



Vaasan yliopisto  
UNIVERSITY OF VAASA

OSUVA Open  
Science

This is a self-archived – parallel published version of this article in the publication archive of the University of Vaasa. It might differ from the original.

## Towards novel deep neuroevolution models: chaotic levy grasshopper optimization for short-term wind speed forecasting

**Author(s):** Jalali, Seyed Mohammad Jafar; Ahmadian, Sajad; Khodayar, Mahdi; Khosravi, Abbas; Ghasemi, Vahid; Shafie-khah, Miadreza; Nahavandi, Saeid; Catalão, João P. S.

**Title:** Towards novel deep neuroevolution models: chaotic levy grasshopper optimization for short-term wind speed forecasting

**Year:** 2022

**Version:** Accepted manuscript

**Copyright** ©2022 Springer. This is a post-peer-review, pre-copyedit version of an article published in *Engineering with Computers*. The final authenticated version is available online at:  
<http://dx.doi.org/10.1007/s00366-021-01356-0>

### Please cite the original version:

Jalali, S. M. J., Ahmadian, S., Khodayar, M., Khosravi, A., Ghasemi, V., Shafie-khah, M., Nahavandi, S. & Catalão, J. P. S. (2022). Towards novel deep neuroevolution models: chaotic levy grasshopper optimization for short-term wind speed forecasting. *Engineering with Computers* 38(Suppl 3), 1787–1811. <https://doi.org/10.1007/s00366-021-01356-0>

# Towards novel deep neuroevolution models: chaotic levy grasshopper optimization for short-term wind speed forecasting

Seyed Mohammad Jafar Jalali · Sajad Ahmadian · Mahdi Khodayar · Abbas Khosravi ·

Vahid Ghasemi · Miadreza Shafie-khah · Saeid Nahavandi · João P. S. Catalão

## Abstract

High accurate wind speed forecasting plays an important role in ensuring the sustainability of wind power utilization. Although deep neural networks (DNNs) have been recently applied to wind time-series datasets, their maximum performance largely leans on their designed architecture. By the current state-of-the-art DNNs, their architectures are mainly configured in manual way, which is a time-consuming task. Thus, it is difficult and frustrating for regular users who do not have comprehensive experience in DNNs to design their optimal architectures to forecast problems of interest. This paper proposes a novel framework to optimize the hyperparameters and architecture of DNNs used for wind speed forecasting. Thus, we introduce a novel enhanced version of the grasshopper optimization algorithm called EGOA to optimize the deep long short-term memory (LSTM) neural network architecture, which optimally evolves four of its key hyperparameters. For designing the enhanced version of GOA, the chaotic theory and levy flight strategies are applied to make an efficient balance between the exploitation and exploration phases of the GOA. Moreover, the mutual information (MI) feature selection algorithm is utilized to select more correlated and effective historical wind speed time series features. The proposed model's performance is comprehensively evaluated on two datasets gathered from the wind stations located in the United States (US) for two forecasting horizons of the next 30-min and 1-h ahead. The experimental results reveal that the proposed model achieves the best forecasting performance compared to seven prominent classical and state-of-the-art forecasting algorithms.

## Keywords

Wind speed forecasting

Deep neuroevolution

Long short-term memory

Enhanced grasshopper optimization algorithm

deal of attention as one of the most environmentally and economically sustainable green energy resources [1]. However, due to wind speed's natural stochastic characteristic, designing an accurate wind energy model in electrical power systems can be considered problematic. Moreover, the wind speed's inconsistency can significantly impact the safety and stability of the micro-grid scheduling and wind turbines control that will further affect the load demand and balance of supply for the wind farm and energy quality [2]. Thus, in energy conversion and management, the optimal and accurate wind speed prediction models' design can bring a stable bias for the generation and transmission of wind energy and diminish the operating costs of the power system.

Over the last few decades, various forecasting techniques have been developed to predict the wind speed time series. Typically, such methods can be categorised within three subgroups, including physical strategies, statistical methods, and artificial intelligence algorithms [3]. Physical

## 1 Introduction

In recent years, with the significant development of huge energy demand and dwindling supplies for renewable energy resources, wind energy has proliferated and gained a great

strategies are the explicit approaches that use meteorological information such as density, temperature, roughness, and atmospheric pressure obstacles [4]. A common technique for numeric weather prediction (NWP) uses mathematical models based on the physical data to forecast wind speed. Nonetheless, this numerical method is not adequate for practical usage as it is not a straightforward process to collect such physical data, particularly for short-term wind speed forecasting.

The statistical methods are the second group used by researchers to forecast wind speed time series. The modeling of different natural phenomena were studied using several data analysis techniques, such as statistical and mathematical modeling containing time series analysis, regression modeling, optimization and numerical analysis [5–9]. For wind speed forecasting, the most well-known statistical methods are auto-regressive models (AR), auto-regressive moving average models (ARMA), and auto-regressive integrated moving average models (ARIMA). Lydia et al. [10] adopted the linear and nonlinear AR models to forecast wind speed from 10-min up to 1-h for a wind energy center in India. Their developed method uses the Gauss–Newton algorithm for parameter tuning of the ARs. They also measured the accuracy of their proposed model using three performance metrics. In another work, Ailliot et al. [11] proposed novel techniques called non-homogeneous Markov-Switching auto-regressive (MS-AR) models to measure wind speed forecasting for an island in France. Different weather types have been analyzed by their method. Torres et al. [12] used the ARMA to forecast the hourly average wind speed from 1-h up to 10-h time horizons ahead. The data for this work have been gathered from a period of 9 years of five locations with different topographic characteristics in Navarre (Spain). They have shown that the ARMA models have a better forecasting performance than the persistence model. Yunus et al. [13] developed an ARIMA model that can cost-effectively capture the probability distribution and time correlation for wind speed data. Their work’s simulation results show that their technique outperforms most of the persistence models to forecast short-term time horizons. As stated in [14], due to the inappropriate pre-assumed linear form, many statistical methods can not cope well against nonlinear wind speed characteristics.

With the rapid growth of feature selection, and machine learning approaches [15–22], numerous artificial intelligence (AI) strategies have been applied for several real-world problems [23–30] and have successfully been designed to address the non-stationary and randomization nature of wind speed time series. Generally, the existing AI-based wind speed forecasting methods can be classified into two categories, including traditional machine learning algorithms and deep learning methods [31]. Support vector machine (SVM) algorithm is one of the prominent categories of the

traditional machine learning algorithms which has a strong generalization potential [32–34]. In recent work, Kong et al. [35] optimized the parameters of a specific type of SVM algorithms called reduced support vector machine (RSVM) using particle swarm optimization (PSO) algorithm for wind speed prediction. In another work, Yu et al. [36] integrated an SVM algorithm with recurrent neural network methods to forecast wind speed with success.

Artificial neural network (ANN) algorithms including backpropagation (BP), Elman neural network (ENN) [37], extreme learning machine (ELM) [38, 39], and radial basis function (RBF) [40] are the most commonly used traditional machine learning algorithms in many areas including the forecasting of wind speed time series. Cadenas and Rivera [41] utilized several BP models to forecast the short-term wind speed of Oaxaca city in Mexico. They showed the structure used for BP has acceptable accuracy for the energy supplier in Oaxaca. In [42], Guo et al. presented a hybrid algorithm based on the BP algorithm and seasonal exponential adjustment (SEA), in which the proposed algorithm was utilized to forecast the daily wind speed 1 year ahead for an area in China from 2001 to 2006. For ENNs, Wang et al. [43] proposed a novel algorithm optimizing these neural networks’ weights and thresholds using a multi-objective whale optimization algorithm for wind speed forecasting. In another work, a multi-objective satin bower-bird optimizer algorithm was employed by [44] to optimize and enhance the forecasting performance of the ENNs based on two real wind farms of China. Salcedo et al. [45] developed a combined wind speed forecasting model using coral reefs optimization algorithm based on a feature selection problem for training ELM [46, 47] networks. In [48], RBF neural networks were trained and optimized by a two-step novel mechanism, including the K-means clustering algorithm and non-dominated sorting genetic algorithm-II (NSGA-II) to maximize the coverage probability of the constructed prediction intervals for a wind speed dataset.

The deep neural network algorithms have gained substantial attention as another successful artificial intelligence category [49–51]. Chen et al. [52] developed a novel nonlinear-hybrid ensemble of deep LSTM models for forecasting wind speed time series. Their hybrid method has been evaluated through two case studies of data from a Chinese wind farm. Liu et al. in [53] proposed a new model for the wind speed multi-step forecasting by deep LSTM networks combined by empirical wavelet transform and ELM [54, 55] algorithms. In another work, Pei et al. [56] proposed a hybrid algorithm including new cell update LSTM combined with empirical wavelet transform for wind speed forecasting simulated on four different datasets. Besides, Khodayar et al. [57] presented a rough deep learning architecture combined by stacked denoising autoencoder (SDAE) and stacked autoencoder (SAE) to forecast wind speed for ultra-short-term

and short-term horizons. Several of these studies and other applications of deep learning have shown that deep learning approaches have more accurate performance than traditional machine learning methods [58, 59]. In general, deep learning has demonstrated tremendous promise as an advanced and efficient machine learning paradigm for the wind speed forecasting field.

In the research work presented in [60], the authors introduced a hybrid method called VMD-DE-ESN, combining variational mode decomposition [61], differential evolution, and echo state for wind speed forecasting. This proposed algorithm showed efficient performance on four stations collected from a wind farm in northwestern Spain. In [62], a new deep learning approach of the gated recurrent unit has been effectively designed to be coupled with the wavelet soft threshold denoising to predict the wind speed series. By adjusting the GRU parameters using a cross-validated grid-search strategy, this deep learning-based hybrid model achieved high adaptability through several case studies. In [63], the authors presented a novel model of day-to-day wind speed forecasting focused on deep CNNs by exploiting the Taguchi's orthogonal array. The experimental findings signify that the proposed efficient design-based CNN outperforms other existing benchmark models.

Among the deep learning approaches [64, 65], LSTM neural network generally has effective and strong performance due to its outstanding ability to cope with long-term time series problems [66, 67]. As a groundbreaking derivative of RNNs (recurrent neural networks), LSTMs can profoundly learn the temporal and long-term dependencies from time-series data and effectively solve the gradient problem compared to traditional RNNs [68]. Thus, these excellent characteristics of LSTM motivated us to consider the deep learning strategy in this work based on LSTM neural network.

Nonetheless, the empirical knowledge for selecting the values of hyperparameters in the LSTM neural network is unknown, and these hyperparameters will affect the forecasting potential of LSTM. Therefore, we introduce a novel deep neuroevolution method based on an enhanced version of grasshopper optimisation algorithm (GOA) to optimize these hyperparameters to increase the wind speed forecasting functionality. GOA is a recent promising optimization algorithm inspired by the swarming behavior of grasshoppers. This algorithm has already been utilized to plenty of stochastic and continuous optimization problems, proving its dominance over the most common meta-heuristics such as differential evolution [69], whale optimizer [70], particle swarm optimization [71], and genetic algorithm [72]. Saxena et al. [73] has introduced an improved version of GOA based on ten forms of chaotic maps in which the performance of these variants is successfully examined on several unimodal and multimodal benchmark functions. In work by Xu et al.

[74], two techniques, namely orthogonal learning and chaotic exploitation, are implemented in the traditional GOA to explore a much more reliable trade-off between both the phases of exploration and exploitation. The analytical findings demonstrate that the modified version can alleviate the shortcomings of GOA and provide more high-quality solutions. An annealing behaved GOA with boosted exploratory and exploitative patterns was proposed by Yu et al. [75] for solving global optimization. For a comprehensive review on GOA, please refer to work presented by Abualigah and Diabat [76]. As mentioned in the previous works discussed above, there are some weaknesses in the standard model of GOA. It can easily collapse into the local optimum and demonstrate a slow convergence rate when faced with several challenging problems.

To further improve the performance quality of GOA, we add two powerful evolutionary operators into the basic GOA for the first time. These operators are based on chaos theory [77] and levy-flight technique, aiming to enhance meta-heuristic evolutionary algorithms' performance for optimization problems. We name this improved version of basic GOA as enhanced GOA (EGOA).

As discussed by Jalali et al. [26], it is of great importance how to select the appropriate hyperparameters for DNN algorithms since their performance depends on the values of such hyperparameters. Due to the decentralized and relational feature representations, deep neural networks can learn nonlinear structures that are deeper and more dynamic than traditional machine learning models such as BP, ENN, ELM, and RBF neural networks, and SVM [78] algorithms. On the other hand, deep LSTM as a prominent deep neural network was successfully deployed to solving different time series real-world problems [53, 79]. The architecture of LSTM neural networks was mostly designed manually, which is a cost-effective and time-consuming procedure [80]. Nonetheless, in the field of wind speed forecasting, there remain little works to utilize the optimal design architecture for LSTM algorithms. In most of the studies that utilized deep learning technologies for wind speed forecasting, the authors designed the architecture of the utilized deep learning manually, which is a time-consuming procedure [81, 82]. Therefore, this paper aims to predict the wind speed with the highest accuracy using a novel optimization algorithm that automatically and efficiently designs the LSTM architecture.

In summary, the principal contributors of this paper are as follows:

1. We introduce an LSTM-based deep neuroevolution time series forecasting algorithm for exploring the implicit knowledge from wind speed time series. Moreover, the mutual information (MI) algorithm is implemented to determine the procedure of input variable selection. The

obtained features by MI aid in selecting the most fitting size of the LSTM input window.

2. While the references such as [67, 83, 84] selects the deep LSTM hyperparameters by the trial-and-error procedure, which is a time-consuming procedure, to efficiently optimize the hyperparameters of the deep LSTM neural network in each layer, an efficient enhanced version of GOA evolutionary algorithm is conducted which we name it as EGOA. This modification enhances the GOA performance centered on chaotic theory and levy-flight strategies to obtain a faster convergence speed and make a more efficient balance between exploitation and exploration phases in the search space.
3. To the best of our knowledge, this work is the first study to utilize an enhanced version of the GOA evolutionary algorithm to optimize the hyperparameters of LSTM neural networks for wind speed forecasting.
4. Our proposed deep hybrid optimization algorithm shows an excellent forecasting performance compared to seven competitive classical and state-of-the-art methodologies for wind speed forecasting.

Two prediction intervals successfully show the proposed model's supremacy: utmost short-term wind speed forecasting for 30-min ahead and short-term wind speed forecasting for 1-h ahead. The datasets used for our experiments are collected from two wind sites near Las Vegas and Denver in the USA. We compare our novel algorithm with several standard and hybrid state-of-the-art time series forecasting algorithms including back propagation (BP) [41], convolutional neural network (CNN) [85], long short-term memory (LSTM) [80], Xgboost [86], empirical mode decomposition and genetic algorithm-BP neural network (EMD-GABP) [87], differential evolution-LSTM (DE-LSTM) [88] and ensemble empirical mode decomposition-GA-particle swarm Optimization Wavelet Neural Network (EGP-WNN) [89] algorithms. The experimental results show that the proposed model is significantly superior to other compared standard models.

The remainder of this study is arranged as follows: Sect. 2 presents the related basic formulation for the proposed method. The experimental procedures for two US collected datasets of two different time-step horizons and discussions of the obtained experimental results are given in Sect. 3. Eventually, all major findings and future works are summarized and presented in Sect. 4.

## 2 Proposed method

This section describes how to develop our enhanced GOA evolutionary algorithm to optimize the structure of the LSTM neural networks by providing details.

### 2.1 Structure of basic GOA

Saremi et al. [72] recently proposed the swarm-based GOA based on imitating the behavior of grasshopper groups in the environment to realize optimal or sub-optimal solutions to the complex multimodal or composite hybrid problems. After initialization, the updating rule follows three laws: social interaction, gravity force, and wind advection. The current position of  $i$ th agent is referred to  $X_i$  and described by

$$X_i = S_i + G_i + A_i, \quad (1)$$

where  $S_i$  denotes the variable for social interaction,  $G_i$  represents the gravity force and  $A_i$  denotes to the wind advection. Social interaction is the most influential component, based on its impact on the motion patterns, which can be determined as follows:

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^N s(d_{ij}) \hat{d}_{ij} \quad (2)$$

$$d_{ij} = |x_j - x_i| \quad (3)$$

$$\hat{d}_{ij} = (x_j - x_i) / d_{ij} \quad (4)$$

$$s(r) = fe^{-r/l} - e^{-r}, \quad (5)$$

where  $d_{ij}$  represents the distance between the agent  $i$  to the  $j$ th agent, and  $\hat{d}_{ij}$  denotes to a unit vector between  $i$ th and  $j$ th agent. The function  $s$  determines the social forces, which can be evolved based on the  $f$  and  $l$  parameters. The distance between agents should be allocated between the [1,4] interval. The gravity force of an agent can be expressed as follows:

$$G_i = -g\hat{e}_g, \quad (6)$$

where  $g$  is the constant of gravity and  $\hat{e}_g$  is the vector of unity towards the center of the earth. Grasshopper wind advection can be computed as following:

$$A_i = u\hat{e}_w, \quad (7)$$

where  $u$  denotes to a constant drift and  $\hat{e}_w$  is a vector of unity in wind direction. Finally, Eq. (1) can then be generalized as follows:

$$X_i = \sum_{\substack{j=1 \\ j \neq i}}^N s\left(|x_j - x_i|\right) \frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + u\hat{e}_w \quad (8)$$

where the number of agents are denoted by  $N$ . The considerable influence of gravity force on the grasshopper is too slow and weak to be simply ignored and implicitly assumes that the direction of wind ( $A$  component) is always in the best solution  $\hat{T}_d$ . The logical model between the agents is also demonstrated in Fig. 1.

After all, the mathematical formula is developed as follows:

$$X_i^d = c \left( \sum_{\substack{j=1 \\ j \neq i}}^N c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j - x_i}{d_{ij}} \right) + \hat{T}_d, \quad (9)$$

where  $ub_d$  is the  $d$ th dimension of upper boundary,  $lb_d$  is the  $d$ th dimension of lower boundary,  $\hat{T}_d$  is the  $d$ th dimension value in the best solution so far obtained, and the parameter  $c$  is continuously updated to minimize exploration phase and help increasing exploitation phase according to the number of iterations through the following equation:

$$c = c_{\max} - l \frac{c_{\max} - c_{\min}}{L}, \quad (10)$$

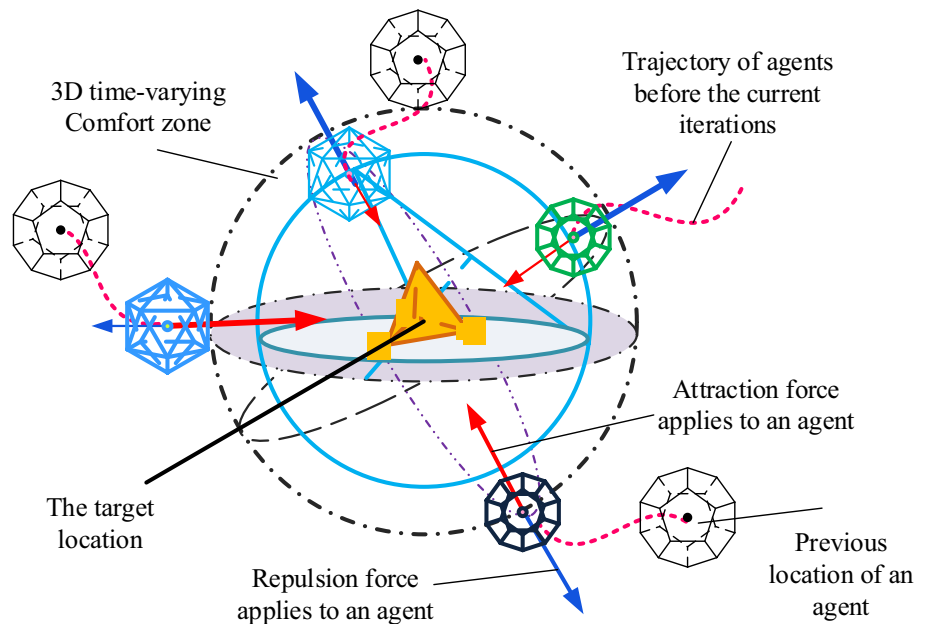
where the maximum value is represented by  $c_{\max}$ , the minimum value is denoted by  $c_{\min}$ ,  $l$  corresponds to the current iteration, and  $L$  denotes the maximum iteration number.

## 2.2 Chaotic-population initialization

Boosting the balance of the swarm-based methods such as GOA is an essential part of the optimization process. For

example, advanced variants of several other evolutionary and swarm intelligence methods such as boosted moth-flame optimizer (LGCMFO) [90], chaotic, random spare ant colony optimization [91], biogeography-based whale optimizer [92], double adaptive moth-flame optimizer [93], orthogonal learning grey wolf optimizer [94], Gaussian bare-bones fruit fly optimizer [95] have found their applications in both basic and advanced versions in many areas based on stabilizing the balance of the exploration and exploitation of the core processes. In this regard, the quality of the initial population can significantly impact the convergence speed, and solution accuracy with evolutionary algorithms that continuously desire optimization via population iteration [96, 97]. The basic GOA typically initializes the population randomly, making it hard to guarantee population diversity, leading to weak search results and performance. Therefore, it is essential to enhance the diversity of the initial population. Generally speaking, chaos is a pseudo-random movement formed by a stochastic deterministic mechanism that is initially sensitive to a value and then generates many pseudo-random patterns [98, 99]. It has the attributes of non-linearity, randomness, and consistency. These characteristics can easily eliminate the algorithm from the local optimal solution when solving function optimization problems to preserve population diversity and increase the global search efficiency [100, 101]. Among various chaotic maps having different function optimization abilities, the tent map has shown its greater performance than the other maps [102]. Therefore, we used tent map agent population initialization, which can be formulated as

**Fig. 1** Primitive patterns between the agents in an update of GOA



$$x_{i+1} = \begin{cases} 2 \times x_i, & 0 \leq x_i \leq 1/2; \\ 2 \times (1 - x_i), & 1/2 \leq x_i \leq 1. \end{cases} \quad (11)$$

Assume  $D$  represents the search space dimension and  $N$  denotes to the population size, the tent map sequence  $x_{ij}(i = 1, 2, \dots, N; j = 1, 2, \dots, D)$  is generated by Eq. (11). Based on Eq. (12), the initialized population  $P_0 = \{X_{ij}\}$  is mapped into the search space as follows:

$$X_{ij} = x_{ij} \times (X_{\max j} - X_{\min j}) + X_{\min j}, \quad (12)$$

where the maximum and minimum of the  $j$ th dimension are represented by  $X_{\max j}$  and  $X_{\min j}$ , respectively.

### 2.3 Levy flight

Levy-flight (LF) was initially proposed in 1937 by Paul Levy, a French mathematician. In terms of levy statistics, many artificial and natural phenomena have been defined [103]. The LF is a well-respected subclass of non-Gaussian stochastic walks to distribute their step-length values concerning a stable Levy distribution. The levy distribution is accomplished as follows:

$$\text{Levy}(\beta) \sim u = t^{-1-\beta}, \quad (13)$$

where  $\beta$  provides a significant levy index for stability adjustment. The levy random number is determined using the given equation:

$$\text{Levy}(\beta) \sim \frac{\varphi \times \mu}{|\nu|^{1/\beta}}, \quad (14)$$

where  $\mu$  and  $\nu$  represent the standard normal distributions,  $\Gamma$  denotes to a standard Gamma function, the value of  $\beta$  parameter is equal to 1.5, and  $\varphi$  is computed as follows:

$$\varphi = \left[ \frac{\Gamma(1 + \beta) \times \sin(\pi \times \beta/2)}{\Gamma\left(\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}\right)} \right]^{1/\beta}. \quad (15)$$

For achieving a potential trade-off between the capability of evolutionary algorithms to exploration and exploitation, LF approach is employed to update the position of each agent, which is calculated as follows:

$$X_i^{\text{levy}} = X_i + r \oplus \text{levy}(\beta), \quad (16)$$

where  $X_i^{\text{levy}}$  represents the new position of the  $i$ th agent  $X_i$ ,  $r$  denotes to a random vector in  $[0,1]$  interval, and  $\oplus$  is the dot product (entry-wise multiplications).

### 2.4 Enhanced GOA

This section outlines the proposed enhanced GOA (EGOA) in detail. In EGOA, first, we adopt the chaos theory to boost the quality of the initial population position as described in detail in Sect. 2.2. Then we utilize the Levy flight strategy into the GOA to address the original GOA's drawback to make a more appropriate balance between exploration and exploitation phases. Section 2.3 defined the fundamental principles of the levy flight strategies in detail. As it is well-known regarding evolutionary algorithms, search agents' diversity is crucially important since diversity enables the population to search functionality towards the global optimum. The levy flight component was utilized in GOA to improve GOA population diversity. To this end, once the position of  $i$ th search agent  $X_i$  is updated, the levy flight component is incorporated to deploy a new candidate solution. The modified mathematical equation for the enhanced GOA is defined as follows:

$$X_i^{\text{levy}} = X_i^* + \text{rand}(d) \oplus \text{levy}(\beta) \quad (17)$$

$$X_i^{t+1} = \begin{cases} X_i^{\text{levy}} & \text{fitness}(X_i^{\text{levy}}) > \text{fitness}(X_i^*) \\ X_i^* & \text{otherwise,} \end{cases} \quad (18)$$

where  $X_i^*$  represents the current agent position after the new update, and  $\text{rand}(d)$  is a random  $d$ -dimensional vector into the interval of  $[0,1]$ . Since levy flight is a randomized procedure where the jump's size typically follows the levy probability distribution function, the new candidate solution obtained via the levy flight algorithm has a significant chance of jumping from the local optimum and achieving a superior solution. Search agents with more excellent fitness are preserved in the population to guarantee the reliability of the population. Therefore, the levy flight mechanism can cause competitive agents to move faster towards the global optimum. As a result, since incorporating the chaotic theory and levy flight strategies help to enhance the capabilities of GOA, we name this novel proposed method as enhanced GOA (EGOA).

### 2.5 LSTM

LSTM neural network is a deep learning algorithm with time-varying inputs and targets. It also has an excellent performance in time-series data processing thanks to its outstanding ability to solve long-term dependency problems. The cornerstone of the LSTM neural network is the memory cell, which can preserve the temporal state. The input gate can add or remove the information to the cell state with the memory cell, forget gate, and the output gate. Figure 2 describes a sample unit of a LSTM network. The

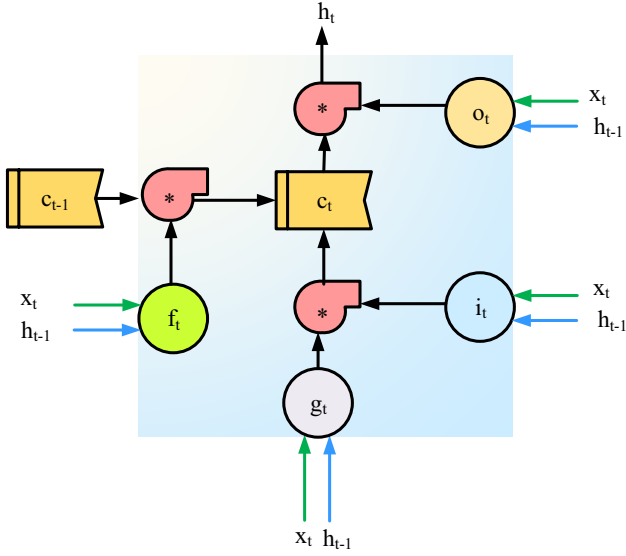


Fig. 2 Structure of the deep LSTM neural network block

key stages of this neural network are explained as follows in three stages:

1. The input gate monitors the input activation when the input gate is activated, and the new input information is received to the memory cell.
2. The forget gate forgets the unimportant contents. Thus, the past cell status is forgotten in the pipeline when the forget gate is enabled.
3. The output gate regulates the output activation. Thus, the current cell output is propagated to the final state when the output gate is enabled.

The three gates are sigmoid units that adjust each item in the interval of  $[0, 1]$ . The standard sigmoid logistics function is specified as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (19)$$

The  $i$ th entry gate regulates the input information that passes into the memory cell, resulting in the following:

$$i_t = \sigma(w_{xi}x_t + w_{hi}h_{t-1} + b_i). \quad (20)$$

Forget gate  $f_t$  regulates forgetting cell information, in which

$$f_t = \sigma(w_{xf}x_t + w_{hf}h_{t-1} + b_f). \quad (21)$$

Output gate  $o_t$  regulates the output information that flows from the cell, deriving from the following equation:

$$o_t = \sigma(w_{xo}x_t + w_{ho}h_{t-1} + b_o). \quad (22)$$

For the time  $t$ , a  $\tanh$  function quantifies the input characteristics by inputting  $x_t$  and the previous hidden state  $h_{t-1}$  as follows:

$$g_t = \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c). \quad (23)$$

Here, the memory cell is updated through regulated input features and the partial forgetting of previous memory cell, which provides

$$c_t = f_t * c_{t-1} + i_t * g_t. \quad (24)$$

The hidden output status  $h_t$  is eventually determined by the output gate  $o_t$  and the memory  $c_t$ , where

$$h_t = o_t * \tanh(c_t). \quad (25)$$

Therefore, the LSTM output  $y_t$  is determined as follows:

$$y_t = \sigma(w_{hy}h_t + b_y). \quad (26)$$

In Eqs. (20)–(26), the  $w_{xi}$ ,  $w_{xf}$ ,  $w_{xo}$ , and  $w_{xc}$  are the proper input weights.  $w_{hi}$ ,  $w_{hf}$ ,  $w_{ho}$ , and  $w_{hc}$  matrices represent the recurrent weight matrices, and  $w_{hy}$  denotes to the matrix of hidden output weight. The corresponding bias vectors are represented by  $b_i$ ,  $b_f$ ,  $b_o$ ,  $b_c$ , and  $b_y$ .

## 2.6 Proposed EGOA-LSTM Method

This section presents the proposed wind speed forecasting method called EGOA-LSTM. This method aims to utilize the improved GOA algorithm to optimize the LSTM neural network's hyperparameters, leading to improving the wind speed forecasting model's accuracy. Before applying EGOA, two issues should be considered, including representation of solutions and calculation of fitness function. It should be noted that four different hyperparameters, including batch size, learning rate, maximum epoch, and neural units, are considered in the proposed method to be optimized by the EGOA algorithm. Therefore, each solution in EGOA can be represented as a vector with four dimensions, each of which corresponds to one of the four hyperparameters. Learning rate is a hyperparameter with continuous values, which EGOA can obtain its optimal value. In contrast, batch size, maximum epoch, and neural units are other hyperparameters with discrete values. As EGOA explores solution space in continuous mode, we need to convert these hyperparameters' optimal values to their corresponding discrete values. To this end, each real value can be converted to an integer value using the following equation:

$$y_{ij} = \left\lfloor b_j \times \frac{x_{ij} - lb}{ub - lb} + 0.5 \right\rfloor, j = 1, \dots, n, \quad (27)$$

where  $b_j$  is the total number of the item of type  $j$ ,  $x_{ij}$  is the real number corresponds to the  $j$ th dimension of the solution  $X_i$ ,  $y_{ij}$  is the converted integer value,  $lb$  and  $ub$  are respectively the lower and upper bounds of the search space.

In the proposed EGOA-LSTM method, first of all, the initial population with  $n$  solutions is randomly initialized using Eq. (8). Each solution is denoted by a four-dimensional vector  $X_{ij}$ ,  $i = 1, \dots, n$  and  $j = 1, \dots, 4$  where each dimension  $j$  corresponds to one of the four LSTM hyperparameters. After the initialization of the initial population, new solutions can be obtained by repeatedly updating the solutions' current positions using Eq. (9). Moreover, the levy flight strategy is applied to the updated positions to balance exploration and exploitation using Eqs. (17) and (18). The procedure repeats until the termination criterion is reached, and then the best-obtained solution is considered as the final result. This obtained solution can be used as the optimal values of the LSTM hyperparameters. To evaluate the usefulness of each solution, we need to define a fitness function. To this end, the input time series data is divided into two sets, including training and test. The training set is used to optimize the LSTM hyperparameters using EGOA, while the test set is used to evaluate the final obtained wind speed forecasting model's performance. Suppose that  $\vec{y}$  is a vector to denote the historical wind speed time series data for  $M$  time steps expressed as follows:

$$\vec{y} = (y_{(0)}, y_{(1)}, \dots, y_{(M-1)}), \quad (28)$$

where  $y_{(t)}$  denotes the actual wind speed value for the time step  $t$ . The purpose of the proposed wind speed forecasting model is to predict the wind speed values of the next  $N$  time steps using LSTM neural network which these predicted values can be represented as follows:

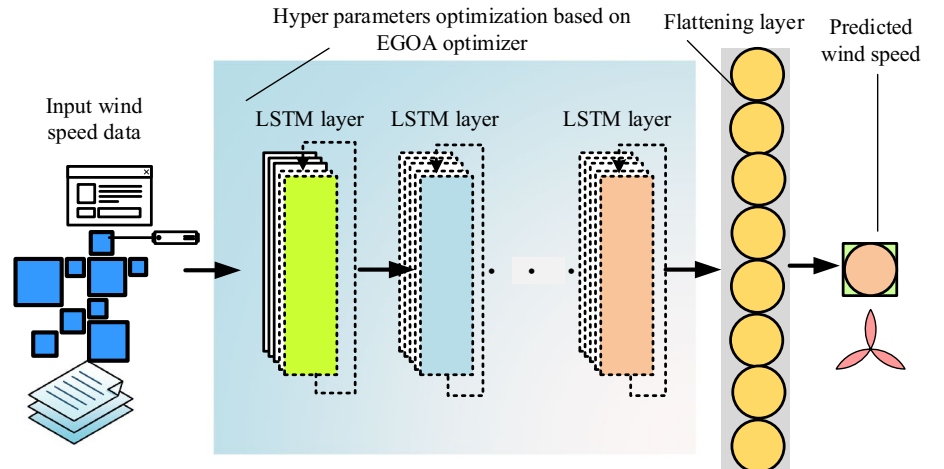
$$\vec{\hat{y}} = (\hat{y}_{(M)}, \hat{y}_{(M+1)}, \dots, \hat{y}_{(M+N-1)}), \quad (29)$$

where  $\hat{y}_{(t)}$  denotes the predicted wind speed value for the time step  $t$ . It should be noted that each solution in EGOA is used to configure an LSTM model based on the hyperparameters' obtained values. Therefore, the configured LSTM model's performance on forecasting wind speed values can be considered as the fitness function. To this end, the input vectors of the LSTM model are represented using Eq. (28) based on the training data. The LSTM model is then utilized to predict the wind speed values of the next  $N$  time steps, which are represented using Eq. (29). To calculate the fitness value of each solution in EGOA, the mean square error can be used as follows:

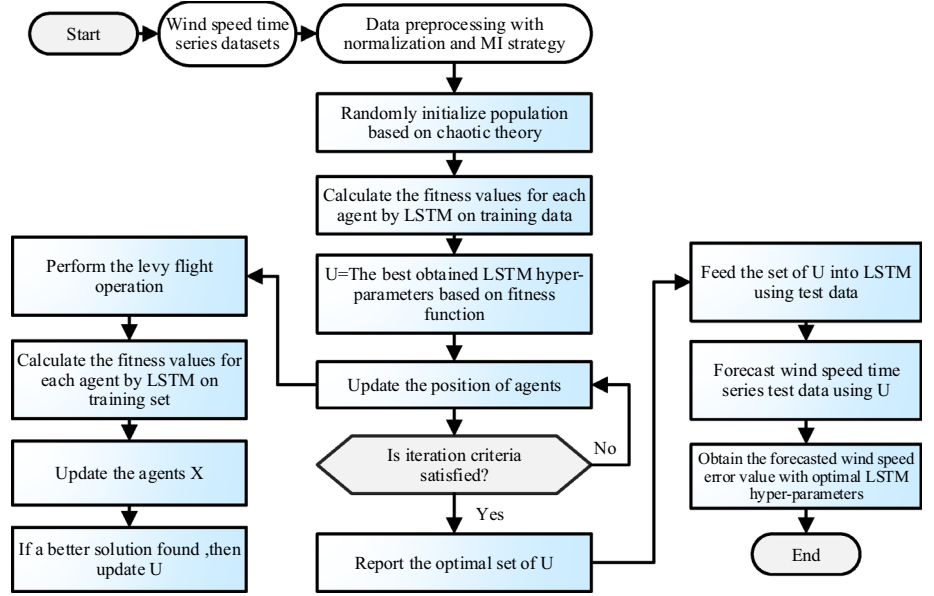
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (30)$$

where  $y_i$  is the actual wind speed value and  $\hat{y}_i$  is the predicted wind speed value obtained by the LSTM neural network. Obviously, a solution with a lower MSE value has a higher fitness value and vice versa. Therefore, the proposed method aims to obtain a solution with the lowest MSE value (i.e., highest fitness value) containing the optimal values of LSTM hyperparameters. This leads to obtaining an LSTM model with maximum performance forecasting wind speed values in the test set. After determining the optimal values of LSTM hyperparameters using EGOA, the configured LSTM model is used to predict wind speed values in the test set. Algorithm 1 represents the overall steps of the proposed EGOA-LSTM method. In Fig. 3, the deep proposed model's whole procedure is illustrated. Also, the flowchart of the proposed wind speed forecasting model is depicted in Fig. 4.

**Fig. 3** The schema of deep EGOA-LSTM model for wind speed forecasting



**Fig. 4** Flowchart of the proposed EGOA-LSTM model for wind speed forecasting




---

**Algorithm 1** Pseudo-code of the proposed wind speed forecasting method (EGOA-LSTM)

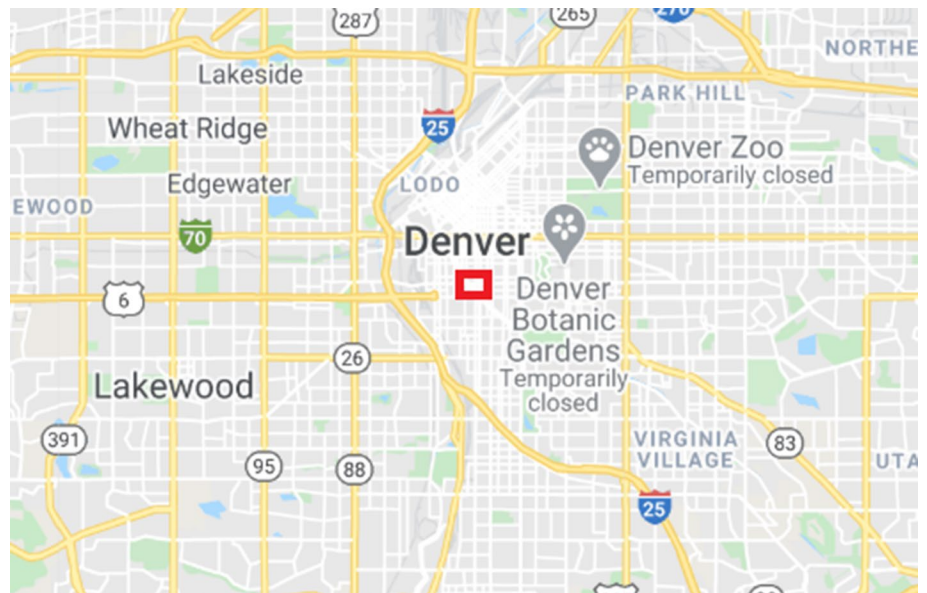
---

- 1: **Input:**  $pop\_size$  (population size),  $c_{max}$ ,  $c_{min}$  and  $L$  (maximum number of iterations).
  - 2: **Output:** Predicted wind speed values.
  - 3: **Begin algorithm:**
  - 4: Split dataset into training set  $Tr$  and test set  $Te$ ;
  - 5: Initialize the agent population  $X_i$  ( $i=1,2,\dots, pop\_size$ ) based on chaotic theory;
  - 6: **for** (each search agent  $X_i$ ) **do**
  - 7:   Set an LSTM model based on the values of hyperparameters obtained by the solution  $X_i$  ;
  - 8:   Calculate the fitness of solution  $X_i$  using Eq. (30) as the MSE of LSTM model obtained based on the training set  $Tr$ ;
  - 9: **end for**
  - 10: Set  $B$ = the best search agent based on the calculated fitness values;
  - 11: Set  $l=1$ ;
  - 12: **while** ( $l < L$ ) **do**
  - 13:   Update  $c$  according to Eq. (10);
  - 14:   **for** each search agent  $X_i$  **do**
  - 15:     Normalize the distance between agents in  $[1,4]$  interval;
  - 16:     Update the current position of  $X_i$  based on Eq. (9);
  - 17:     Apply the levy flight strategy by using Eqs. (15) and (16);
  - 18:     Check the  $X_i$  values to be in the boundaries and bring them back if they go outside;
  - 19:     Set an LSTM model based on the values of hyperparameters obtained by the solution  $X_i$ ;
  - 20:     Calculate the fitness of solution  $X_i$  using Eq. (30) as the MSE of LSTM model obtained based on the training set  $Tr$ ;
  - 21:     **if** the fitness of solution  $X_i$  is better than the fitness of  $B$  **then**
  - 22:       Set  $B=X_i$ ;
  - 23:     **end if**
  - 24:   **end for**
  - 25:   Set  $l=l+1$ ;
  - 26: **end while**
  - 27: Set an LSTM model based on the hyperparameters obtained by the best solution  $B$ ;
  - 28: Predict the wind speed values in the test set  $Te$  using the best obtained LSTM model;
  - 29: Return the predicted wind speed values as the output;
  - 30: **End algorithm**
-

**Fig. 5** Location of wind speed site for Las Vegas case study



**Fig. 6** Location of wind speed site for Denver case study



### 3 Experimental results

#### 3.1 Data

In contrast to several studies such as [53, 56, 82] which used a small amount of wind speed data (usually less than one-thousand samples) for showing the efficiency of their proposed deep learning algorithms, the 30-min interval between consecutive historical samples of two wind stations in the US for the whole year 2012 has been used in this study. Western Wind Dataset [104] created by the National Renewable Energy Laboratory (NREL) and 3TIER, the wind speed time series estimated for two wind sites in Las Vegas and

Denver are used to examine the efficiency of the proposed EGOA-LSTM algorithm. The location of these two wind sites are shown in Figs. 5 and 6. In total, there are 17520 wind speed values measured in intervals of 30 min for each of two wind stations. Thus, sufficient data are available to train and test our proposed deep learning method. Similar to [80], 70% of each dataset is considered for training sets while 10% is used for validation sets and the rest is dedicated to testing sets. At the beginning of the experiments, raw datasets were pre-processed into the interval of [0,1] using Eq. (31) to improve the forecasting efficiency. The final goal is to predict the next forecasting horizons for the next 30 min (one-step) and 1-h (two-step) ahead.

$$z = \frac{z - z_{\min}}{z_{\max} - z_{\min}}. \quad (31)$$

### 3.2 Evaluation metrics

Four loss functions are employed to assess the prediction performance of the proposed model as the criterion related to the wind speed values including root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and R squared ( $R^2$ ). The lower the loss function value, the higher the model accuracy for wind speed forecasting. The formulas of the performance evaluation metrics are as follows:

$$\text{RMSE} = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (y'_i - y_i)^2} \quad (32)$$

$$\text{MAE} = \left(\frac{1}{n}\right) \sum_{i=1}^n |y'_i - y_i| \quad (33)$$

$$\text{MAPE} = \left(\frac{1}{n}\right) \sum_{i=1}^n \left| \frac{y'_i - y_i}{y_i} \right| \quad (34)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y'_i - y_i)^2}{\sum_{i=1}^n (y'_i - \tilde{y}_i)^2}, \quad (35)$$

where  $y'_i$  represents the predicted wind speed value of corresponding  $y_i$  and  $n$  indicates the number of data points in the test set.

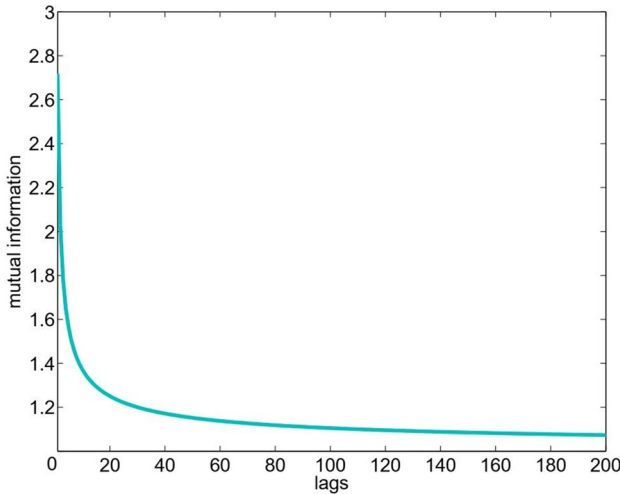


Fig. 7 Mutual information of various time-lags for Las Vegas dataset

### 3.3 Input feature selection

Input feature selection [105–107] is a fundamental, and yet crucial consideration in determining the optimal structure of data-driven models. In the literature, several studies such as [108] have operated auto-correlation function (ACF) to achieve the cross-correlation of wind speed time series at various time instances. As ACF can only calculate linear dependency of variables with themselves, and the wind speed information is highly nonlinear in nature, mutual information (MI) is an effective strategy to estimate the data's nonlinear and linear correlations. Assume  $X$  and  $Y$  are considered as two random variables. The entropy of  $X$  represented by  $H(X)$  is a metric of its uncertainty, and the joint entropy of  $X$  and  $Y$  are denoted by  $H(X, Y)$ . The conditional entropy calculated by  $H(Y|X) = H(X, Y) - H(X)$  indicates the uncertainty of  $Y$  due to the observation of the variable  $X$ . The MI is a nonlinear equation between two random variables to calculate the amount of information acquired about a variable if the other variable is observed. MI is determined by  $I(X, Y) = H(Y) - H(Y|X)$  which reduces the uncertainty of variable  $Y$  due to the observation of variable  $X$ , and vice versa.

Suppose  $v(t)$  as the value of wind speed at time  $t$ , the MI between  $v(t - l + 1)$  and  $v(t + 1)$  is calculated considering  $l$  as the time lag. Following the selection of the most relevant inputs for our deep EGOA-LSTM algorithm, the wind speed values equivalent to time-lags with MI more than  $x = 0.4$  are considered for input sets to highlight the correlation in two wind datasets for 30-min and 1-h ahead forecasting horizons. In Fig. 7, MI for the lag  $l = 1$  to  $l = 200$  of the Las Vegas dataset for 30-min ahead interval is illustrated. As it is indicated from this figure, the correlation among the wind speed observations is increased by the time-lag. As a result, time-lags from  $l = 1$  to  $l = 29$  are incorporated. Assume the current time is  $t$  and we are going to predict the wind speed values for a future time horizon. Then, our input set is a  $29+28 = 57$  dimensional vector  $v(t - 28), \Delta v(t - 27), v(t - 26), \dots, v(t)$  with the sequential difference  $\Delta v(t) = v(t) - v(t - 1)$  of the wind speed data.

Table 1 The hyperparameters of the deep LSTM network during the evolution

Hyperparameter	Range
$M_e$	[1–500]
$N_u$	[1–60]
$B_s$	[1–200]
$L_r$	[0.0001–0.1]

**Table 2** Error estimated results of the predictions of the 30-min ahead wind speed time series for Las Vegas dataset. The bold values represent the best performance evaluation metric

Algorithm	RMSE	MAE	$R^2$	MAPE
XGBOOST	0.158511	0.147968	0.467719	418.5747
BP	0.050608	0.033037	0.900678	58.43221
CNN	0.047746	0.031668	0.911593	55.59189
LSTM	0.046534	0.032627	0.916025	65.59595
DE-LSTM	0.045165	0.030470	0.920891	51.78664
EMD-GABP	0.043582	0.028625	0.926339	47.20511
EGP-WNN	0.037143	0.025895	0.946497	51.28683
EGOA-LSTM	<b>0.033647</b>	<b>0.019135</b>	<b>0.956096</b>	<b>24.42821</b>

**Table 3** Error estimated results of the predictions of the 1-h ahead wind speed time series for Las Vegas dataset. The bold values represent the best performance evaluation metric

Algorithm	RMSE	MAE	$R^2$	MAPE
XGBOOST	0.176611	0.164245	0.206522	472.6453
BP	0.106510	0.076001	0.561186	161.1773
CNN	0.102338	0.073158	0.594892	153.8014
LSTM	0.095044	0.068763	0.650583	127.7113
DE-LSTM	0.092181	0.065956	0.671315	125.8747
EMD-GABP	0.088070	0.062095	0.699977	113.9565
EGP-WNN	0.065221	0.041562	0.835458	81.66707
EGOA-LSTM	<b>0.064619</b>	<b>0.038741</b>	<b>0.838482</b>	<b>66.02486</b>

### 3.4 Parameter settings

In this section, we describe the default configurations for performing our proposed EGOA-LSTM algorithm. Regarding the initialized parameters for EGOA, we set the number of population = 30, the maximum number of iterations = 20, and the number of runs = 20 for each dataset. Two main parameters for GOA are  $C_{\max}$  and  $C_{\min}$ , which their values are set to 1 and 0.00004, respectively. These values are selected based on the recommended literature [72]. There are four key hyperparameters for training the deep LSTM neural network, including maximum epoch ( $M_e$ ), neural units in the hidden layer ( $N_u$ ), batch size ( $B_s$ ), and learning rate ( $L_r$ ), which are fed to EGOA. The range of these hyperparameters is shown in Table 1. Previous works [109–111] have used a more limited range of hyperparameters. However, in this study, we have chosen a wider range of hyperparameters to train LSTM. Moreover, the number of layers that have been used for designing the LSTM architecture is denoted to three. To further assess our proposed approach’s predictive ability, the proposed deep neuroevolution model is compared with the recently proposed deep learning models. The single and hybrid algorithms presented herein are used as compared models to highlight the efficiency

of the EGOA-LSTM. These models are backpropagation (BP) [41], convolutional neural network (CNN) [85], long short-term memory (LSTM) [80], Xgboost [86], empirical mode decomposition and genetic algorithm-BP neural network (EMD-GABP) [87], differential evolution-LSTM (DE-LSTM) [88] and ensemble empirical mode decomposition-GA-particle swarm Optimization Wavelet Neural Network (EGP-WNN) [89] algorithms. The configuration for these compared single and hybrid approaches are based on their recommended literature. The proposed EGOA-LSTM model is implemented in the Python programming language [112] version 3.7, TensorFlow 1.15, CUDA 10.1, cuDNN 8.0.5 and executed on an NVIDIA GTX 1080 Ti GPU, RAM of 32 GB, and Intel Core i7 machine with 3.7 GHz 12 cores CPU.

### 3.5 Analysis of the results and discussion

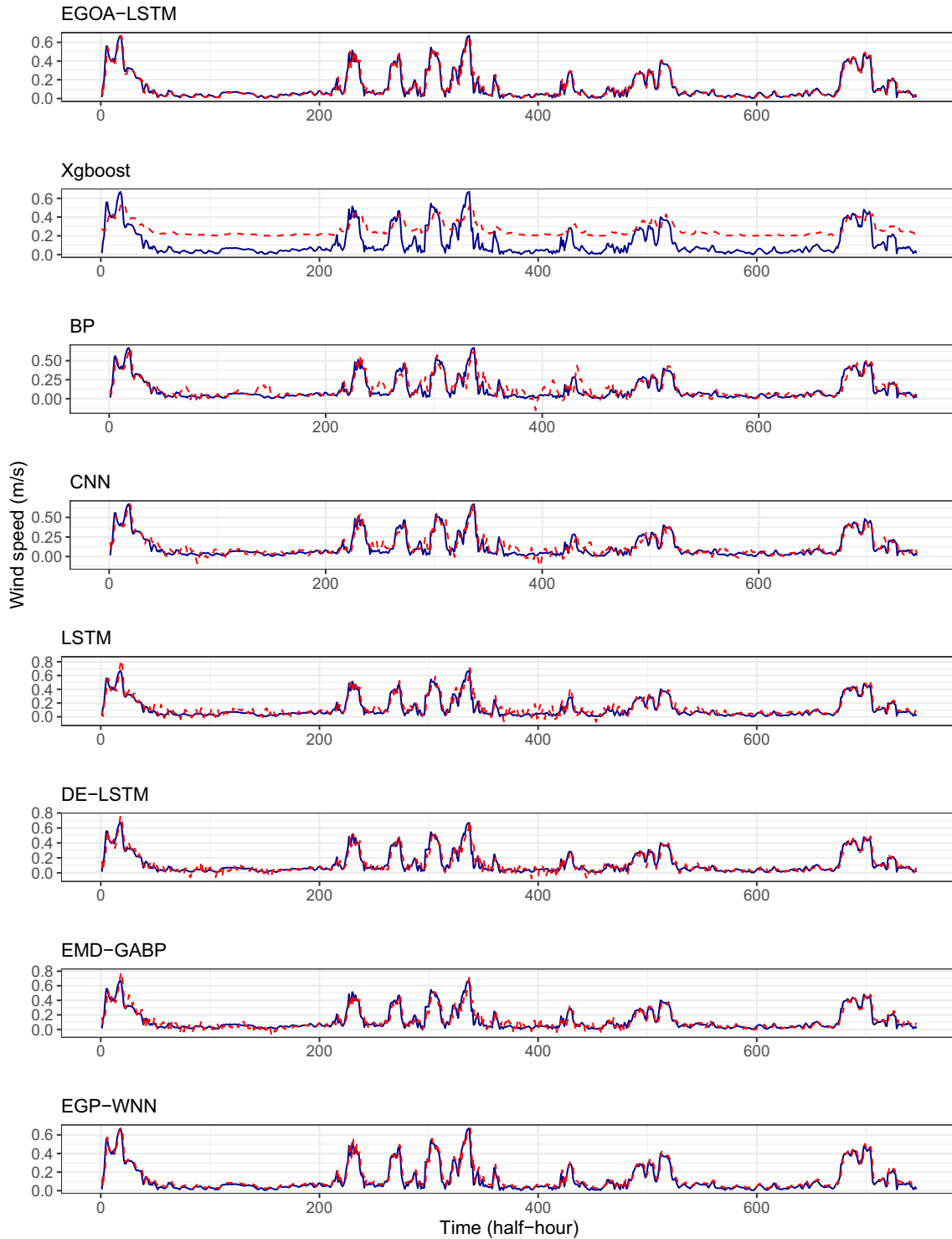
In this section, we report the results of experiments for two case studies with two forecasting horizons. We then discuss these results in detail.

#### 3.5.1 Las Vegas case study

In this case study, the wind speed data recorded for every 30 min was utilized as the dataset. We consider this wind speed dataset for forecasting of utmost short-term 30 min (one-step ahead) ahead and short-term 1 h (two-step ahead) ahead.

Tables 2 and 3 report the forecasting performance of the different prediction algorithms for the 30-min ahead and 1-h ahead wind speed data, respectively. Moreover, Fig. 8 demonstrates the actual and predicted values of different forecasting algorithms for the next 30-min ahead of the Las Vegas dataset. The blue and red colors seen in these figures represent the actual and predicted data values of the algorithms used in this paper, respectively. The convergence curve for two different horizons of the Las Vegas dataset is also demonstrated in Fig. 9. Also, the violin plots of four hyperparameters involved in optimizing LSTMs using our novel deep neuroevolution method are illustrated in Figs. 10 and 11.

From Table 2 and Fig. 8, it is noteworthy that the proposed EGOA-LSTM carries out better than the compared forecasting techniques with the minimum value of RMSE as 0.033647, MAE as 0.019135, MAPE as 24.42821 and the maximum value of  $R^2$  as 0.956096 in terms of next 30-min wind speed prediction. On the other hand, the best algorithm among compared predictive models is EGP-WNN with RMSE as 0.037143, MAE as 0.025895, MAPE as 51.28683 and  $R^2$  as 0.946497 whereas Xgboost is the worst one with RMSE as 0.158511, MAE as 0.147968, MAPE as 418.574722 and  $R^2$  as 0.467719. It appears from Fig. 8 that the EGOA-LSTM demonstrates better curve fitting of the



**Fig. 8** The wind speed forecasting results of 30-min ahead obtained by different algorithms on Las Vegas case study

actual wind speed time series compared to other forecasting models.

Table 3 shows that the EGOA-LSTM achieves better performance than the compared forecasting algorithms in terms of next 1-h ahead wind speed forecasting, including the

minimum value of RMSE as 0.064619, MAE as 0.038741, MAPE as 66.02486, and the maximum value of  $R^2$  as 0.838482. Among the compared models, EGP-WNN is the leading algorithm with minimum values in terms of RMSE, MAE, MAPE, and maximum value for  $R^2$ . The convergence

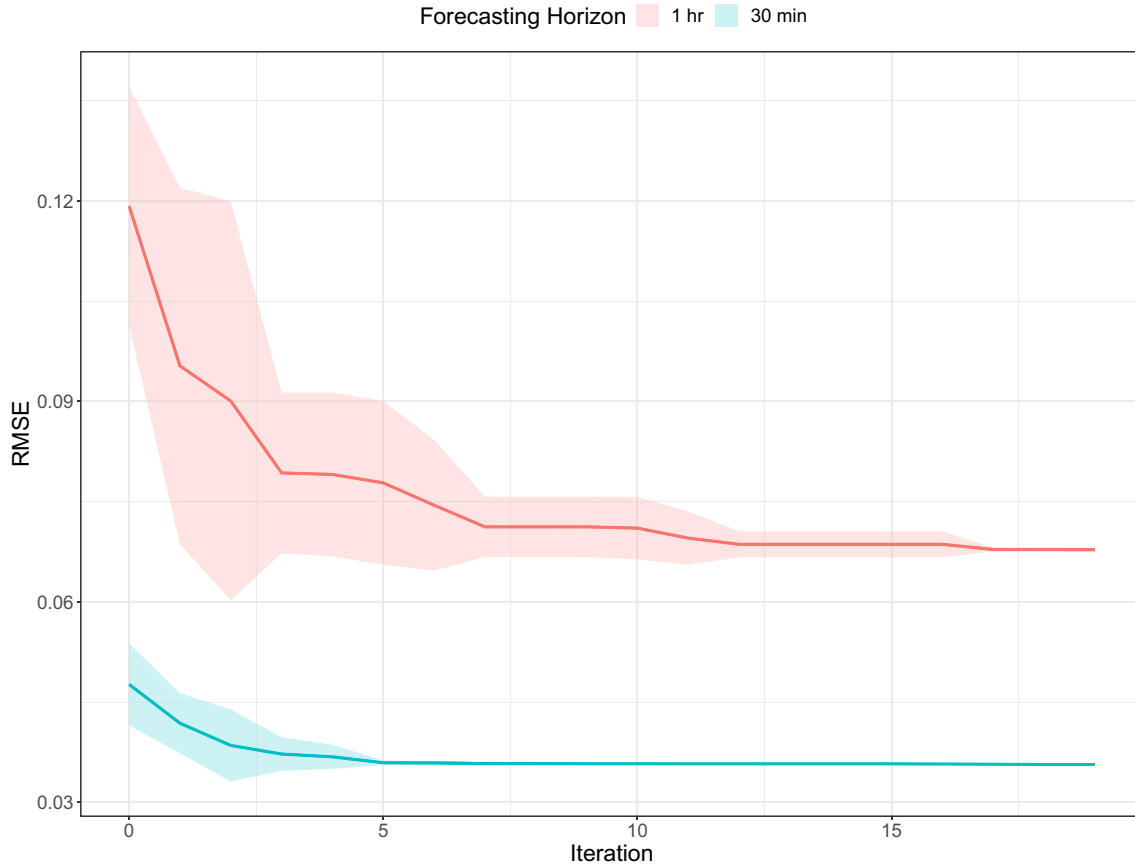


Fig. 9 The convergence profile of EGOA-LSTM algorithm for two forecasting horizons of Las Vegas case study

profile of the proposed EGOA-LSTM algorithm for the Las Vegas dataset using two different forecasting horizons is shown in Fig. 9. As we can see in this figure, the prediction error for 1-h ahead of forecasting is much higher than 30-min ahead of forecasting. Moreover, our proposed method converges properly to the end of the maximum iteration number for both forecasting horizons.

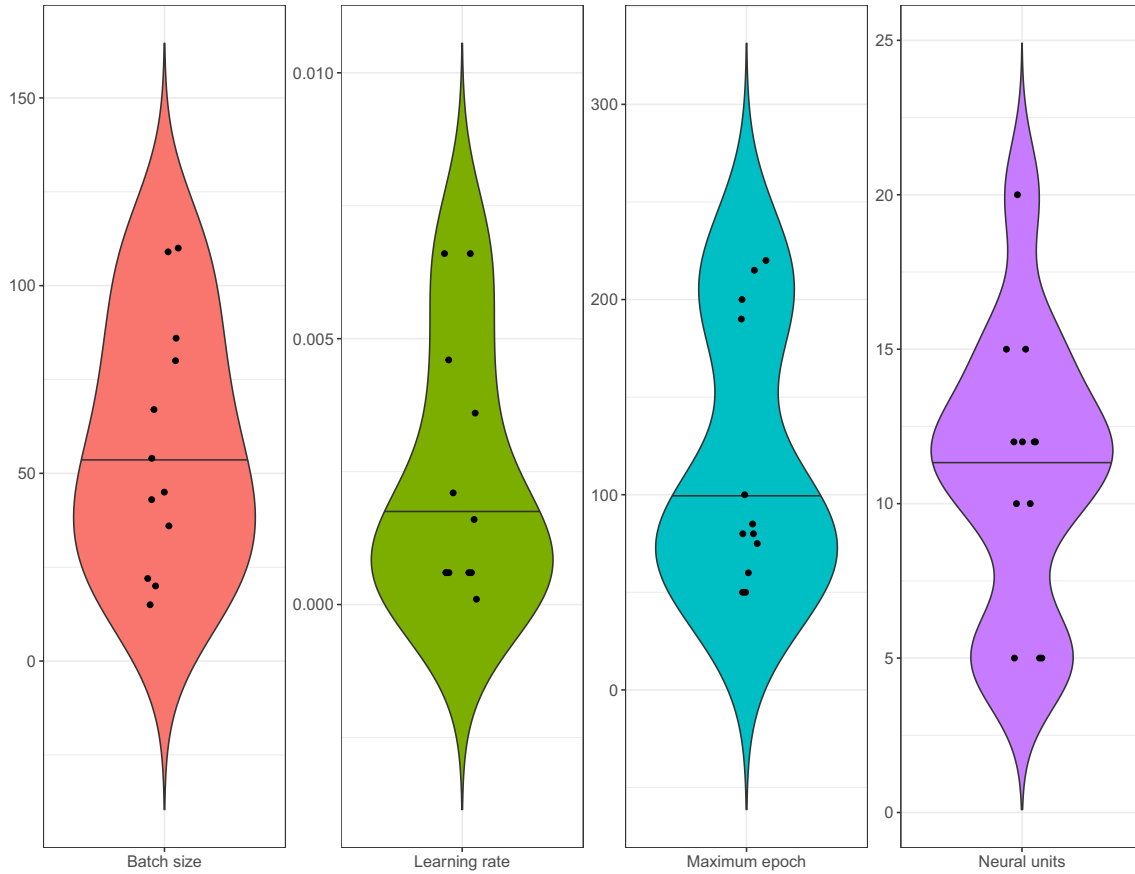
The violin plots using four different hyperparameters evolved into EGOA-LSTM algorithm for 30-min and 1-h ahead wind speed forecasting are illustrated in Figs. 10 and 11. In an overview of these two figures, we reveal that the EGOA-LSTM assigns values to deep LSTM hyperparameters that do not have high computational volumes (usually less than the maximum value of the interval). For instance, by looking into the batch size values for both 30-min and 1-h intervals of the Las Vegas dataset, we understand that most of the assigned values are around and less than the median (the line shown in the figure). Such an interpretation applies to the other three hyperparameters and indicates the high capability of the proposed evolutionary search algorithm in initializing hyperparameters of the LSTM neural network.

To evaluate the proposed algorithm’s performance statistically, we demonstrate the boxplots of RMSE rates in Figs. 12 and 13 for the proposed EGOA-LSTM versus the other benchmarks in tackling Las Vegas dataset for two forecasting horizons. As seen from these two figures, in two forecasting horizons, the dominance of the proposed deep EGOA-LSTM is evident.

### 3.5.2 Denver case study

This section investigates the forecasting of the next 30-min and 1-h ahead for wind speed time series of Denver collected dataset. Tables 4 and 5 display the performance results of forecasting compared algorithms. The visualization for different forecasting algorithms based on the test set’s actual and predicted points is shown in Fig. 14.

Table 4 demonstrates that the proposed EGOA-LSTM model dominates other compared forecasting methods with the minimum value of RMSE as 0.042213, MAE as 0.028105, MAPE as 40.930122 and maximum value of  $R^2$  as 0.916746 for utmost short-term 30-min wind speed forecasting. Among compared prediction algorithms, the best forecasting performance is denoted to EGP-WNN with RMSE



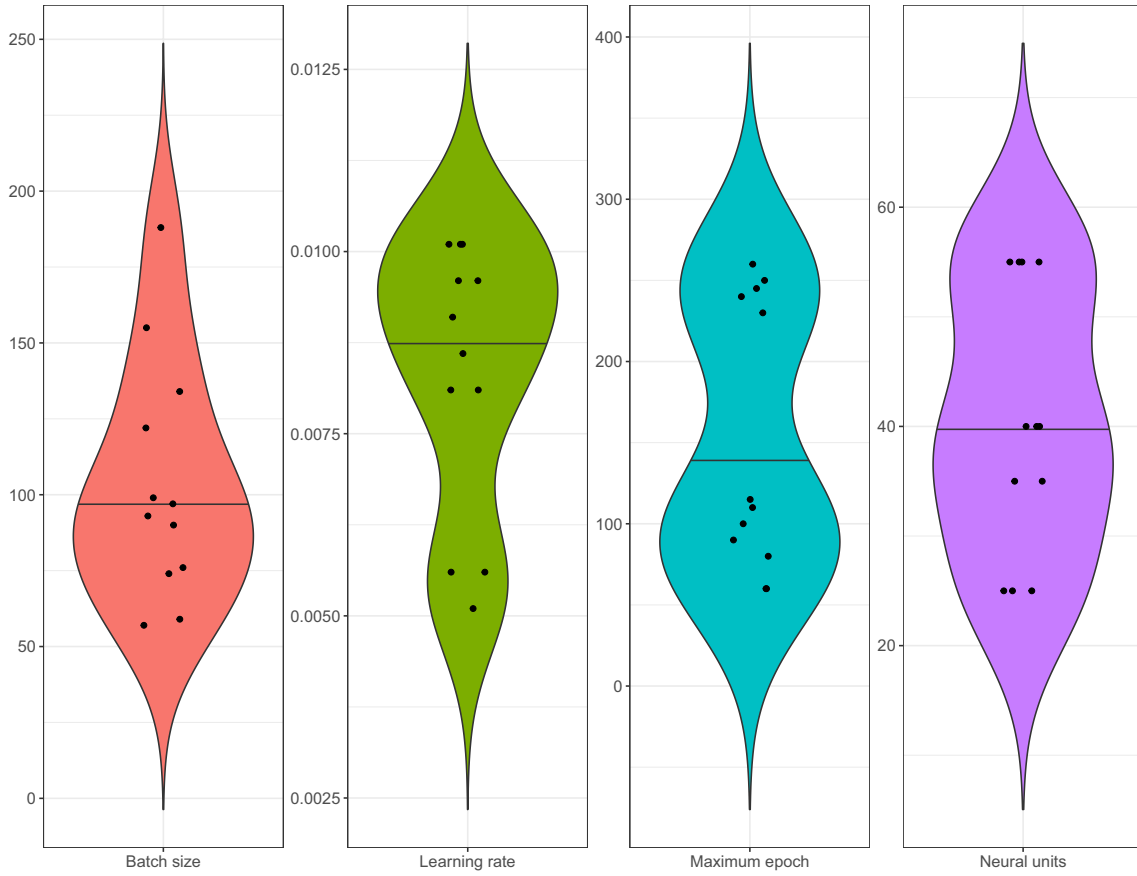
**Fig. 10** Violin diagram of the values obtained from the four hyperparameters used in the EGOA-LSTM algorithm for the Las Vegas case study of the 30-min ahead forecasting

as 0.045463, MAE as 0.030756, MAPE as 43.49489, and  $R^2$  as 0.896977. From Fig. 14, we notice that our novel deep neuroevolution method's actual and predicted points are met properly. We also observe such dominance of our proposed method in Table 5 for 1-h ahead wind speed forecasting with the maximum value of  $R^2$  as 0.746495, minimum values of RMSE as 0.071538, MAE as 0.049332 and MAPE as 73.45831 while the performance of the best predictive algorithm among compared models indices to EGP-WNN with RMSE as 0.073322, MAE as 0.052651, MAPE as 85.09159 and  $R^2$  as 0.74649. As it can be seen in Fig. 14, the wind speed predicted by the EGOA-LSTM model demonstrates more similarities with the actual data points and conducts fewer errors in the Denver case study.

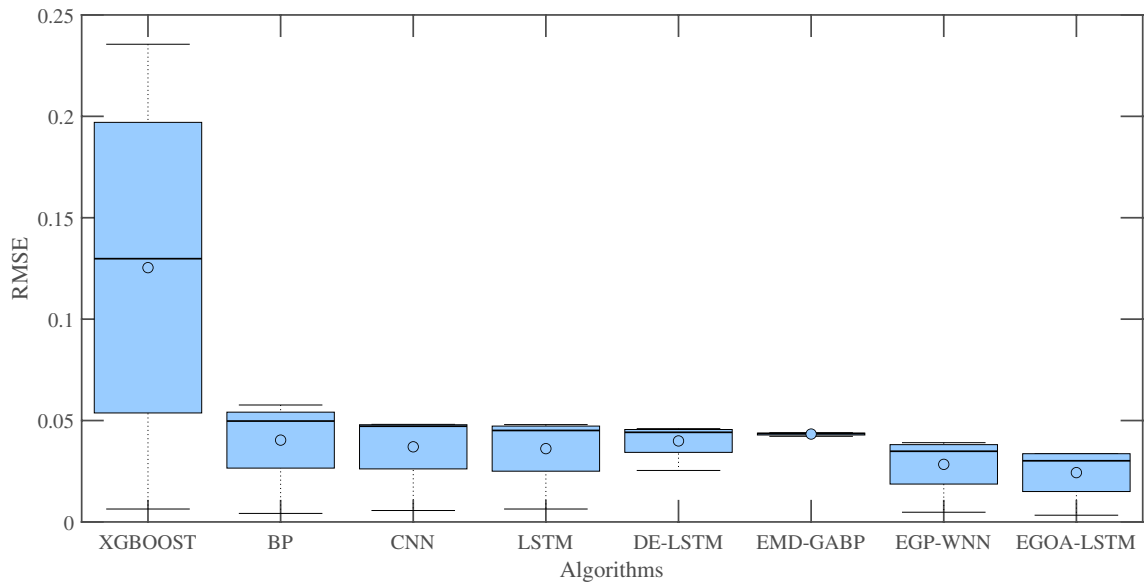
Figure 15 shows the convergence curve for the EGOA-LSTM algorithm using 30-min and 1-h ahead horizons for the Denver case study. Like the Las Vegas case study, EGOA-LSTM is easily converged to the maximum iteration number (20), and it generates fewer error values for 30-min ahead prediction compared with 1-h ahead horizon. Besides, four utilized LSTM hyperparameters involved in optimization procedures with EGOA obtain low computational

volumes of hyperparameters, as shown in Figs. 16 and 17. For example, the proposed algorithm for the initializing of learning rate hyperparameter in both cases mostly chooses values that are closer to the beginning of the interval or the median, indicating that the algorithm is effective in initializing the LSTM hyperparameters. Moreover, the boxplots of two forecasting horizons of the Denver case study are illustrated in Figs. 18 and 19. We notice from these two figures that the novel EGOA-LSTM performs better than all single and hybrid benchmarks. Finally, we present the best architectures obtained by the proposed algorithm for both databases in the 1-step (30-min) and 2-step (1-h) ahead time periods in Table 6. As an example, we can see that for the prediction of the next 30-min ahead of the Denver case study, the algorithm selects the maximum epoch = 50, the number of units = 23, batch size = 50, and learning rate = 0.0001, which results in the RMSE equal to 0.033562.

We focus on discussing and comparing the proposed EGOA-LSTM model with other conventional forecasting algorithms in a nutshell. We can notice from the experimental findings for utmost short-term wind speed forecasting and short-term wind speed forecasting for both case studies that



**Fig. 11** Violin diagram of the values obtained from the four hyperparameters used in the EGOA-LSTM algorithm for the Las Vegas case study of the 1-h ahead forecasting



**Fig. 12** RMSE box plots of all models for Las Vegas 30-min ahead forecasting

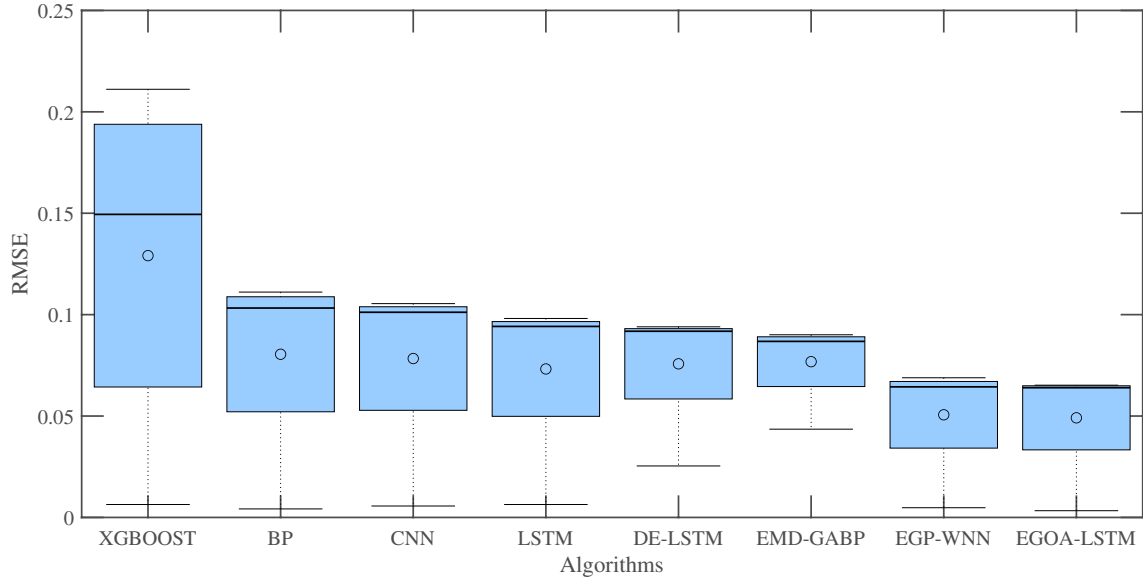


Fig. 13 RMSE box plots of all models for Las Vegas 1-h ahead forecasting

Table 4 Error estimated results of the predictions of the 30-min ahead wind speed time series for Denver dataset. The bold values represent the best performance evaluation metric

Algorithm	RMSE	MAE	$R^2$	MAPE
XGBOOST	0.150719	0.137569	0.584349	287.0161
BP	0.072462	0.052733	0.738284	70.73292
CNN	0.067939	0.050329	0.769935	72.37543
LSTM	0.054729	0.039344	0.850702	56.17056
DE-LSTM	0.056017	0.040672	0.843593	60.11669
EMD-GABP	0.053664	0.038637	0.856455	56.57227
EGP-WNN	0.045463	0.030756	0.896977	43.49489
EGOA-LSTM	<b>0.042213</b>	<b>0.028105</b>	<b>0.911182</b>	<b>40.93012</b>

Table 5 Error estimated results of the predictions of the 1-h ahead wind speed time series for Denver dataset. The bold values represent the best performance evaluation metric

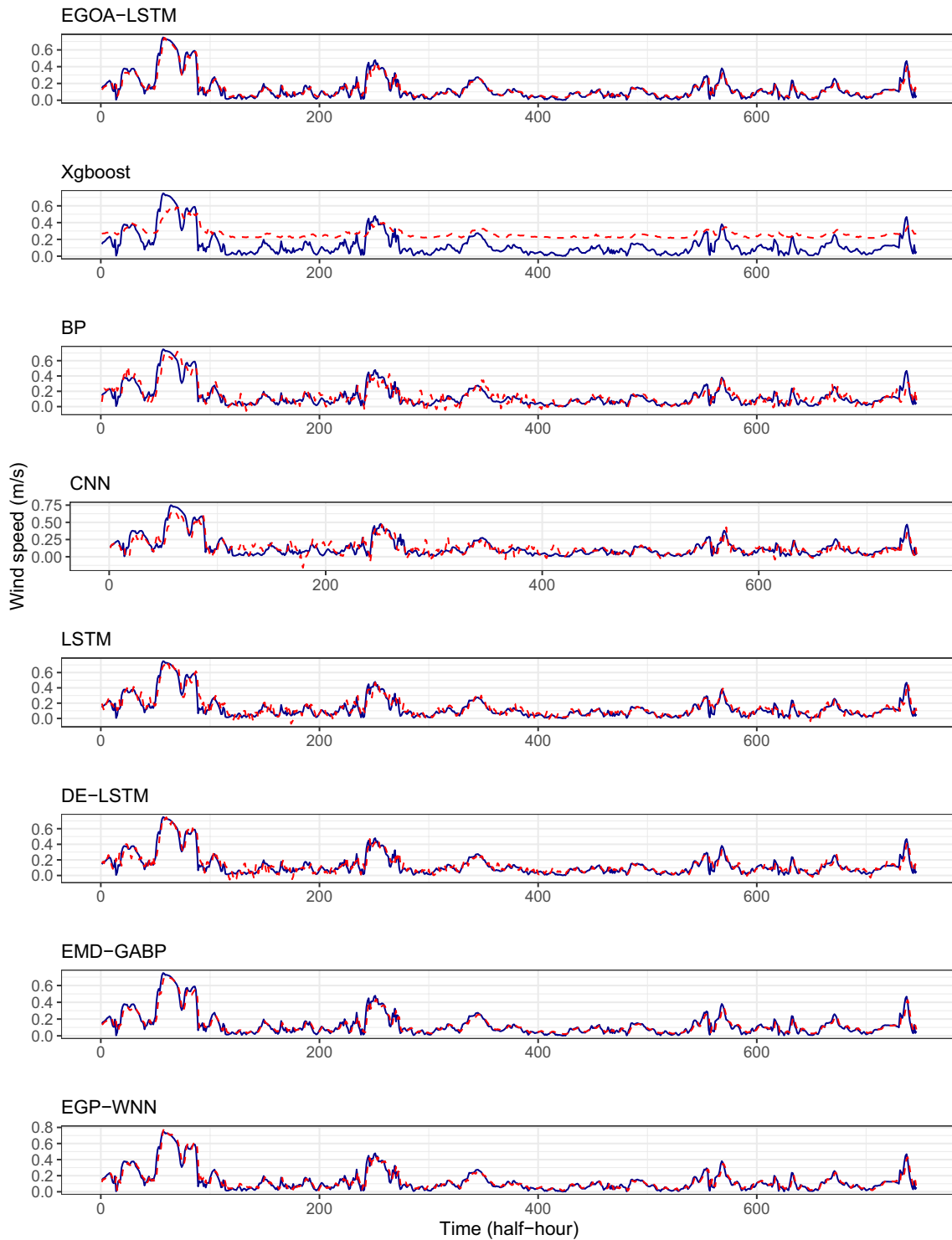
Algorithm	RMSE	MAE	$R^2$	MAPE
XGBOOST	0.164598	0.148963	0.342062	294.7785
BP	0.107107	0.080082	0.431719	116.0144
CNN	0.105921	0.078654	0.444242	113.3859
LSTM	0.112653	0.084376	0.371346	108.6512
DE-LSTM	0.109007	0.082086	0.411381	119.9755
EMD-GABP	0.105245	0.078292	0.451306	111.0744
EGP-WNN	0.073322	0.052651	0.733687	85.09159
EGOA-LSTM	<b>0.071538</b>	<b>0.049332</b>	<b>0.746495</b>	<b>73.45831</b>

EGOA-LSTM performs superior in error prediction indicators (RMSE, MAE, MAPE and  $R^2$ ) among comparative forecasting algorithms, including CNN, LSTM, BP, Xgboost, EMD-GABP, DE-LSTM, and EGP-WNN. Moreover, the

proposed EGOA-LSTM model meets most of the actual and predicted points of both case studies.

From both convergence curves for two case studies, we can easily understand that short-term wind speed forecasting is more costly and challenging than utmost short-term wind speed forecasting when the prediction time horizon was lengthened from 30 min to 1 h with wind speed time-horizon rising. Besides, to show the effectiveness of the proposed EGOA-LSTM from the statistical point, we evaluated it using the boxplots for different horizons of two datasets. The results indicate that the novel EGOA-LSTM performs superior compared to other benchmarks used in the experiments.

Based on the evaluation error results, EGP-WNN itself shows a robust performance among compared benchmarks. Our proposed deep neuroevolution, which optimizes the four key hyperparameters of LSTM networks, improves the generalization robustness and competency of single LSTMs. On the other hand, the results of optimized hyperparameters visualized in violin plots for two forecasting horizons of two datasets show that EGOA-LSTM has not chosen the complex and heavy values during optimization of LSTMs. This behavior shows the low cost-efficiency of the EGOA-LSTM algorithm. According to the discussions in this section, we conclude that the EGOA-LSTM algorithm proposed in this study is efficient and promising, which can be considered as a reliable alternative strategy for wind speed time series forecasting.



**Fig. 14** The wind speed forecasting results of 30-min ahead obtained by different algorithms on Denver case study

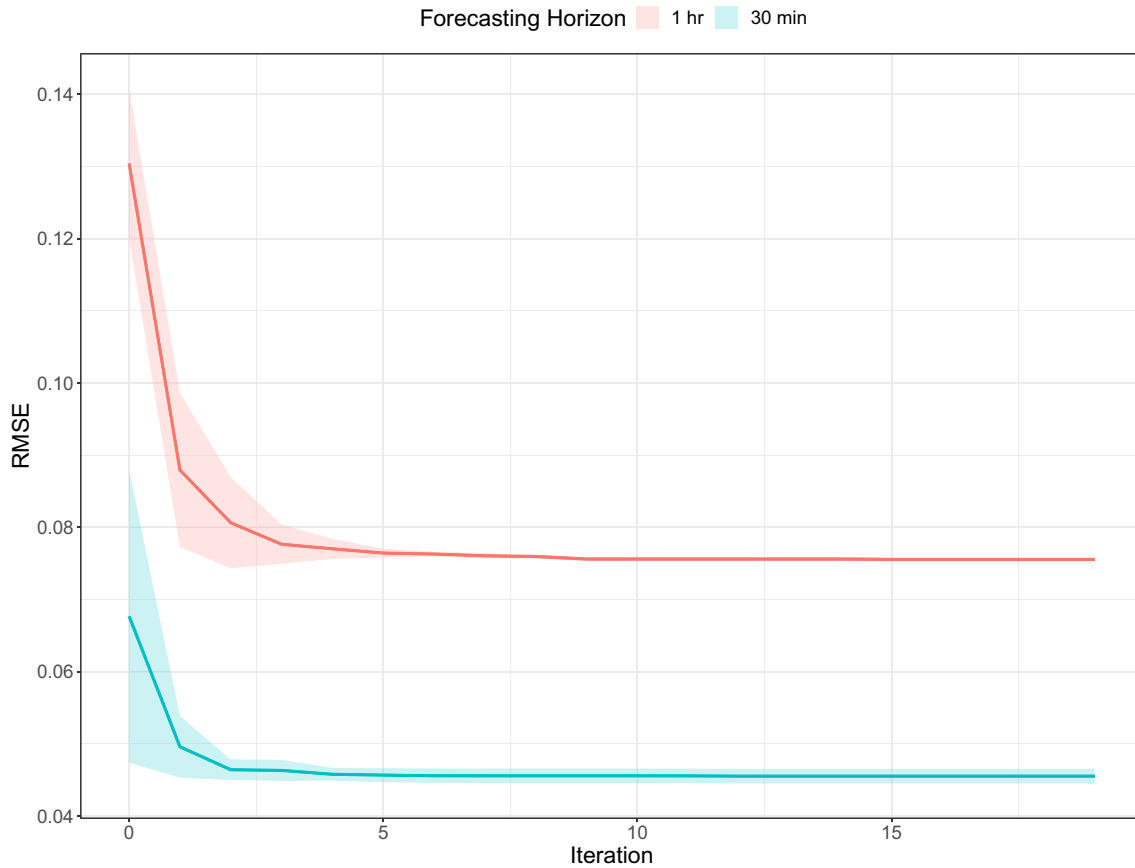
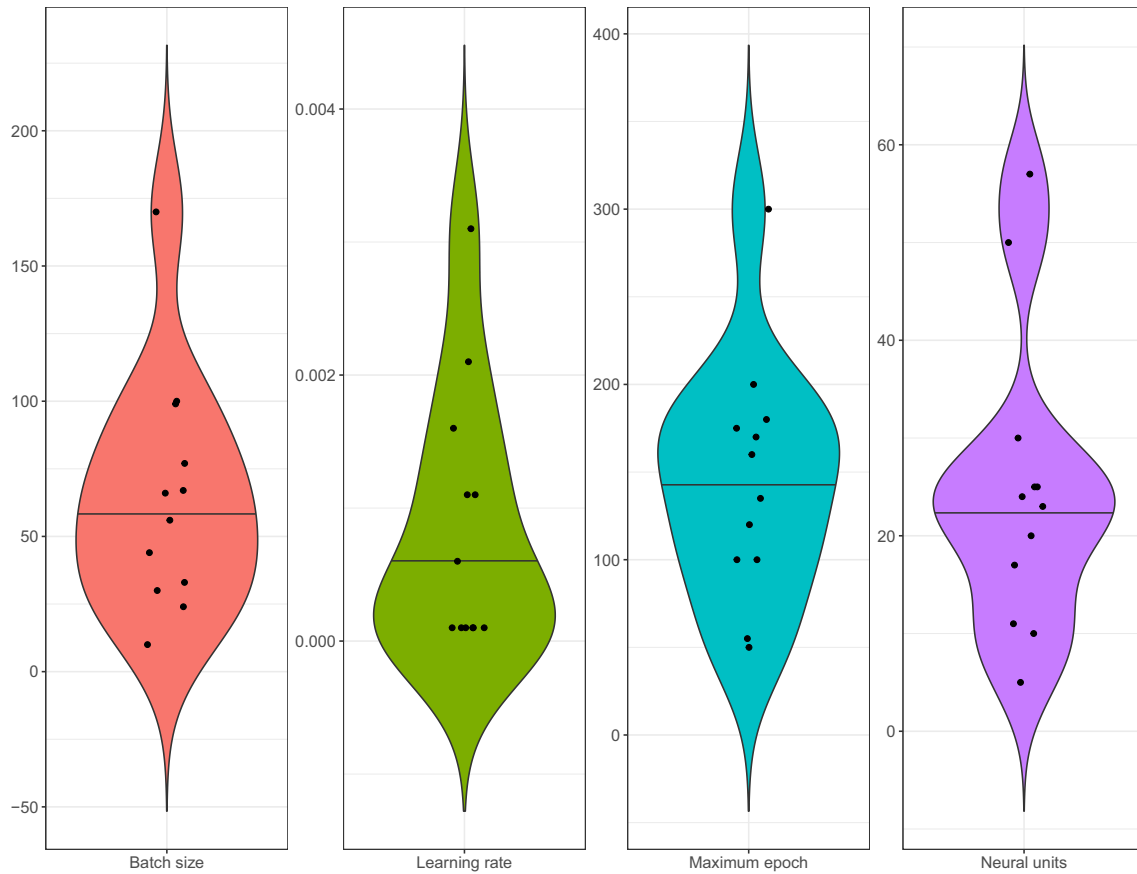


Fig. 15 The convergence profile of EGOA-LSTM algorithm for two forecasting horizons of Denver case study

#### 4 Conclusions and future directions

Wind speed forecasting is an essential problem in the conversion, consumption, and wind energy operation, which has received much attention in recent years. This paper presented a novel deep neuroevolution approach for wind speed forecasting, using the optimization of deep learning time series LSTM algorithm based on an enhanced version of GOA (EGOA) involving chaotic theory levy-flight operators. Involving these two powerful evolutionary operators into the original GOA makes adjusting and balancing the primary GOA exploration and exploitation phases. In this study, evolved LSTM neural networks were introduced to EGOA to optimize the hyperparameters of LSTMs to learn and predict the data of wind speed time series. To confirm the feasibility

of the proposed EGOA-LSTM, two data-collection case studies from two wind stations near Las Vegas and Denver in the USA were introduced to forecast the utmost short-term wind speeds including 30-min short-term wind speed and 1 h ahead. We used the mutual information as the feature selection strategy to determine our proposed deep learning model's optimal inputs. Compared to other prominent forecasting methods such as LSTM, CNN, BP, Xgboost, EMD-GABP, DE-LSTM, and EGP-WNN, our novel EGOA-LSTM algorithm obtained the best prediction performance with the minimum values of RMSE, MAE, and MAPE and the maximum value of  $R^2$ . Furthermore, the analysis of the evolved hyperparameters' impact on the forecasting performance of the LSTMs presented that the hyperparameters of LSTMs optimized by the EGOA obtained a low computational cost.

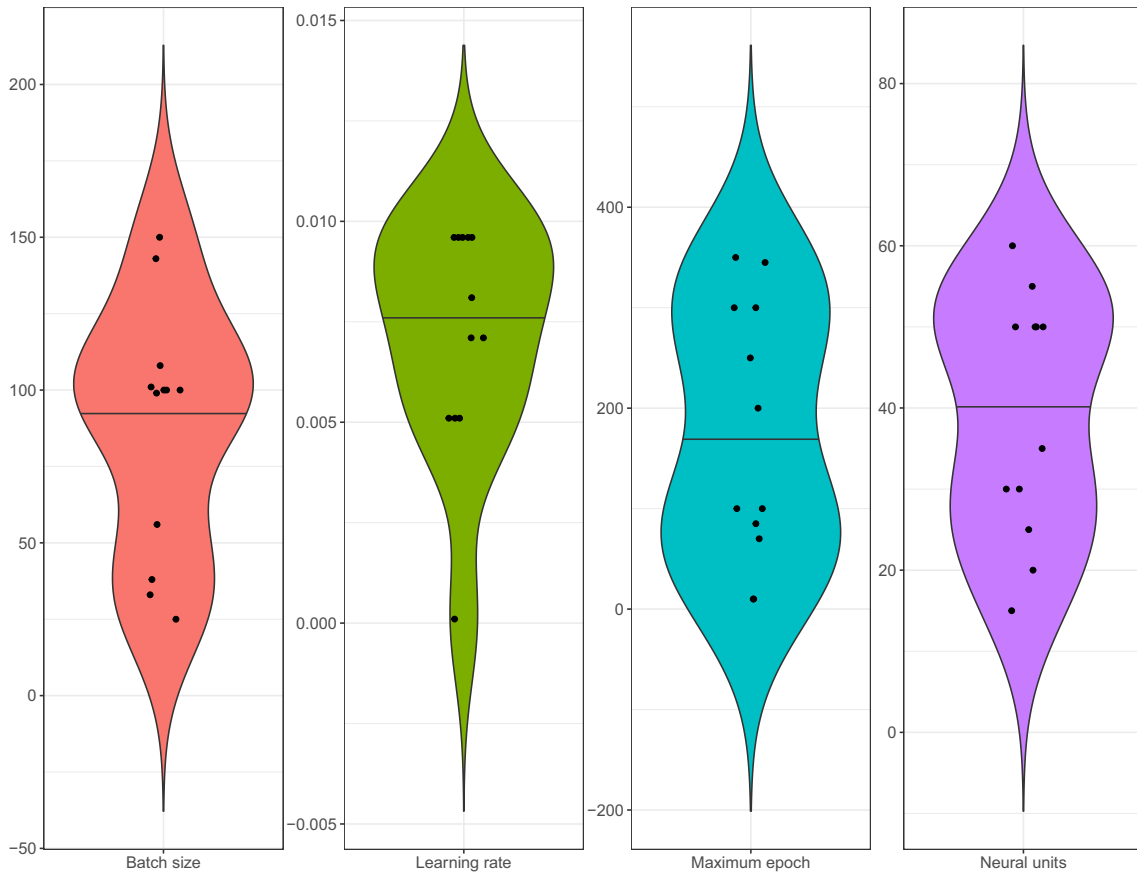


**Fig. 16** Violin diagram of the values obtained from the four hyperparameters used in the EGOA-LSTM algorithm for the Denver study of the 30-min ahead forecasting

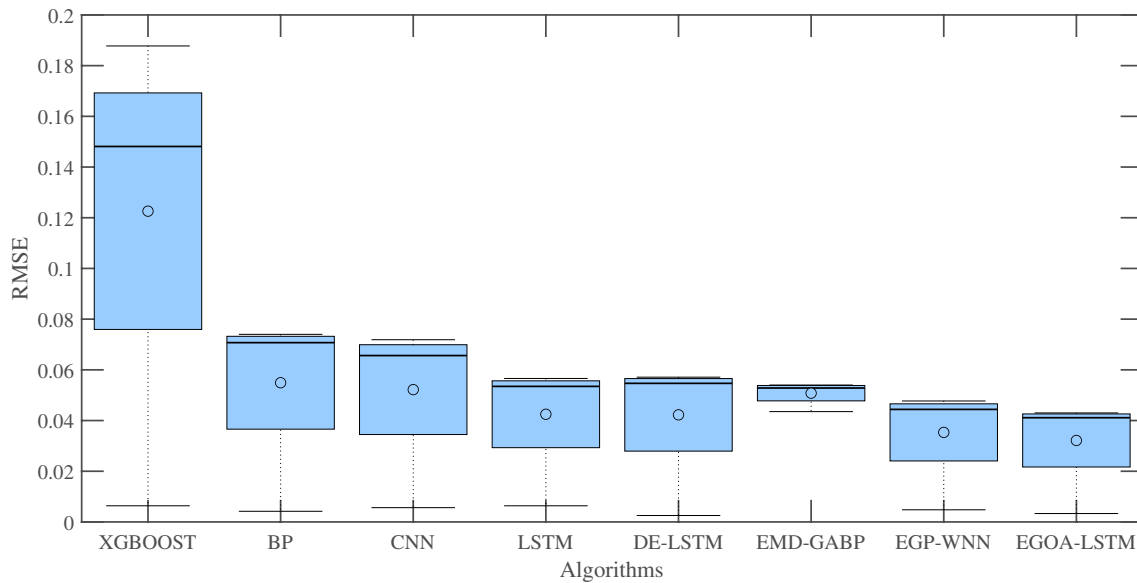
The proposed EGOA-LSTM algorithm achieved adequate wind speed forecasting performance based on the nonlinear-learning features of LSTMs and EGOA.

In this paper, we analyzed the univariate time series prediction for wind speed forecasting. For future works, the scholars can research for multivariate time series prediction of further complicated wind speed prediction based on more advanced deep neuroevolution models using more interdependent attributes such as power system statuses and weather conditions. Moreover, an attempt can be made to develop

more optimal deep learning algorithms to promote green energy resources forecasting. A further valuable orientation might be to expand the datasets' size, which would allow the training process more robust against over-fitting. An analysis of the wind speed results for the forecasts of the next few hours and multi-day would be undertaken as another future work. We may further use the proposed deep neuroevolution strategy proposed in this work to obtain probabilistic forecasts to quantify the corresponding uncertainties in the wind speed datasets. Also, the new GOA-based model can



**Fig. 17** Violin diagram of the values obtained from the four hyperparameters used in the EGOA-LSTM algorithm for the Denver case study of the 1-h ahead forecasting



**Fig. 18** RMSE box plots of all models for Denver 30-min ahead forecasting

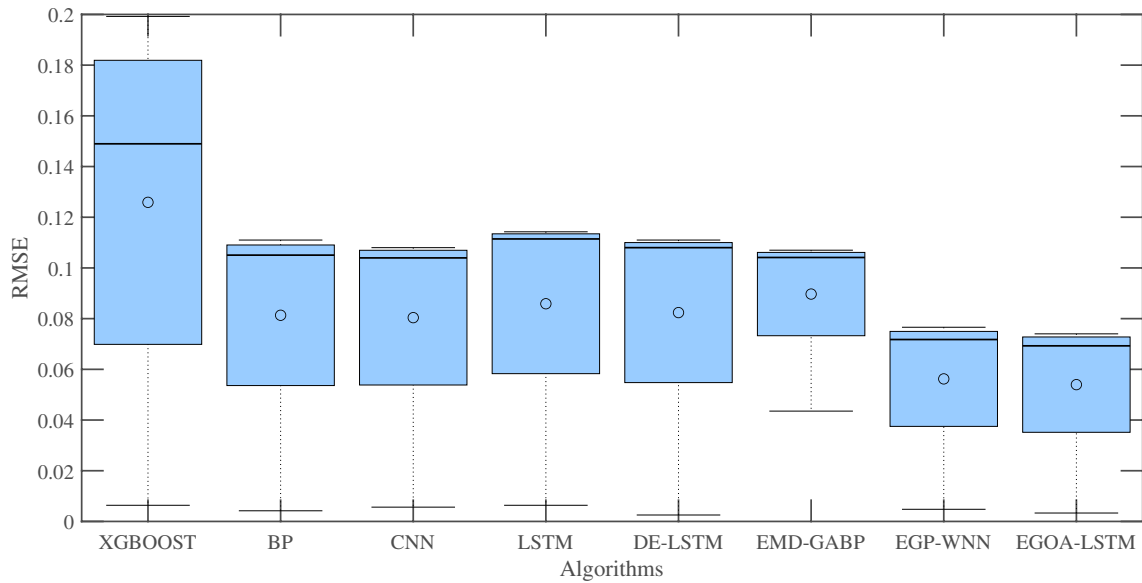


Fig. 19 RMSE box plots of all models for Denver 1-h ahead forecasting

Table 6 The best architectures obtained by EGOA-LSTM based on RMSE error metric

Dataset	Time-step	$M_e$	$N_u$	$B_s$	$L_r$	RMSE
Las Vegas	1	50	6	40	0.0011	0.030012
	2	70	17	70	0.0001	0.061139
Denver	1	50	23	50	0.0001	0.033562
	2	60	33	40	0.0006	0.068242

be applied to areas such as neural network-based robotic systems [113].

**Acknowledgements** This research was partially supported by the Australian Research Council Discovery Projects funding scheme (project DP190102181 and DP210101465).

## References

- Liu M, Cao Z, Zhang J, Wang L, Huang C, Luo X (2020) Short-term wind speed forecasting based on the jaya-svm model. *Int J Electric Power Energy Syst* 121:106056
- Watil A, El Magri A, Raihani A, Lajouad R, Giri F (2020) Multi-objective output feedback control strategy for a variable speed wind energy conversion system. *Int J Electric Power Energy Syst* 121:106081
- Abedi A, Rahimiyan M (2020) Day-ahead energy and reserve scheduling under correlated wind power production. *Int J Electric Power Energy Syst* 120:105931
- Wang J, Song Y, Liu F, Hou R (2016) Analysis and application of forecasting models in wind power integration: a review of multi-step-ahead wind speed forecasting models. *Renew Sustain Energy Rev* 60:960–981
- Hassan S, Khosravi A, Jaafar J (2015) Examining performance of aggregation algorithms for neural network-based electricity demand forecasting. *Int J Electric Power Energy Syst* 64:1098–1105
- Mahmoudi MR, Heydari MH, Avazzadeh Z, Pho K-H (2020) Goodness of fit test for almost cyclostationary processes. *Digit Signal Proc* 96:102597
- Mahmoudi MR, Maleki M, Pak A (2018) Testing the equality of two independent regression models. *Commun Stat-Theory Methods* 47:2919–2926
- Haghighin H, Mahmoudi MR, Shishebor Z (2015) Large sample inference on the ratio of two independent binomial proportions. *J Math Ext* 5:87–95
- Mahmoudi MR, Behboodian J, Maleki M (2017) Large sample inference about the ratio of means in two independent populations. *J Stat Theory Appl* 16:366–374
- Lydia M, Kumar SS, Selvakumar AI, Kumar GEP (2016) Linear and non-linear autoregressive models for short-term wind speed forecasting. *Energy Convers Manag* 112:115–124
- Ailliot P, Monbet V (2012) Markov-switching autoregressive models for wind time series. *Environ Model Softw* 30:92–101
- Torres JL, Garcia A, De Blas M, De Francisco A (2005) Forecast of hourly average wind speed with arma models in navarre (spain). *Sol Energy* 79:65–77
- Yunus K, Thiringer T, Chen P (2015) Arima-based frequency-decomposed modeling of wind speed time series. *IEEE Trans Power Syst* 31:2546–2556
- Jahangir H, Golkar MA, Alhameli F, Mazouz A, Ahmadian A, Elkamel A (2020) Short-term wind speed forecasting framework based on stacked denoising auto-encoders with rough ann. *Sustain Energy Technol Assess* 38:100601
- Zhang X, Wang D, Zhou Z, Ma MYJITOPA (2019) Intelligence, robust low-rank tensor recovery with rectification and alignment.

- IEEE Trans Pattern Anal Mach Intell. <https://doi.org/10.1109/TPAMI.2019.2929043>
16. Zhang X, Wang T, Wang J, Tang G, Zhao L (2020) Pyramid channel-based feature attention network for image dehazing. *Comput Vis Image Understand* 197–198:103003. <http://www.sciencedirect.com/science/article/pii/S1077314220300709>
  17. Zhang X, Jiang R, Wang T, Wang JJITOC (2020) S. f. V. technology, recursive neural network for video deblurring. *IEEE Trans Circ Syst Video Technol.* <https://doi.org/10.1109/TCSVT.2020.3035722>
  18. Zhang X, Wang T, Luo W, Huang PJITOC (2020) S. f. V. Technology, Multi-level fusion and attention-guided cnn for image dehazing. *IEEE Trans Circ Syst Video Technol.* <https://doi.org/10.1109/TCSVT.2020.3046625>
  19. Zhang X, Wang J, Wang T, Jiang R, Xu J, Zhao LJIS (2020) Robust feature learning for adversarial defense via hierarchical feature alignment. *Inf Sci.* <https://doi.org/10.1016/j.ins.2020.12.042>
  20. Jalali SMJ, Moro S, Mahmoudi MR, Ghaffary KA, Maleki M, Alidoostan A (2017) A comparative analysis of classifiers in cancer prediction using multiple data mining techniques. In *J Bus Intell Syst Eng* 1:166–178
  21. Jalali SMJ, Khosravi A, Alizadehsani R, Salaken SM, Kebria PM, Puri R, Nahavandi S (2019) Parsimonious evolutionary-based model development for detecting artery disease. In: 2019 IEEE International Conference on industrial technology (ICIT), IEEE, pp 800–805
  22. Jalali SMJ, Ahmadian S, Khosravi A, Mirjalili S, Mahmoudi MR, Nahavandi S (2020) Neuroevolution-based autonomous robot navigation: a comparative study. *Cogn Syst Res* 62:35–43
  23. Mousavirad SJ, Schaefer G, Jalali SMJ, Korovin I (2020) A benchmark of recent population-based metaheuristic algorithms for multi-layer neural network training. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference companion*, pp 1402–1408
  24. Jalali SMJ, Ahmadian S, Kebria PM, Khosravi A, Lim CP, Nahavandi S (2019) Evolving artificial neural networks using butterfly optimization algorithm for data classification. In: *International Conference on neural information processing*, Springer, pp 596–607
  25. Hasani H, Jalali SMJ, Rezaei D, Maleki M (2018) A data mining framework for classification of organisational performance based on rough set theory. *Asian J Manag Sci Appl* 3:156–180
  26. Jalali SMJ, Kebria PM, Khosravi A, Saleh K, Nahavandi D, Nahavandi S (2019) Optimal autonomous driving through deep imitation learning and neuroevolution. In: 2019 IEEE International Conference on systems, man and cybernetics (SMC), IEEE, pp 1215–1220
  27. Mousavirad SJ, Jalali SMJ, Ahmadian S, Khosravi A, Schaefer G, Nahavandi S (2020) Neural network training using a biogeography-based learning strategy. In: *International Conference on neural information processing*, Springer, pp 147–155
  28. Jalali SMJ, Khosravi A, Kebria PM, Hedjam R, Nahavandi S (2019) Autonomous robot navigation system using the evolutionary multi-verse optimizer algorithm. In: 2019 IEEE International Conference on systems, man and cybernetics (SMC), IEEE, pp 1221–1226
  29. Ahmadian S, Khanteymooi AR (2015) Training back propagation neural networks using asexual reproduction optimization. In: 2015 7th Conference on information and knowledge technology (IKT), IEEE, pp 1–6
  30. Quan H, Srinivasan D, Khosravi A (2016) Integration of renewable generation uncertainties into stochastic unit commitment considering reserve and risk: A comparative study. *Energy* 103:735–745
  31. Qiu T, Shi X, Wang J, Li Y, Qu S, Cheng Q, Cui T, Sui S (2019) Deep learning: a rapid and efficient route to automatic metasurface design. *Adv Sci* 6:1900128
  32. Li C, Hou L, Sharma BY, Li H, Chen C, Li Y, Zhao X, Huang H, Cai Z, Chen HJCMPI (2018) Biomedicine, developing a new intelligent system for the diagnosis of tuberculous pleural effusion. *Comput Methods Programs Biomed* 153:211–225
  33. Wang M, Chen HJASC (2020) Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. *Appl Soft Comput* 88:105946
  34. Chen H-L, Wang G, Ma C, Cai Z-N, Liu W-B, Wang S-JJN (2016) An efficient hybrid kernel extreme learning machine approach for early diagnosis of parkinsons disease. *Neurocomputing* 184:131–144
  35. Kong X, Liu X, Shi R, Lee KY (2015) Wind speed prediction using reduced support vector machines with feature selection. *Neurocomputing* 169:449–456
  36. Yu C, Li Y, Bao Y, Tang H, Zhai G (2018) A novel framework for wind speed prediction based on recurrent neural networks and support vector machine. *Energy Convers Manag* 178:137–145
  37. Yang S, Deng B, Wang J, Li H, Lu M, Che Y, Wei X, Loparo KA (2019) Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. *IEEE Trans Neural Netw Learn Syst* 31:148–162
  38. Zhang Y, Liu R, Heidari AA, Wang X, Chen Y, Wang M, Chen HJN (2020) Towards augmented kernel extreme learning models for bankruptcy prediction: algorithmic behavior and comprehensive analysis. *Neurocomputing.* <https://doi.org/10.1016/j.neucom.2020.10.038>
  39. Xia J, Chen H, Li Q, Zhou M, Chen L, Cai Z, Fang Y, Zhou H. J. C. m. (2017) p. i. biomedicine, Ultrasound-based differentiation of malignant and benign thyroid nodules: An extreme learning machine approach. *Comput Methods Programs Biomed* 147:37–49
  40. Chen H, Qiao H, Xu L, Feng Q, Cai K (2019) A fuzzy optimization strategy for the implementation of rbf lssvr model in vis-nir analysis of pomelo maturity. *IEEE Trans Ind Inf* 15:5971–5979
  41. Cadenas E, Rivera W (2009) Short term wind speed forecasting in la Venta, Oaxaca, México, using artificial neural networks. *Renew Energy* 34:274–278
  42. Guo Z-H, Wu J, Lu H-Y, Wang J-Z (2011) A case study on a hybrid wind speed forecasting method using bp neural network. *Knowl-Based Syst* 24:1048–1056
  43. Wang J, Du P, Niu T, Yang W (2017) A novel hybrid system based on a new proposed algorithm—multi-objective whale optimization algorithm for wind speed forecasting. *Appl Energy* 208:344–360
  44. Tian C, Hao Y, Hu J (2018) A novel wind speed forecasting system based on hybrid data preprocessing and multi-objective optimization. *Appl Energy* 231:301–319
  45. Salcedo-Sanz S, Pastor-Sánchez A, Prieto L, Blanco-Aguilera A, García-Herrera R (2014) Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization-extreme learning machine approach. *Energy Convers Manag* 87:10–18
  46. Zhao X, Zhang X, Cai Z, Tian X, Wang X, Huang Y, Chen H, Hu L. J. C. b. (2019) chemistry, Chaos enhanced grey wolf optimization wrapped elm for diagnosis of paraquat-poisoned patients. *Comput Biol Chem* 78:481–490
  47. Wang M, Chen H, Yang B, Zhao X, Hu L, Cai Z, Huang H, Tong CJN (2017) Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. *Neurocomputing* 267:69–84
  48. Zhang C, Wei H, Xie L, Shen Y, Zhang K (2016) Direct interval forecasting of wind speed using radial basis function neural networks in a multi-objective optimization framework. *Neurocomputing* 205:53–63

49. Zhang H, Qiu Z, Cao J, Abdel-Aty M, Xiong L (2019) Event-triggered synchronization for neutral-type semi-Markovian neural networks with partial mode-dependent time-varying delays. *IEEE Trans Neural Netw Learn Syst* 31:4437–4450
50. Lv Z, Qiao L (2020) Deep belief network and linear perceptron based cognitive computing for collaborative robots. *Appl Soft Comput* 92:106300
51. Khodayar M, Khodayar ME, Jalali SMJ (2021) Deep learning for pattern recognition of photovoltaic energy generation. *Electric J* 34:106882
52. Chen J, Zeng G-Q, Zhou W, Du W, Lu K-D (2018) Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization. *Energy Convers Manag* 165:681–695
53. Liu H, Mi X-W, Li Y-F (2018) Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and elman neural network. *Energy Convers Manag* 156:498–514
54. Hu L, Hong G, Ma J, Wang X, Chen H. J. C. i. B. (2015) Medicine, An efficient machine learning approach for diagnosis of paraquat-poisoned patients. *Comput Biol Med* 59:116–124
55. Shen L, Chen H, Yu Z, Kang W, Zhang B, Li H, Yang B, Liu DJK-BS (2016) Evolving support vector machines using fruit fly optimization for medical data classification. *Knowl-Based Syst* 96:61–75
56. Pei S, Qin H, Zhang Z, Yao L, Wang Y, Wang C, Liu Y, Jiang Z, Zhou J, Yi T (2019) Wind speed prediction method based on empirical wavelet transform and new cell update long short-term memory network. *Energy Convers Manag* 196:779–792
57. Khodayar M, Kaynak O, Khodayar ME (2017) Rough deep neural architecture for short-term wind speed forecasting. *IEEE Trans Ind Inf* 13:2770–2779
58. Li T, Xu M, Zhu C, Yang R, Wang Z, Guan Z (2019) A deep learning approach for multi-frame in-loop filter of hevcc. *IEEE Trans Image Process* 28:5663–5678
59. Chen H, Chen A, Xu L, Xie H, Qiao H, Lin Q, Cai K (2020) A deep learning cnn architecture applied in smart near-infrared analysis of water pollution for agricultural irrigation resources. *Agric Water Manag* 240:106303
60. Hu H, Wang L, Tao R (2021) Wind speed forecasting based on variational mode decomposition and improved echo state network. *Renew Energy* 164:729–751
61. Mousavi AA, Zhang C, Masri SF, Gholipour G (2020) Structural damage localization and quantification based on a ceemdan Hilbert transform neural network approach: a model steel truss bridge case study. *Sensors* 20:1271
62. Peng Z, Peng S, Fu L, Lu B, Tang J, Wang K, Li W (2020) A novel deep learning ensemble model with data denoising for short-term wind speed forecasting. *Energy Convers Manag* 207:112524
63. Hong Y-Y, Satriani TRA (2020) Day-ahead spatiotemporal wind speed forecasting using robust design-based deep learning neural network. *Energy* 209:118441
64. Qian J, Feng S, Tao T, Hu Y, Li Y, Chen Q, Zuo C (2020) Deep-learning-enabled geometric constraints and phase unwrapping for single-shot absolute 3d shape measurement. *APL Photon* 5:046105
65. Qian J, Feng S, Li Y, Tao T, Han J, Chen Q, Zuo C (2020) Single-shot absolute 3d shape measurement with deep-learning-based color fringe projection profilometry. *Opt Lett* 45:1842–1845
66. Wu Y-X, Wu Q-B, Zhu J-Q (2019) Data-driven wind speed forecasting using deep feature extraction and lstm. *IET Renew Power Gener* 13:2062–2069
67. Yu R, Gao J, Yu M, Lu W, Xu T, Zhao M, Zhang J, Zhang R, Zhang Z (2019) Lstm-efg for wind power forecasting based on sequential correlation features. *Future Gener Comput Syst* 93:33–42
68. Wang B, Zhang L, Ma H, Wang H, Wan S (2019) Parallel lstm-based regional integrated energy system multienergy source-load information interactive energy prediction. *Complexity* 2019:1–13
69. Sun G, Li C, Deng L (2021) An adaptive regeneration framework based on search space adjustment for differential evolution. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-021-05708-1>
70. Cao Y, Li Y, Zhang G, Jermsittiparsert K, Nasserri M (2020) An efficient terminal voltage control for pemfc based on an improved version of whale optimization algorithm. *Energy Rep* 6:530–542
71. Bai B, Guo Z, Zhou C, Zhang W, Zhang J (2021) Application of adaptive reliability importance sampling-based extended domain pso on single mode failure in reliability engineering. *Inf Sci* 546:42–59
72. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. *Adv Eng Softw* 105:30–47
73. Saxena A, Shekhawat S, Kumar R (2018) Application and development of enhanced chaotic grasshopper optimization algorithms. *Model Simul Eng* 2018:1–14
74. Xu Z, Hu Z, Heidari AA, Wang M, Zhao X, Chen H, Cai X (2020) Orthogonally-designed adapted grasshopper optimization: a comprehensive analysis. *Expert Syst Appl* 150:113282
75. Yu C, Chen M, Cheng K, Zhao X, Ma C, Kuang F, Chen HJEW (2021) Sgoa: annealing-behaved grasshopper optimizer for global tasks. *Eng Comput*. <https://doi.org/10.1007/s00366-020-01234-1>
76. Abualigah L, Diabat A (2020) A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications. *Neural Comput Appl* 32:1–24
77. Wang B, Zhang B, Liu X (2021) An image encryption approach on the basis of a time delay chaotic system. *Optik* 225:165737
78. Jiang Q, Wang G, Jin S, Li Y, Wang Y (2013) Predicting human microRNA-disease associations based on support vector machine. *Int J Data Min Bioinform* 8:282–293
79. Song X, Liu Y, Xue L, Wang J, Zhang J, Wang J, Jiang L, Cheng Z (2020) Time-series well performance prediction based on long short-term memory (lstm) neural network model. *J Petrol Sci Eng* 186:106682
80. Chang Z, Zhang Y, Chen W (2019) Electricity price prediction based on hybrid model of adam optimized lstm neural network and wavelet transform. *Energy* 187:115804
81. Wang H, Wang G, Li G, Peng J, Liu Y (2016) Deep belief network based deterministic and probabilistic wind speed forecasting approach. *Appl Energy* 182:80–93
82. Liu H, Mi X, Li Y (2018) Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, lstm network and elm. *Energy Convers Manag* 159:54–64
83. Ghimire S, Deo RC, Raj N, Mi J (2019) Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. *Appl Energy* 253:113541
84. Qing X, Niu Y (2018) Hourly day-ahead solar irradiance prediction using weather forecasts by lstm. *Energy* 148:461–468
85. Zahid M, Ahmed F, Javaid N, Abbasi RA, Kazmi Z, Syeda H, Javaid A, Bilal M, Akbar M, Ilahi M (2019) Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids. *Electronics* 8:122
86. Li L (2019) Geographically weighted machine learning and downscaling for high-resolution spatiotemporal estimations of wind speed. *Remote Sens* 11:1378
87. Wang S, Zhang N, Wu L, Wang Y (2016) Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and ga-bp neural network method. *Renew Energy* 94:629–636

88. Peng L, Liu S, Liu R, Wang L (2018) Effective long short-term memory with differential evolution algorithm for electricity price prediction. *Energy* 162:1301–1314
89. Filik T (2016) Improved spatio-temporal linear models for very short-term wind speed forecasting. *Energies* 9:168
90. Xu Y, Chen H, Luo J, Zhang Q, Jiao S, Zhang XJIS (2019) Enhanced moth-flame optimizer with mutation strategy for global optimization. *Inf Sci* 492:181–203
91. Zhao D, Liu L, Yu F, Heidari AA, Wang M, Liang G, Muhammad K, Chen HJK-BS (2020) Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2d Kapur entropy. *Knowl-Based Syst* 216:106510
92. Tu J, Chen H, Liu J, Heidari AA, Zhang X, Wang M, Ruby R, Pham Q-VJK-BS (2021) Evolutionary biogeography-based whale optimization methods with communication structure: towards measuring the balance. *Knowl-Based Syst* 212:106642
93. Shan W, Qiao Z, Heidari AA, Chen H, Turabieh H, Teng YJK-BS (2020) Double adaptive weights for stabilization of moth flame optimizer: balance analysis, engineering cases, and medical diagnosis. *Knowl-Based Syst* 214:106728
94. Hu J, Chen H, Heidari AA, Wang M, Zhang X, Chen Y, Pan ZJK-BS (2020) Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection. *Knowl-Based Syst* 213:106684
95. Yu H, Li W, Chen C, Liang J, Gui W, Wang M, Chen HJEWc (2020) Dynamic gaussian bare-bones fruit fly optimizers with abandonment mechanism: method and analysis. *Eng Comput* 1–29. <https://doi.org/10.1007/s00366-020-01174-w>
96. Xu X, Chen H-LJSC (2014) Adaptive computational chemotaxis based on field in bacterial foraging optimization. *Soft Comput* 18:797–807
97. Chen H, Heidari AA, Chen H, Wang M, Pan Z, Gandomi AH (2020) Multi-population differential evolution-assisted Harris hawks optimization: framework and case studies. *Future Gener Comput Syst* 111:175–198. <https://doi.org/10.1016/j.future.2020.04.008>. <http://www.sciencedirect.com/science/article/pii/S0167739X19313263>. Accessed Oct 2020
98. Wu T, Cao J, Xiong L, Zhang H (2019) New stabilization results for semi-markov chaotic systems with fuzzy sampled-data control. *Complexity* 2019
99. Shi K, Tang Y, Zhong S, Yin C, Huang X, Wang W (2018) Non-fragile asynchronous control for uncertain chaotic lurie network systems with bernoulli stochastic process. *Int J Robust Nonlinear Control* 28:1693–1714
100. Liu J, Wu C, Wu G, Wang X (2015) A novel differential search algorithm and applications for structure design. *Appl Math Comput* 268:246–269
101. Shi K, Tang Y, Liu X, Zhong S (2017) Non-fragile sampled-data robust synchronization of uncertain delayed chaotic Lurie systems with randomly occurring controller gain fluctuation. *ISA Trans* 66:185–199
102. Fan Q, Chen Z, Li Z, Xia Z, Yu J, Wang D (2020) A new improved whale optimization algorithm with joint search mechanisms for high-dimensional global optimization problems. *Eng Comput* 1–28. <https://doi.org/10.1007/s00366-019-00917-8>
103. Hakli H, Uğuz H (2014) A novel particle swarm optimization algorithm with levy flight. *Appl Soft Comput* 23:333–345
104. Western wind data set, <https://www.nrel.gov/grid/western-wind-data.html>, [online] Accessed 15 Jan 2020
105. Zhao X, Li D, Yang B, Ma C, Zhu Y, Chen HJASC (2014) Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton. *Appl Soft Comput* 24:585–596
106. Zhang Y, Liu R, Wang X, Chen H, Li C. J. s. (2020) Boosted binary Harris hawks optimizer and feature selection. *Eng Comput* 25:26
107. Zhang X, Fan M, Wang D, Zhou P, Tao DJITONN, Systems L (2020) Top-k feature selection framework using robust 0–1 integer programming. *IEEE Trans Neural Netw Learn Syst*. <https://doi.org/10.1109/TNNLS.2020.3009209>
108. Bhaskar K, Singh S (2012) Awnn-assisted wind power forecasting using feed-forward neural network. *IEEE Trans Sustain Energy* 3:306–315
109. Sagheer A, Kotb M (2019) Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing* 323:203–213
110. Cao J, Li Z, Li J (2019) Financial time series forecasting model based on ceemdan and lstm. *Phys A* 519:127–139
111. Sagheer A, Kotb M (2019) Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems. *Sci Rep* 9:1–16
112. Shida H, Fei, G Quan Z, Ding H (2020) Mrmd2.0: A python tool for machine learning with feature ranking and reduction. *Curr Bioinform* 15: 1213–1221. <https://doi.org/10.2174/1574893615999200503030350>. <http://www.eurekaselect.com/node/181578/article>. Accessed Feb 2021
113. Ding L, Li S, Gao H, Chen C, Deng Z (2018) Adaptive partial reinforcement learning neural network-based tracking control for wheeled mobile robotic systems. *IEEE Trans Syst Man Cybern Syst* 50:2512–2523