified whale optimization algorithm to perform the neural architectural search procedure, and the

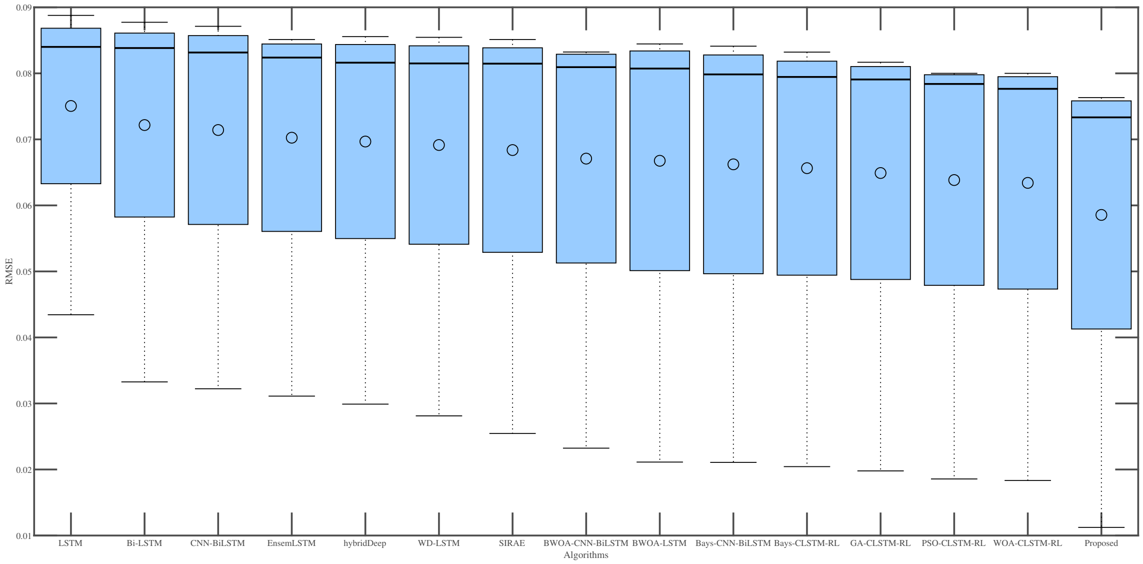

Fig. 9: Boxplots of various algorithms based on RMSE for the collected wind farm of the year 2019.

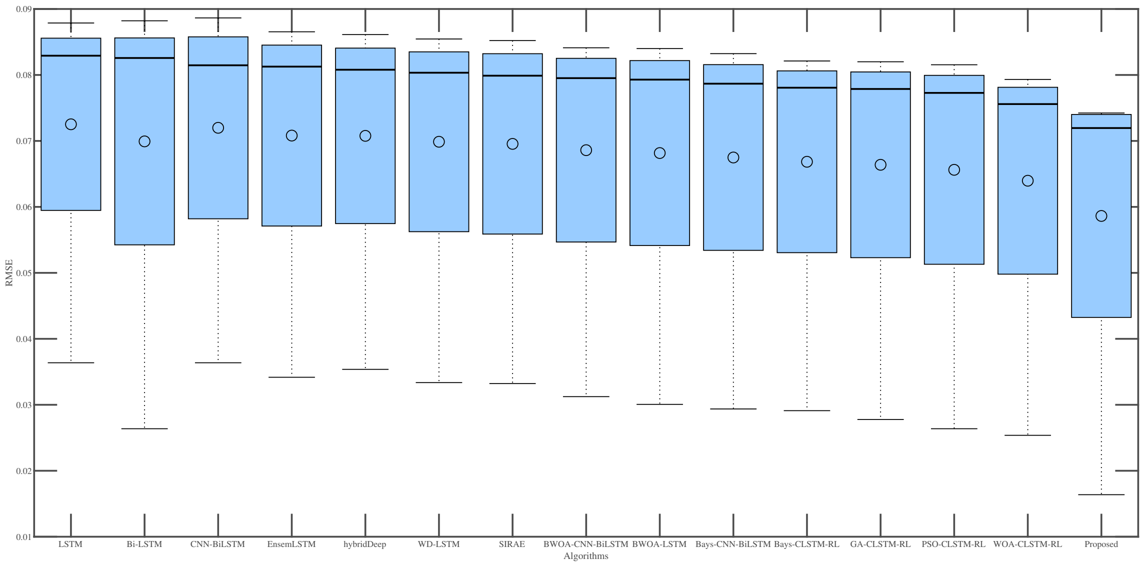

Fig. 10: Boxplots of various algorithms based on RMSE for the collected wind farm of the year 2020.

ensemble of deep Q-learning reinforcement learning model called DOCREL for wind power forecasting. The wind power data points obtained from a Spanish wind farm for the years of 2019 and 2020 are used to verify the effectiveness of the hybrid DOCREL framework. The experimental results showed the excellent performance of our proposed DOCREL algorithm in comparison to several state-of-the-art models for wind power forecasting.

References

[1] S. Arango-Aramburo, J. Ríos-Ocampo, and E. Larsen, "Examining the decreasing share of renewable energy amid growing thermal capacity: The case of south america," *Renewable and Sustainable Energy Reviews*, vol. 119, p. 109648, 2020.

[2] J. Li, S. Chen, Y. Wu, Q. Wang, X. Liu, L. Qi, X. Lu, and L. Gao, "How to make better use of intermittent and variable energy? a review of wind and photovoltaic power consumption in china," *Renewable and Sustainable Energy Reviews*, vol. 137, p. 110626, 2021.

[3] P. Scarabaggio, S. Grammatico, R. Carli, and M. Dotoli, "Distributed demand side management with stochastic wind power forecasting," *IEEE Transactions on Control Systems Technology*, 2021.

[4] G. J. Osório, M. Lotfi, V. M. Campos, and J. P. Catalão, "Extended hybrid wind power forecasting approach to short-term decisions," in *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*. IEEE, 2020, pp. 1–6.

[5] H. Quan, A. Khosravi, D. Yang, and D. Srinivasan, "A survey of computational intelligence techniques for wind power uncertainty quantification in smart grids," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 11, pp. 4582–4599, 2019.

[6] A. Ahmed and M. Khalid, "A review on the selected applications of forecasting models in renewable power systems," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 9–21, 2019.

[7] M. Khodayar, O. Kaynak, and M. E. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 2770–2779, 2017.

[8] M. Khodayar and J. Wang, "Spatio-temporal graph deep neural network for short-term wind speed forecasting," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 2, pp. 670–681, 2018.

[9] C. Stathopoulos, A. Kaperoni, G. Galanis, and G. Kallos, "Wind power prediction based on numerical and statistical models," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 112, pp. 25–38, 2013.

[10] S. Ahmadian and A. R. Khanteymoori, "Training back propagation neural networks using asexual reproduction optimization," in *7th Conference on Information and Knowledge Technology (IKT)*. IEEE, 2015, pp. 1–6.

[11] S. M. J. Jalali, S. Ahmadian, A. Khosravi, S. Mirjalili, M. R. Mahmoudi, and S. Nahavandi, "Neuroevolution-based autonomous robot navigation: A comparative study," *Cognitive Systems Research*, vol. 62, pp. 35–43, 2020.

[12] S. M. J. Jalali, S. Ahmadian, P. M. Kebria, A. Khosravi, C. P. Lim, and S. Nahavandi, "Evolving artificial neural networks using butterfly optimiza-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIA.2021.3126272, IEEE Transactions on Industry Applications

tion algorithm for data classification," in *International Conference on Neural Information Processing (ICONIP)*. Springer, 2019, pp. 596–607.

[13] S. M. J. Jalali, M. Khodayar, S. Ahmadian, M. Shafie-khah, A. Khosravi, S. M. S. Islam, S. Nahavandi, and J. P. Catalão, "A new ensemble reinforcement learning strategy for solar irradiance forecasting using deep optimized convolutional neural network models," in *2021 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, 2021, pp. 1–6.

[14] S. J. Mousavirad, S. M. J. Jalali, S. Ahmadian, A. Khosravi, G. Schaefer, and S. Nahavandi, "Neural network training using a biogeography-based learning strategy," in *International Conference on Neural Information Processing*. Springer, 2020, pp. 147–155.

[15] M. R. C. Qazani, S. M. J. Jalali, H. Asadi, and S. Nahavandi, "Optimising control and prediction horizons of a model predictive control-based motion cueing algorithm using butterfly optimization algorithm," in *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.

[16] S. M. J. Jalali, A. Khosravi, R. Alizadehsani, S. M. Salaken, P. M. Kebria, R. Puri, and S. Nahavandi, "Parsimonious evolutionary-based model development for detecting artery disease," in *2019 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2019, pp. 800–805.

[17] V. Chandran, C. K. Patil, A. M. Manoharan, A. Ghosh, M. Sumithra, A. Karthick, R. Rahim, and K. Arun, "Wind power forecasting based on time series model using deep machine learning algorithms," *Materials Today: Proceedings*, 2021.

[18] N. A. Treiber, J. Heinermann, and O. Kramer, "Wind power prediction with machine learning," in *Computational sustainability*. Springer, 2016, pp. 13–29.

[19] Z. Li, L. Ye, Y. Zhao, X. Song, J. Teng, and J. Jin, "Short-term wind power prediction based on extreme learning machine with error correction," *Protection and Control of Modern Power Systems*, vol. 1, no. 1, pp. 1–8, 2016.

[20] J. Zhao, Z.-H. Guo, Z.-Y. Su, Z.-Y. Zhao, X. Xiao, and F. Liu, "An improved multi-step forecasting model based on wrf ensembles and creative fuzzy systems for wind speed," *Applied Energy*, vol. 162, pp. 808–826, 2016.

[21] G. Galanis, E. Papageorgiou, and A. Liakatas, "A hybrid bayesian kalman filter and applications to numerical wind speed modeling," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 167, pp. 1–22, 2017.

[22] J.-Z. Wang, Y. Wang, and P. Jiang, "The study and application of a novel hybrid forecasting model–a case study of wind speed forecasting in china," *Applied Energy*, vol. 143, pp. 472–488, 2015.

[23] K. Wang, X. Qi, H. Liu, and J. Song, "Deep belief network based k-means cluster approach for short-term wind power forecasting," *Energy*, vol. 165, pp. 840–852, 2018.

[24] S. M. J. Jalali, S. Ahmadian, M. Khodayar, A. Khosravi, V. Ghasemi, M. Shafie-khah, S. Nahavandi, and J. P. Catalão, "Towards novel deep neuroevolution models: chaotic levy grasshopper optimization for short-term wind speed forecasting," *Engineering with Computers*, pp. 1–25, 2021.

[25] M. Khodayar, M. E. Khodayar, and S. M. J. Jalali, "Deep learning for pattern recognition of photovoltaic energy generation," *The Electricity Journal*, vol. 34, no. 1, p. 106882, 2021.

[26] S. M. J. Jalali, P. M. Kebria, A. Khosravi, K. Saleh, D. Nahavandi, and S. Nahavandi, "Optimal autonomous driving through deep imitation learning and neuroevolution," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 1215–1220.

[27] S. M. J. Jalali, S. Ahmadian, A. Khosravi, M. Shafie-khah, S. Nahavandi, and J. P. Catalão, "A novel evolutionary-based deep convolutional neural network model for intelligent load forecasting," *IEEE Transactions on Industrial Informatics*, pp. 1–8, 2021.

[28] S. M. J. Jalali, M. Ahmadian, S. Ahmadian, A. Khosravi, M. Alazab, and S. Nahavandi, "An oppositional-cauchy based gsk evolutionary algorithm with a novel deep ensemble reinforcement learning strategy for covid-19 diagnosis," *Applied Soft Computing*, vol. 111, p. 107675, 2021.

[29] Q. Wu, F. Guan, C. Lv, and Y. Huang, "Ultra-short-term multi-step wind power forecasting based on cnn-lstm," *IET Renewable Power Generation*, vol. 15, no. 5, pp. 1019–1029, 2021.

[30] S. Aslam, H. Herodotou, S. M. Mohsin, N. Javaid, N. Ashraf, and S. Aslam, "A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids," *Renewable and Sustainable Energy Reviews*, vol. 144, p. 110992, 2021.

[31] Y. Ju, G. Sun, Q. Chen, M. Zhang, H. Zhu, and M. U. Rehman, "A model combining convolutional neural network and lightgbm algorithm for ultra-short-term wind power forecasting," *IEEE Access*, vol. 7, pp. 28 309–28 318, 2019.

[32] S. Mujeeb, T. A. Alghamdi, S. Ullah, A. Fatima, N. Javaid, and T. Saba, "Exploiting deep learning for wind power forecasting based on big data analytics," *Applied Sciences*, vol. 9, no. 20, p. 4417, 2019.

[33] H. Zhen, D. Niu, M. Yu, K. Wang, Y. Liang, and X. Xu, "A hybrid deep learning model and comparison for wind power forecasting considering temporal-spatial feature extraction," *Sustainability*, vol. 12, no. 22, p. 9490, 2020.

[34] Y. Zhang, S. Zhou, Z. Zhang, L. Yan, and L. Liu, "Design of an ultra-short-term wind power forecasting model combined with cnn and lstm networks," in *International Conference on Intelligent Computing, Communication & Devices*. Springer, 2019, pp. 141–145.

[35] C. Yildiz, H. Acikgoz, D. Korkmaz, and U. Budak, "An improved residual-based convolutional neural network for very short-term wind power forecasting," *Energy Conversion and Management*, vol. 228, p. 113731, 2021.

[36] H. Shao, X. Deng, and Y. Jiang, "A novel deep learning approach for short-term wind power forecasting based on infinite feature selection and recurrent neural network," *Journal of Renewable and Sustainable Energy*, vol. 10, no. 4, p. 043303, 2018.

[37] Z. Lin, X. Liu, and M. Collu, "Wind power prediction based on high-frequency scada data along with isolation forest and deep learning neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 118, p. 105835, 2020.

[38] H. Hu, L. Wang, and S.-X. Lv, "Forecasting energy consumption and wind power generation using deep echo state network," *Renewable Energy*, vol. 154, pp. 598–613, 2020.

[39] R. Meka, A. Alaeddini, and K. Bhaganagar, "A robust deep learning framework for short-term wind power forecast of a full-scale wind farm using atmospheric variables," *Energy*, vol. 221, p. 119759, 2021.

[40] S. M. J. Jalali, S. Ahmadian, A. Kavousi Fard, A. Khosravi, and S. Nahavandi, "Automated deep cnn-lstm architecture design for solar irradiance forecasting," *IEEE Transactions on Systems, Man, and Cybernetics - Systems*, vol. 1, pp. 1–12, 2021.

[41] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[42] M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," in *Proceedings of the genetic and evolutionary computation conference*, 2017, pp. 497–504.

[43] J. F. Torres, D. Gutiérrez-Avilés, A. Troncoso, and F. Martínez-Álvarez, "Random hyper-parameter search-based deep neural network for power consumption forecasting," in *International Work-Conference on Artificial Neural Networks*. Springer, 2019, pp. 259–269.

[44] H. Cuayáhuitl, D. Lee, S. Ryu, Y. Cho, S. Choi, S. Indurthi, S. Yu, H. Choi, I. Hwang, and J. Kim, "Ensemble-based deep reinforcement learning for chatbots," *Neurocomputing*, vol. 366, pp. 118–130, 2019.

[45] A. Antwi, K. A. Kyei, and R. S. Gill, "The use of mutual information to improve value-at-risk forecasts for exchange rates," *IEEE Access*, vol. 8, pp. 179 881–179 900, 2020.

[46] M. Tzelepi and A. Tefas, "Improving the performance of lightweight cnns for binary classification using quadratic mutual information regularization," *Pattern Recognition*, vol. 106, p. 107407, 2020.

[47] T. Fernando, H. Maier, and G. Dandy, "Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach," *Journal of Hydrology*, vol. 367, no. 3-4, pp. 165–176, 2009.

[48] X. Liu, H. Zhang, X. Kong, and K. Y. Lee, "Wind speed forecasting using deep neural network with feature selection," *Neurocomputing*, vol. 397, pp. 393–403, 2020.

[49] A. Hacine-Gharbi, P. Ravier, R. Harba, and T. Mohamadi, "Low bias histogram-based estimation of mutual information for feature selection," *Pattern recognition letters*, vol. 33, no. 10, pp. 1302–1308, 2012.

[50] T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx, "A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 62–77, 2020.

[51] H. Zang, L. Liu, L. Sun, L. Cheng, Z. Wei, and G. Sun, "Short-term global horizontal irradiance forecasting based on a hybrid cnn-lstm model with spatiotemporal correlations," *Renewable Energy*, vol. 160, pp. 26–41, 2020.

[52] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using cnn-lstm neural networks," *Energy*, vol. 182, pp. 72–81, 2019.

[53] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.

[54] J. Chen, G.-Q. Zeng, W. Zhou, W. Du, and K.-D. Lu, "Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization," *Energy Conversion and Management*, vol. 165, pp. 681–695, 2018.

[55] Y.-Y. Hong and C. L. P. P. Rioflorido, "A hybrid deep learning-based neural network for 24-h ahead wind power forecasting," *Applied Energy*, vol. 250, pp. 530–539, 2019.

[56] B. Liu, S. Zhao, X. Yu, L. Zhang, and Q. Wang, "A novel deep learning approach for wind power forecasting based on wd-lstm model," *Energies*, vol. 13, no. 18, p. 4964, 2020.

[57] L. Wang, R. Tao, H. Hu, and Y.-R. Zeng, "Effective wind power prediction using novel deep learning network: Stacked independently recurrent autoencoder," *Renewable Energy*, vol. 164, pp. 642–655, 2021.