# Solar Irradiance Forecasting using a Novel Hybrid Deep Ensemble Reinforcement Learning Algorithm

Seyed Mohammad Jafar Jalali[a,*], Sajad Ahmadian[b], Bahareh Nakisa[c], Mahdi Khodayar[d], Abbas Khosravi[a], Saeid Nahavandi[a], Syed Mohammed Shamsul Islam[e], Miadreza Shafie-khah[f], João P.S. Catalão[g]

[a] *Institute for Intelligent Systems Research and Innovation, (IISRI), Deakin University, Geelong, Australia*

[b] *Faculty of Information Technology, Kermanshah University of Technology, Kermanshah, Iran*

[c] *School of Information Technology, Faculty of Science Engineering and Built Environment, Deakin University, Geelong, Vic, Australia*

[d] *Department of Computer Science, University of Tulsa, Tulsa, USA*

[e] *School of Science, Edith Cowan University, Western Australia, Australia*

[f] *School of Technology and Innovations, University of Vaasa, Vaasa, Finland*

[g] *Faculty of Engineering of University of Porto and INESC TEC, Porto, Portugal*

---

**Abstract** Solar irradiance forecasting is a major priority for the power transmission systems in order to generate and incorporate the performance of massive photovoltaic plants efficiently. As such, prior forecasting techniques that use classical modelling and single deep learning models that undertake feature extraction procedures manually were unable to meet the output demands in specific situations with dynamic variability. Therefore, in this study, we propose an efficient novel hybrid solar irradiance forecasting model based on three steps. In the first step, we employ a powerful variable input selection strategy named as partial mutual information (PMI) to calculate the linear and nonlinear correlations of the original solar irradiance data. In the second step, unlike the traditional deep learning models designing their architectures manually,

*Corresponding author

*Email addresses:* mohammadjj.it@gmail.com (Seyed Mohammad Jafar Jalali), s.ahmadian239@gmail.com (Sajad Ahmadian), bahar.nakisa@deakin.edu.au (Bahareh Nakisa), mak1736@utulsa.edu (Mahdi Khodayar), abbas.khosravi@deakin.edu.au (Abbas Khosravi), saeid.nahavandi@deakin.edu.au (Saeid Nahavandi), syed.islam@ecu.edu.au (Syed Mohammed Shamsul Islam), mshafiek@uwasa.fi (Miadreza Shafie-khah), catalao@fe.up.pt (João P.S. Catalão)

we utilize several deep long short term memory–convolutional neural network (LSTM–CNN) models optimized by a novel modified whale optimization algorithm in order to compute the forecasting results of the solar irradiance datasets. Finally, in the third step, we deploy a deep reinforcement learning strategy for selecting the best subset of the combined deep optimized LSTM–CNN models. Through analysing the forecasting results over two real-world datasets gathered from the USA solar irradiance stations, it can be inferred that our proposed algorithm outperforms other powerful benchmarked algorithms in 1-step, 2-step, 12-step, and 24-step ahead forecasting.

*Keywords:* Solar irradiance forecasting, Deep neural networks, Evolutionary computation, Ensemble strategy, Deep reinforcement learning.

## 1. Introduction

In recent years, the serious concerns of environmental protection using non-conventional renewable energy sources (NRES) such as solar photovoltaic (PV) systems have become an attractive alternative. PV systems and isolated PV panels as one of the most common solar energy applications have been developed widely all around the world. It has been reported that the annual growth rate of PVs as one of the most key emerging technologies has been over 40% on average. Several market leaders predicted that revenues in the photovoltaics, innovations, and industries will be doubled, from 35-40 billion euros in 2010 to 70 billion euros in 2015 1. The main attribute of the solar resources that need to have well characterized in order to operate effectively in photovoltaic and concentrated solar energy plants are, in particular, global horizontal irradiance (GHI).

Although NRES can bring us several advantages, they are quite challenging because of mismatch between power supply and energy demand, generating issues on stability, safety, reliability and frequency response of the grid 2. There are several studies that focus on implementation of storage 3, 4 and demand response 5. However, to have an appropriate storage

energy management, we need to forecast the source availability 6, 7, 8. It should be noted that weather variability can cause fluctuation of solar irradiation and affect the power supply from grid-connected photovoltaic plant. This fluctuation in solar irradiation is causing considerable difficulty in creating balance between power storage and demand responses 9. To solve this issue, it is essential to predict grid-connected power supply of PV plants. A prediction system can offer several beneficial inputs and raw data for different regional power system operation activities including economical transportation management of grid connection and safety evaluation. An accurate solar irradiance forecasting system will be necessary for the established generators to schedule various energy plants for maintaining resources, and provide more details upon on the solar energy trades. This system can increase the degree of transparency, safety and contribute to more economical operational electricity grid preferences.

One of the areas in power systems domain that gained huge attention in recent years is deep neural network (DNN) models 10, 11, 12, 13, 14, 15, 16. Also, DNNs have gained a considerable attention in other well-known applications 17, 18, 19, 20, 21. Convolutional Neural network (CNN) as a promising technique for the time-series data analysis has been successfully used to predict solar irradiance 22. In a study 23, CNN and long short-term memory (LSTM) is used for solar irradiance forecasting. In this study, a model is applied to decompose the solar dataset in different climate regions and they have achieved outperforming performance compared to other state-of-the-art algorithms. Moreover, in 24, a CNN algorithm is developed for forecasting 5 to 20 minutes ahead of solar irradiance by a dataset of collected sky images. By assessing the efficiency of utilized CNN with other compared forecasting models, the experimental results show that the proposed CNN algorithm is more effective for short-term solar irradiance forecasting. Although the performance of LSTM networks in resolving different problems is promising, training these networks like

other neural networks 25, 26 needs to set their hyperparameters that determine many aspects of algorithm behavior 27. Most of the studies relied on manual hyperparameter optimization which needs human experts and it is time consuming 28, 29, 30, 31. In other words, manual hyperparameter optimization is based on the trial and error where human experts interpret how the hyperparameters affect the performance of the model. To overcome this issue, evolutionary algorithms can be employed to automatically find the optimal values of the hyperparameters 32, 33, 34, 35.

Recently, reinforcement learning (RL) has been utilized to conduct wide research in developing time series-based forecasting models 36, 37, 38. In 39, the forecasting model is constructed using three different deep neural networks, and the forecasting outputs for each sub-series are calculated separately. Then, these three deep neural networks are combined using the reinforcement learning technique. In comparison to nineteen state-of-the-art algorithms, the proposed ensemble deep reinforcement learning engine can achieve precise findings in all situations and offers the highest accuracy for wind speed forecasting problem. In another work presented by 40, for the purpose of predicting short-term load, an unique asynchronous deep reinforcement learning model is offered. The temporal correlation of various samples is first disrupted using an innovative asynchronous deep deterministic policy gradient technique, which lowers the agent's overstatement of such total expected discount reward. The convergence of model training is further stabilised by a novel reward incentive mechanism that considers the trend of agent activities at various time steps. According to the experimental findings, the proposed model outperforms eleven benchmark approaches in terms of predicting accuracy, time expenditure, and convergence stability. Besides, the authors in 41 used RL for solving a wind speed forecasting problem. In order to accomplish dynamic selection, the non-dominated combined weighting alternatives are integrated into a deep reinforcement learning environment. It is feasible to contin-

uously generate non-dominated solutions per each forecast in accordance with the time-varying features of wind speed by appropriately constructing the reinforcement learning environment. The outcomes demonstrate the competitiveness of the proposed dynamic ensemble model for wind speed prediction and also it greatly excels six ensemble approaches and five conventional intelligence forecasting techniques. In the work presented by 42, the authors employed the forecasting model pool offered by Q-learning to construct a powerful and innovative hybrid framework for online model selection. The proposed framework is the first framework proposed for the RL approach to dynamically select the optimal forecasting model online. To increase accuracy, a Q-learning agent is constructed that dynamically chooses the optimum forecasting model at each time-step. The real-time wind speed datasets are used in two experiments. According to experimental findings, the proposed algorithm outperformed benchmark methods in both case studies by 47% and 48%.

In this paper, a novel ensemble solar irradiance forecasting model is introduced based on the integration of deep LSTM-CNN neural networks and reinforcement learning (RL). Specifically, at first, we utilize the partial mutual information strategy to extract a set of features from the original input data leading to an enhancement in the quality of input vectors for the deep LSTM-CNN models. Then, the extracted features are used as the inputs of deep LSTM-CNN models to predict solar time-series data. To obtain more accurate predictions, we develop a novel evolutionary algorithm based on the whale optimization algorithm to optimize the hyperparameters of deep LSTM-CNN models. To this end, two effective evolutionary operators are incorporated into the search process of the original version of whale optimization algorithm to speed-up the search process and escaping from the local optima. The aim of the proposed evolutionary algorithm is to automatically obtain the optimal values of hyperparameters of deep LSTM-CNN models. After performing the proposed evolutionary

algorithm in order to obtain the optimal LSTM-CNN architectures, the forecasting results of all these optimized models are integrated into an ensemble strategy based on deep reinforcement learning algorithm to output the best forecasting GHI performance. The experimental results on two real-world datasets confirm that the proposed forecasting model exhibits the best performance among other state of the art algorithms for solar GHI prediction.

The rest of the paper is organized as follows: in Section II, the proposed forecasting model is represented in details. The description of solar irradiance datasets and the initialized setting for the proposed model is presented in Section III. Section IV denotes to the discussion of the obtained results from the proposed model and finally, the paper is concluded in Section V.

## 2. Methodology

In this section, we introduce our novel solar irradiance forecasting model, which is based on three main steps: (1) partial mutual information strategy, (2) hyperparameter optimization, and (3) ensemble strategy. The details of these steps will be provided in the following.

### *2.1. Partial Mutual Information Strategy*

We first utilize the partial mutual information (PMI) strategy as a powerful input variable selection for the deep LSTM–CNN models. The aim of using PMI strategy is to obtain an appropriate set of input data to be utilized in training deep LSTM-CNN models. Therefore, instead of using the original input data, we use an improved set of inputs leading to enhance the performance of deep LSTM-CNN models in forecasting solar irradiance time-series. PMI approach is fundamentally similar to the partial correlation-based model. However it incorporates mutual information

instead of sequential relation to select the input data 43. The value of PMI is a closely related entropy between the output $Y$ and the candidate $C_j$, which is, therefore, not already in $S$, represented by $MI\left(C_j : Y \mid S\right)$. Conditional expectation of $x$ given $S$, is provided by the following formula as the non-parametric regression kernel approximation:

$$E[x \mid S = s] = \frac{1}{n} \frac{\sum_{i=1}^{n} x_i K_h\left(s - s_i\right)}{\sum_{i=1}^{n} K_h\left(s - s_i\right)} \tag{1}$$

where the selected input set is denoted by $S$, $n$ represents the total number of samples, $x$ represents the $c_y$ or $y$, and the Gaussian kernel function ($K_{\mathrm{h}}$) is expressed by:

$$K_h\left(x - x_1\right) = \frac{1}{(\sqrt{2\pi h})^d \sqrt{|\sigma|}} \exp\left(-\frac{\left(x - x_i\right)^T \sigma^{-1}\left(x - x_i\right)}{2h^2}\right) \tag{2}$$

where $\sigma$ represents a matrix of covariance sample, and the dimensionality of $x$ is denoted by $d$. The kernel bandwith which is represented by $h$ is described by the following formula:

$$h = \left(\frac{4}{d+2}\right)^{\frac{1}{d+4}} n^{\frac{-1}{d+4}} \tag{3}$$

Therefore, the PMI value is computed by the following formula:

$$PMI\left(C_j; Y \mid S\right) = MI(u; v) \approx \frac{1}{n} \sum_{i=1}^{n} \log_e \left[\frac{f(u, v)}{f(u) f(v)}\right] \tag{4}$$

where $u = Y - Y(S)$ and $v = C_j - \widehat{C}_j(S)$ in which $S$ shows a set of selected inputs and $j$ represents a set of candidate inputs by using the estimators of the non-parametric kernel indicated by $C_1(S) = E\left[c_j \mid S = s\right]$ and $\varphi(S) = E[y \mid S = s]$. It should be noted that variable $\varphi$ denotes to the sample covariance matrix, f(u) and f(v), and f(u, v) represent probability density functions of u, v, and joint u and v, respectively, and the variables of $\widehat{C}_j$ and $\hat{Y}$ represent the linear least squares regression estimates for the $C_j$ and $Y$.

### *2.2. Hyperparameter Optimization*

The second step optimizes the hyperparameters of the base LSTM–CNN models using a novel optimization approach, which is based on the the-

orem of evolutionary computation. To this end, we introduce our novel evolutionary algorithm, which is used to optimize the hyperparameters of deep LSTM-CNN models. LSTM architecture encompasses specific units, i.e., memory cells, in spot of conventional neurons. Furthermore, LSTM employs a gate pathway that encompasses input, forget, and output gates, giving the LSTM the ability to update and regulate the information transmission of data. LSTM helps to overcome the downsides of recurrent neural networks (RNNs) through reducing the possibility of gradient vanishing and effectively carries long-term interconnections in data with the assistance of intrinsic memory cells and gate pathway 44. On the other hand, convolutional neural network (CNN) is a type of deep neural networks that processes grid-based structured data 45. Despite the fact that CNNs have been effectively used in a wide range of feasible application domains, quite few researchers have described CNN for solar irradiance forecasting problems. To improve the capabilities in designing complex information, the CNN algorithm encompasses three mapping layers: convolutional layer, pooling layer, and fully-connected layer. Every convolutional layer is designed to retrieve patterns from the input variables (i.e., GHI) as well as its associated input data (i.e., historical GHI values).

In the proposed method, we utilize the competitive advantages among both LSTM and CNN models to establish an innovative method to forecast solar irradiance more quickly and precisely. By coordinating memory cells which can be used for updating the hidden layer situation, LSTM practises the features in historical solar irradiance dataset and conserves the long-term interconnections to comprehend the relationships among the features. The features derived by LSTM from solar irradiance time-series are then transmitted to the CNN model's input nodes. Furthermore, the CNN provides structural information about the target and its neighbours' locations. CNN's convolutional layer uses the convolution function on time-series data of target object and neighbour placement to retrieve

underlying features from GHI variable. Then, a fully connected such as dense layer is employed to integrate and forecast the solar irradiance variable (i.e., GHI) depending on the retrieved features. The dropout layer is also utilized in the framework to avoid over-fitting problem.

In order to optimize the hyperparameters of deep LSTM-CNN models, we develop an evolutionary approach based on whale optimization algorithm (WOA). This algorithm has shown excellent performance in numerous engineering applications 46. However, to further improve its capabilities in optimizing the hyperparameters of deep LSTM-CNN models, we incorporate two effective evolutionary strategies termed as the chaotic map (CM) 47 and the opposition-based learning (OBL) 48 into the search process of WOA algorithm. The following formulas are presented throughout the optimization to mathematically model the surrounding phenomenon.

$$\overrightarrow{D} = \left| \overrightarrow{C} \cdot \overrightarrow{X^*}(t) - \overrightarrow{X}(t) \right| \tag{5}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X^*}(t) - \overrightarrow{A}.\overrightarrow{D} \tag{6}$$

where the current iteration is represented by $t$, $\overrightarrow{A}$ and $\overrightarrow{C}$ denote to coefficient vectors, $X(t)$ denotes to the position vector (a random whale) and $X^*$ represents the optimal solution position vector that has been so far achieved. The $\overrightarrow{A}$ and $\overrightarrow{C}$ coefficient vectors are determined according to:

$$\overrightarrow{A} = 2\overrightarrow{a} \cdot \overrightarrow{r} - \overrightarrow{a} \tag{7}$$

$$\overrightarrow{C} = 2 \cdot \overrightarrow{r} \tag{8}$$

where $\overrightarrow{a}$ is declined linearly from 2 to 0 during iterations, and $r$ represents a random vector in the interval of [0, 1]. The procedure of updating position of each search agents based on the spiral (to simulate bubble-net attacking mechanism of humpback whales) is given numerically as follows:

$$\overrightarrow{D'} = \left| \overrightarrow{X^*}(t) - \overrightarrow{X}(t) \right| \tag{9}$$

$$\overrightarrow{X'}(t+1) = \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) \tag{10}$$

where $\overrightarrow{D'}$ determines the difference to the best solution from the $i$th search agent, $b$ denotes to a parameter used to describe logarithmic spiral modes and $l$ represents a randomized value of the [-1,1] range. For further simplification, we generally suppose that Eq. (13) or Eq. (17) would update the position of the search agents, each having a 50% probability, that can be given by the following mathematical formula:

$$\overrightarrow{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \overrightarrow{A} \cdot \overrightarrow{D} & if\ p < 0.5 \\ \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) & if\ p > 0.5 \end{cases} \tag{11}$$

where the variable $p$ represents a random value within the interval of [0, 1].

Referring to the aforementioned mathematical formulas, the fundamental version of WOA has insufficient global search capabilities in the beginning and a slow convergence speed in the latter stages. These shortcomings motivated us to further improve the basic version of WOA from two aspects. Firstly, the chaos theory is used to increase the initial population position's efficiency. Furthermore, to counterbalance the exploration and exploitation of WOA, the opposition-based learning technique is deployed. The details of these two modifications are provided in the following.

**First Improvement:**

A chaos model is a system of random behavior created by a stochastic dynamic function, which combines consistency and randomization. While tackling functional operational issues, these qualities might cause the method to readily stray away the local optimal solution, preserving population variety and improving global search efficiency. Known chaotic maps encompass logistic map, tent map, Chebyshev map, and so on. Due to the great efficiency of the tent map, we utilize this strategy in the proposed evolutionary algorithm to initialize the whale population, which can be computed by the following formula:

$$x_{i+1} = \begin{cases} 2 \times x_i, & 0 \leq x_i \leq 1/2 \\ 2 \times (1 - x_i), & 1/2 \leq x_i \leq 1 \end{cases} \tag{12}$$

Let the population size be $N$ and the search dimension be $D$, we can generate the tent map sequence $x_{ij} (i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, D)$. Thus, the initial population $P_0 = \{X_{ij}\}$ is obtained by mapping it into the search space as follows:

$$X_{ij} = x_{ij} \times (X_{\max j} - X_{\min j}) + X_{\min j}. \tag{13}$$

where $X_{\max j}$ and $X_{\min j}$ are the maximum and minimum of the $j$ dimension in the search space, respectively.

**Second Improvement:**

During the WOA optimization procedure, its population tends to reach the optimal solution, resulting in a decrease in population variety. As a result, we advocate that the opposition-based learning (OBL) technique be used per iteration in order to continually update the individual position. It is utilized to compute the opposing solutions of search agents at the conclusion of each iteration, in order to not only enhance the richness and variety of the population, but also raise the possibility of looking for global optimal solutions. The OBL strategy works as follows:

In the first step, the generated population for all iterations is considered as $P = \{X_{ij}\}, i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, D;$. Then, the opposite population $P' = \left\{X'_{ij}\right\}$ is formed according to the given population computed by $X'_{ij} = X_{\max j} + X_{\min j} - X_{ij}$ where $X_{\max_j}$ and $X_{\min_j}$ respectively represent the maximum and minimum values within the search space. The populations $P$ and $P'$ are combined and organized in increasing order according to the fitness value. The first $N$ search agents having the highest fitness value are chosen as the next population.

We show the flowchart of the proposed modified WOA (MWOA) in Fig. 1. Now, it is the time to optimize the hyperparameters of LSTM–CNN models using the MWOA strategy. To this end, eight important LSTM–CNN hyperparameters as tabulated in Table 1, which have the critical role in the architectural design of LSTM–CNN models are optimized by the MWOA algorithm. Thus, in the solution space of the MWOA, each solution is interpreted as an eight dimensional vector corresponded to the eight hyperparameters. The continuous variables in the LSTM–CNN hyperparameters are transferred as $D = [Hyp_1, Hyp_2, ..., Hyp_n]$ into a discrete search space. The following formulas are taken into account for formulating the discretization model:

$$\Lambda = 1 + n \times K \tag{14}$$

$$\omega = min(\lfloor \Lambda \rfloor, n) \tag{15}$$

where $K$ is a continuous variable in the $[0, 1]$ exploration range for the search space, $\Lambda$ is a mapping operator of $K$ to $[1, n + 1]$ and $\omega$ represents another $\Lambda$ mapping operator to $[1, 2, 3, ..., n]$ interval. Any integer value that belongs to the continuous dimension of the solution can thus be determined using the following equation:

$$X_{ij} = H_\omega \tag{16}$$

where $X_{ij}$, $i = 1..n$ and $j = 1..8$ represents an 8-dimensional vector standing for the $i^{th}$ solution encoding the eight LSTM–CNN hyperparameters and $H_\omega$ denotes to these hyperparameters mapped from discretization manner. Based on the obtained values of hyperparameters, we consider a fitness function to test the efficiency of the configured LSTM–CNN architectures for GHI forecasting. In this regard, we take into consideration the mean square error (MSE) to calculate the fitness value of each MWOA solutions given by the following equation:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - y_i')^2 \tag{17}$$

where $y_i$ and $y'_i$ are the actual and forecasted GHI values by the LSTM–CNN model. The aim of the proposed method is therefore to elicit the lowest $MSE$ value solution containing the optimal values of the LSTM–CNN hyperparameters. This means that an LSTM–CNN model has been achieved with the best performance in the GHI test set.

### *2.3. Ensemble Strategy*

After obtaining the best LSTM–CNN models based on the optimal hyperparameters, the reinforcement learning strategy is adopted in the third step to achieve an optimum subset of these models which can be deployed in the ensemble strategy as the final selected base models. Reinforcement learning is one of the powerful machine learning techniques which is extensively utilized in different optimization problems. The aim of using this technique is to obtain an optimal subset of the optimized deep LSTM-CNN models to make the ensemble forecasting model. There are many methods such as linear regression and linear combination of various models to be used as the optimization strategy in our model instead of reinforcement learning. However, it should be noted that the optimization techniques that are based on linear regression mainly suffer from falling into local optima. Whilst, in reinforcement learning, we use a discrete search space instead of a continuous one leading to better exploration and exploitation of the searching phase. Therefore, using reinforcement learning can reduce the probability of falling into local optima resulting in an improvement in the performance of the proposed ensemble forecasting model. The Q-learning method is among the most common reinforcement learning algorithms based on Q values being updated in the environment. We utilize this approach in the proposed method because of its flexibility and effective convergence. In other words, we employ Q-learning to achieve an optimal subset of the optimized LSTM–CNN models in order to enhance the accuracy of the proposed ensemble model. To this end, we apply the

bagging mechanism to create a collection of optimized LSTM–CNN regression models ($Bagg_M = \{b_1, b_2, \ldots, b_M\}$) where $M$ denotes to number of optimized LSTM–CNN models and the principle is to use reinforcement training strategy to choose an optimal subgroup of the optimized models called $Bagg'_{M'}$.

We require to specify the states set $S$ and the actions set $A$ to undertake reinforcement learning strategy. Each state in the proposed model is represented by a $s_t = [L_t, MSE_t]$ tuple in which $L_t$ refers to a vector containing $M$ elements with the values 0 or 1 where 0 means that the corresponding LSTM-CNN model is not selected in the ensemble model while 1 means that the corresponding LSTM-CNN model is selected in the ensemble model. Moreover, $MSE_t$ belongs to the error metric of the ensemble regression model built by the selected optimized LSTM–CNN regression models. In the proposed method, we use $1 - MSE_t$ as the reward obtained for a pair of state and action. Each action is represented by $a_t = i, i = 1, 2, ..., M$ referring to the index of the optimized LSTM-CNN model which will be added/removed to/from the subset of selected models. In other words, for $a_t = i$, if the value of $ith$ element in the $L_t$ vector is 0 demonstrating that the $ith$ optimized model is not selected, then its value will be changed to 1 demonstrating that the $ith$ optimized model is selected, and vice versa. The average of the outputs of the selected optimized LSTM–CNN models are calculated to measure the output performance of the ensemble model. The proposed method is designed to ensure that an ensemble regression model with minimum error is obtained using the Q-Learning algorithm by choosing an efficient subgroup of the optimized LSTM–CNN models.

The reinforcement learning algorithm maps the state to an action described as the $\Pi : S \rightarrow A$ policy 49, 50. The Q-Learning approach is founded on a function-value named as $Q^{\Pi}(s_t, a_t)$, where the states and actions are respectively represented by $s_t$ and $a_t$. The overall estimated

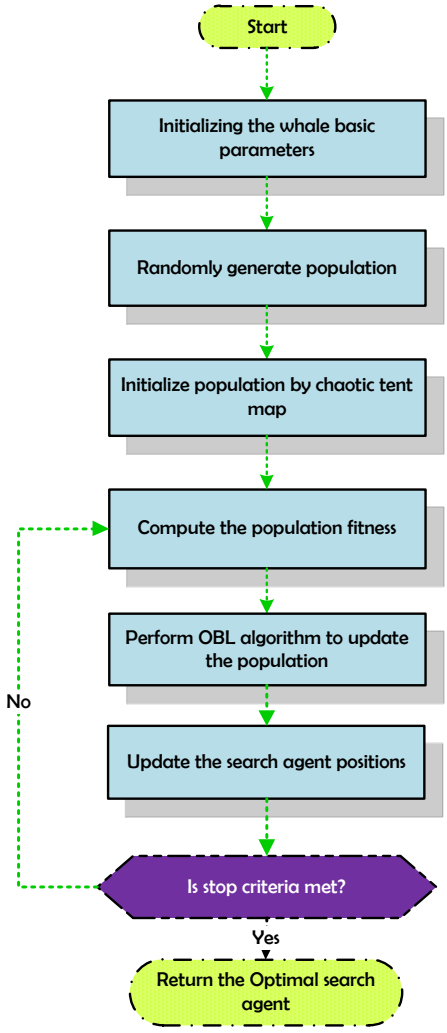


Figure 1: The flowchart for the proposed MWOA.

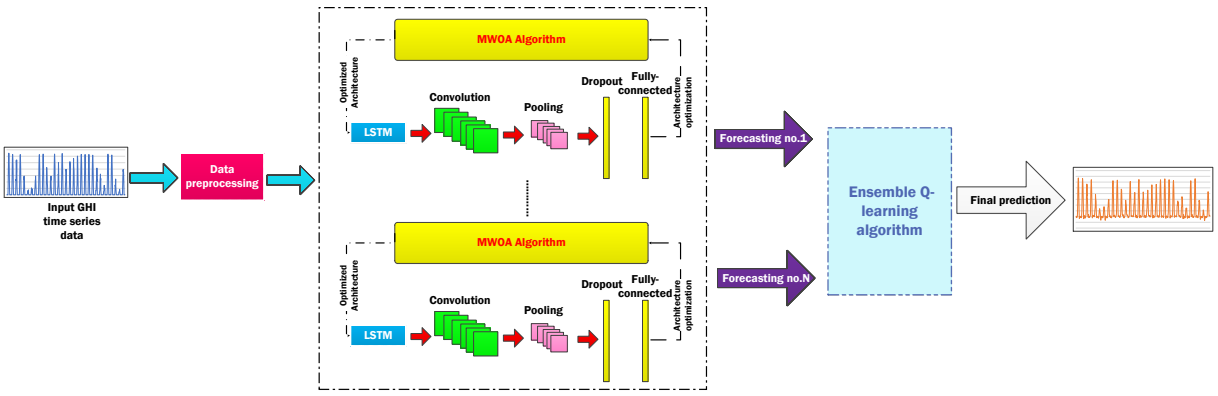


Figure 2: An overall schema of the novel HYDEREL framework.

discount reward value depending on the optimal Π strategy is defined by the following equation:

$$Q^{\pi}(\mathbf{s}, a) = E\left(\sum_{k=0}^{\infty} \gamma^{i} r_{t} \mid \mathbf{s}_{0} = \mathbf{s}, a_{0} = a, \pi\right) \tag{18}$$

where $\gamma$ corresponds to the discount coefficient of $0 \leq \gamma \leq 1$. Using the Q-learning algorithm, the optimized action-value function $Q^{*}(s, a)$ can be quantified by:

$$Q^{*}(\mathrm{s}, a) = E\left(r_{t+1} + \gamma \max_{a'} Q^{*}\left(\mathrm{s}_{t+1}, a'\right) \mid \mathrm{s}_{t} = \mathrm{s}, a_{t} = a\right) \tag{19}$$

The technique proceeds with a series of episodes in which the action value function $Q$ is updated based on the following formula to determine an optimal search strategy:

$$\begin{aligned} Q\left(\mathbf{s}_{t}, a_{t}\right) \leftarrow Q\left(\mathbf{s}_{t}, a_{t}\right) + \alpha\left[r_{t+1} + \lambda \max_{a} Q\left(\mathbf{s}_{t+1}, a\right)\right. \\ \left. - Q\left(\mathbf{s}_{t}, a_{t}\right)\right] \end{aligned} \tag{20}$$

The desirable subgroup of the optimized LSTM–CNN regression models $Bagg'_{L'}$ can be defined based on the best result achieved, after the Q-learning technique has been executed. This optimum subset is subsequently applied to yield the proposed ensemble GHI time series forecasting strategy as a final frame for the optimized LSTM–CNN models. We show

the entire framework of our optimized deep RL ensemble model briefly named as HYDEREL in Fig. 2 and from technical point of view in Algorithm 1.

## 3. Experimental setups and solar irradiance datasets

In order to execute the proposed HYDEREL framework, we program it with python using the powerful deep learning libraries including Keras and TensorFlow. A machine with one 16 GB RAM, one GPU of GeForce GTX 1080 Ti and the Ubuntu operating system is utilized to conduct the experiments. Before conducting the experiments, we need to set the input parameters based on a strategy. To this end, we use a greedy search procedure to obtain the values of the input parameters through trial and error. Accordingly, for configuring the proposed MWOA algorithm, we choose the number of population to 20 and maximum iteration number to 30. It should be mentioned that the proposed HYDEREL model and the other competitor models are executed 10 times and the average of results is reported. Furthermore, the LSTM–CNN hyperparameters that are evolved with MWOA are described in Table 1. We optimize eight hyperparameters during the constructing of the optimal LSTM–CNN architectures including number of filters ($N_f$), kernel size ($K_s$), maxpooling size ($MP_s$), batch size ($B_s$), number of convolutional layers ($N_c$), number of epochs ($N_e$), dropout rate ($D_r$), and learning rate ($L_r$). Moreover, we choose powerful Adam optimizer and Relu as the activation function during the training process. For RL ensemble configurations, we set the number of episodes in the Q-learning algorithm to 200 and consider 15 base optimized LSTM–CNN regression models for selecting an optimal subset of regression models with deep RL ensemble strategy.

In order to show the competitiveness of our proposed HYDEREL framework, we compare it with seven hybrid powerful state of the art deep learning models including adaptive hybrid model (AHM) 51, hybrid feature se-

**Algorithm 1** The pseudo-code of the proposed deep HYDEREL model for solar GHI forecasting.

**Input**: $N$ (Population size), $GEN$ (Maximum number of generations), and $L$ (Number of base regression models).
**Output**: Predicted GHI values.
**Begin algorithm**:
Split GHI dataset into training $Tr$ and testing $Te$ sets;
Generate a set bag of base regression models $Bagg_M = \{b_1, b_2, \ldots, b_M\}$;
Set $m = 1$;
**while** ($m < L$) **do**
    Generate a random initial population $X_i$ ($i$=1,2,..., $N$) by chaotic tent map;
    Set $g$=1;
    **while** ($g < GEN$) **do**
        Set a LSTM–CNN model for each solution based on their hyperparameter values;
        Calculate the population fitness as the MSE of LSTM–CNN algorithm obtained by $Tr$ set;
        **for** each search agent **do**
            Update $a$, $A$, $C$ and $p$;
            **if** ($|A|$<1) **then**
                Update the position of search agent using Eq.(5);
            **else if** ($|A|\geq$1) **then**
                Update the position of the search agent by the Eq.(11);
            **end if**
        **end for**
        Perform OBL strategy;
        Calculate the fitness of all search agents and update $X^*$ if a better solution is found.
        Set $g$=$g$+1;
    **end while**
    Consider the $b_m$ LSTM–CNN regression model with hyperparameters obtained by the best search agent;
    Set $m$=$m$+1;
**end while**
Perform deep Q-learning model over the LSTM–CNN regression model $Bagg_M$ in order to chose an optimal subset of the regression models indicated by $Bagg'_{M'}$;
Apply the ensemble strategy to forecast the GHI points in the test set $Te$ using the selected regression models $Bagg'_{M'}$;
Return the predicted GHI as the output;
**End algorithm**

Table 1: The symbols and corresponding values of the LSTM–CNN hyperparameters optimized by the proposed MWOA model.

| Symbol | Value |
|---|---|
| $N_f$ | [1, 600] |
| $K_s$ | [1, 30] |
| $N_e$ | [1, 400] |
| $N_c$ | [1, 2,..., 24] |
| $MP_s$ | [1, 40] |
| $B_s$ | [10, 20,..., 350] |
| $D_r$ | [0.2, 0.25,..., 0.65] |
| $L_r$ | [0.001, 0.006,.., 0.1] |

lection method (HFS) 52, XGboost, Outlier-robust hybrid model (ORHM) 53, novel hybrid deep neural network model (NHDNNM) 54, OHS-LSTM 55, and CNN-LSTM that have shown their strength in time-series forecasting problems. Also, in order to demonstrate the search capabilities of our proposed HYDEREL algorithm, we use powerful evolutionary algorithms such as genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization algorithm (ACO), biogeography-based optimization (BBO) and whale optimization algorithm (WOA) on the framework proposed in this work. These models refer to cases in which the proposed MWOA algorithm is replaced by another optimization method (GA, PSO, etc.) in the HYDEREL framework. Also, MWOA-LSTM-CNN has been considered in order to show the role of proposed evolutionary approach without using the element of ensemble-based reinforcement learning algorithm.

We use National Renewable Energy Laboratory 56 solar irradiance datasets collected from two solar stations located in Los Angeles (LA) and Phoenix in the western side of the United states. The general overview of these two stations are provided in Figs. 3 and 4 for Los Angeles and

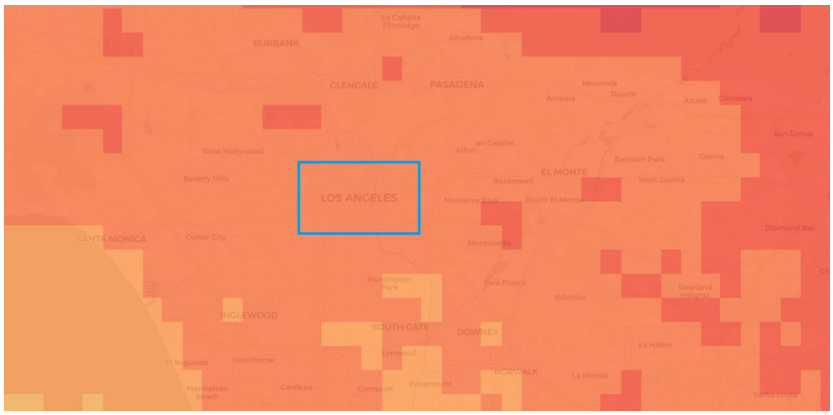


Figure 3: An overview of the LA-based solar stations.

Phoenix, respectively. Each of these two solar stations contain 8760 GHI time series data points in one-hour intervals for the year 2018 in which the GHI values are normalised. We consider 75% of each dataset as the training set and the remaining 25% is allocated to test set. We should note that the 25/75% split of the training and test sets is not random, thus that both sets are made of distinct days. Based on these two sets of solar irradiance data, GHI has an increase from 8:00 to 13:00, and then, has a decrease until it meets zero from about 18:00 to 20:00. For selecting the input features of deep LSTM–CNN models, we utilize the partial mutual information strategy (PMI). The PMI values considered above of the threshold equal to $\Xi = 0.4$ is chosen which results in 57 input GHI features for training of the deep learning models. The root mean square error (RMSE) and mean absolute error (MAE) which are widely used evaluation metrics in the literature are considered to calculate the performance of the forecasting models.

## 4. Experimental Results

In this section, we test the forecasting performance of our proposed algorithm compared to other powerful benchmark algorithms. Tables 2 and

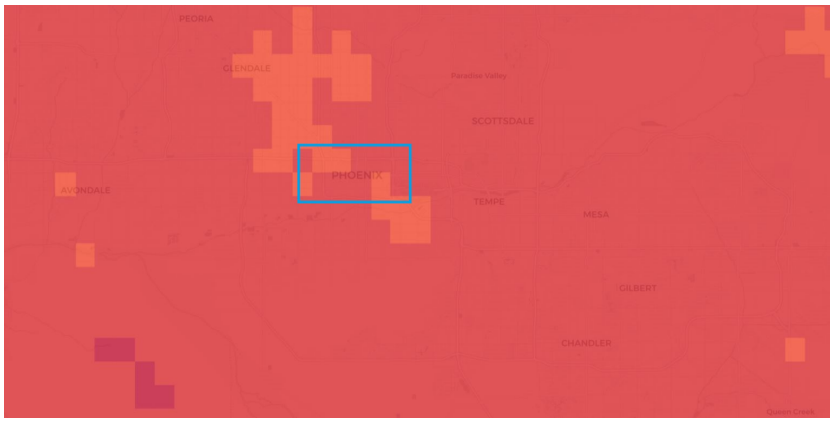


Figure 4: An overview of the Phoenix-based solar stations.

3 show that in all testing GHI data sets, the RMSE and MAE of our proposed model are smaller than all other twelve benchmark models. For instance, we notice that the RMSE of 1-step ahead prediction in Phoenix station for the proposed HYDEREL model equal to 0.034208 significantly outperforms in comparison with HFS as the closest competitor algorithm equal to 0.034436, while the MAE of the proposed model equal to 0.015111 significantly outperforms the GA-RL-Ens model equal to 0.015411 as the most compatible model to HYDEREL. Furthermore, for the LA test dataset, the proposed HYDEREL method with a RMSE value of 0.033451 and a MAE value of 0.016792 has the lowest values among all the compared powerful competitive methods. It can be noted that in comparison with AHM, HFS, ORHM, NHDNNM and HOS-LSTM as the state of the art models for time series energy forecasting problems, we find that the performance of our proposed HYDEREL model is much more reliable than these recent powerful models. Besides, the predicted performance of the proposed model is still good compared to other hybrid evolutionary-RL ensemble models. These results indicate that the HYDEREL algorithm can capture complex solar GHI features compared to other hybrid deep LSTM–CNN models optimized by RL ensemble models.

Table 2: The performance results of RMSE and MAE metrics for Phoenix dataset.

| Model | 1-step | | 2-step | | 12-step | | 24-step | |
|---|---|---|---|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| AHM | 0.037729 | 0.020656 | 0.049905 | 0.024177 | 0.074583 | 0.034471 | 0.093377 | 0.045822 |
| HFS | 0.034436 | 0.016075 | 0.051074 | 0.024943 | 0.078714 | 0.034137 | 0.095067 | 0.044216 |
| XGboost | 0.036247 | 0.019855 | 0.051963 | 0.024639 | 0.072804 | 0.033414 | 0.091782 | 0.043693 |
| ORHM | 0.035758 | 0.018975 | 0.053833 | 0.026384 | 0.073988 | 0.032091 | 0.094503 | 0.043452 |
| NHDNNM | 0.035562 | 0.015896 | 0.050868 | 0.023332 | 0.077522 | 0.032355 | 0.095013 | 0.042383 |
| OHS-LSTM | 0.036146 | 0.019742 | 0.051916 | 0.024527 | 0.072677 | 0.033397 | 0.091561 | 0.043514 |
| LSTM-CNN | 0.037826 | 0.021139 | 0.050562 | 0.024913 | 0.075143 | 0.034891 | 0.093662 | 0.046091 |
| MWOA-LSTM-CNN | 0.036112 | 0.019422 | 0.051222 | 0.023922 | 0.072564 | 0.032818 | 0.091142 | 0.042989 |
| GA-RL-Ens | 0.034677 | 0.015411 | 0.051319 | 0.024451 | 0.074832 | 0.033511 | 0.094008 | 0.044192 |
| PSO-RL-Ens | 0.035192 | 0.016511 | 0.051922 | 0.023396 | 0.079166 | 0.035495 | 0.092381 | 0.045613 |
| ACO-RL-Ens | 0.039802 | 0.019389 | 0.060149 | 0.034888 | 0.079961 | 0.033995 | 0.095894 | 0.043617 |
| BBO-RL-Ens | 0.038226 | 0.020389 | 0.049823 | 0.023615 | 0.071998 | 0.032598 | 0.094142 | 0.042466 |
| WOA-RL-Ens | 0.034988 | 0.016503 | 0.053221 | 0.025096 | 0.083541 | 0.039562 | 0.096508 | 0.048115 |
| HYDEREL | **0.034208** | **0.015111** | **0.049533** | **0.022997** | **0.070286** | **0.031914** | **0.090416** | **0.041516** |

Table 3: The performance results of RMSE and MAE metrics for Los Angeles dataset.

| Model | 1-step | | 2-step | | 12-step | | 24-step | |
|---|---|---|---|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| AHM | 0.034389 | 0.018903 | 0.053352 | 0.033071 | 0.067799 | 0.036235 | 0.082938 | 0.040784 |
| HFS | 0.034481 | 0.017176 | 0.047231 | 0.029983 | 0.067668 | 0.041654 | 0.083894 | 0.046652 |
| XGboost | 0.034195 | 0.018007 | 0.048122 | 0.028775 | 0.067116 | 0.034133 | 0.082593 | 0.040888 |
| ORHM | 0.034641 | 0.017746 | 0.045537 | 0.024858 | 0.069499 | 0.034732 | 0.085318 | 0.038183 |
| NHDNNM | 0.039785 | 0.027732 | 0.046621 | 0.026433 | 0.076965 | 0.042568 | 0.092854 | 0.047328 |
| OHS-LSTM | 0.034156 | 0.017949 | 0.048055 | 0.028697 | 0.067045 | 0.034067 | 0.082507 | 0.040785 |
| LSTM-CNN | 0.034233 | 0.018032 | 0.048276 | 0.028754 | 0.067113 | 0.034119 | 0.082188 | 0.040712 |
| MWOA-LSTM-CNN | 0.034098 | 0.017988 | 0.048002 | 0.028599 | 0.067088 | 0.033915 | 0.082122 | 0.040615 |
| GA-RL-Ens | 0.033914 | 0.017187 | 0.053509 | 0.034491 | 0.066831 | 0.031911 | 0.081566 | 0.036217 |
| PSO-RL-Ens | 0.036706 | 0.020492 | 0.046581 | 0.024804 | 0.075191 | 0.041089 | 0.089096 | 0.046807 |
| ACO-RL-Ens | 0.036681 | 0.022705 | 0.060691 | 0.041818 | 0.067592 | 0.032981 | 0.082292 | 0.039489 |
| BBO-RL-Ens | 0.034011 | 0.017006 | 0.046575 | 0.023298 | 0.066461 | 0.035466 | 0.085197 | 0.039225 |
| WOA-RL-Ens | 0.034218 | 0.018416 | 0.062488 | 0.038092 | 0.078402 | 0.042139 | 0.092656 | 0.048307 |
| HYDEREL | **0.033451** | **0.016792** | **0.044396** | **0.023139** | **0.066385** | **0.031354** | **0.081216** | **0.035767** |

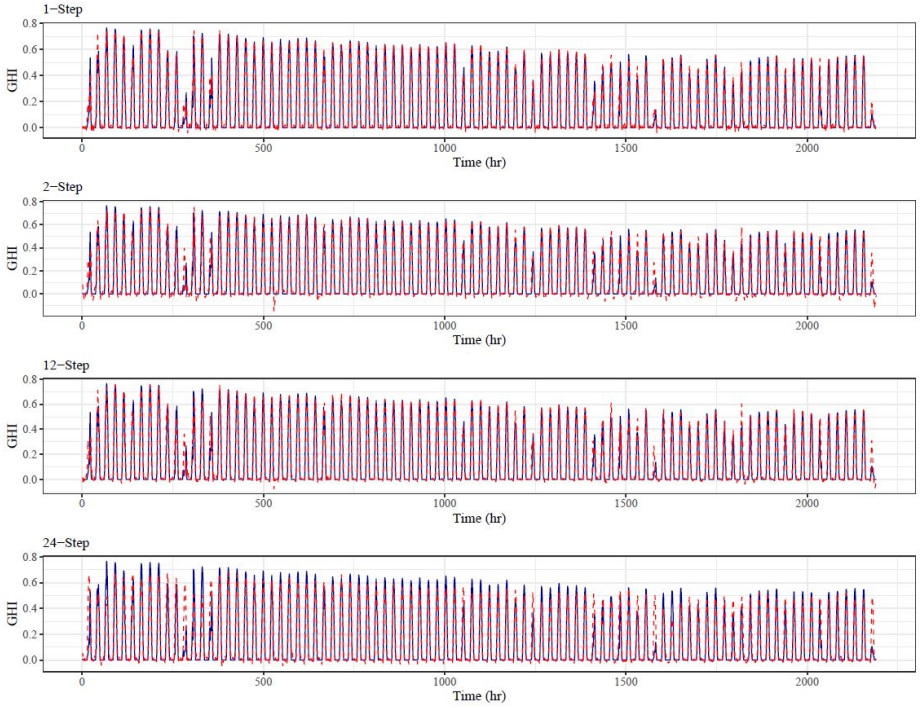


Figure 5: Actual vs predicted points for Phoenix GHI test dataset.

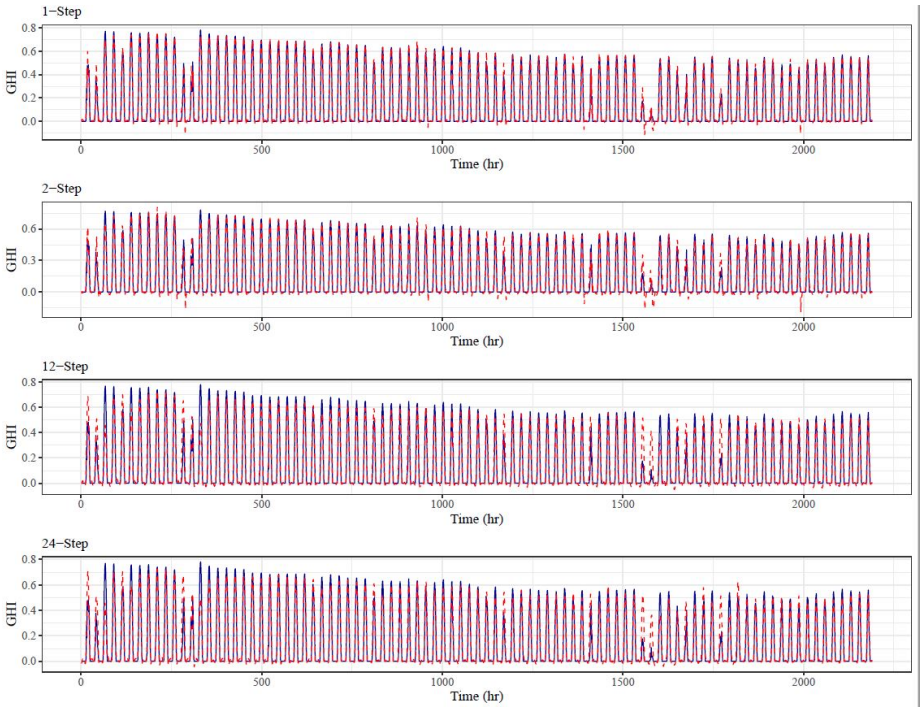


Figure 6: Actual vs predicted points for LA GHI test dataset.

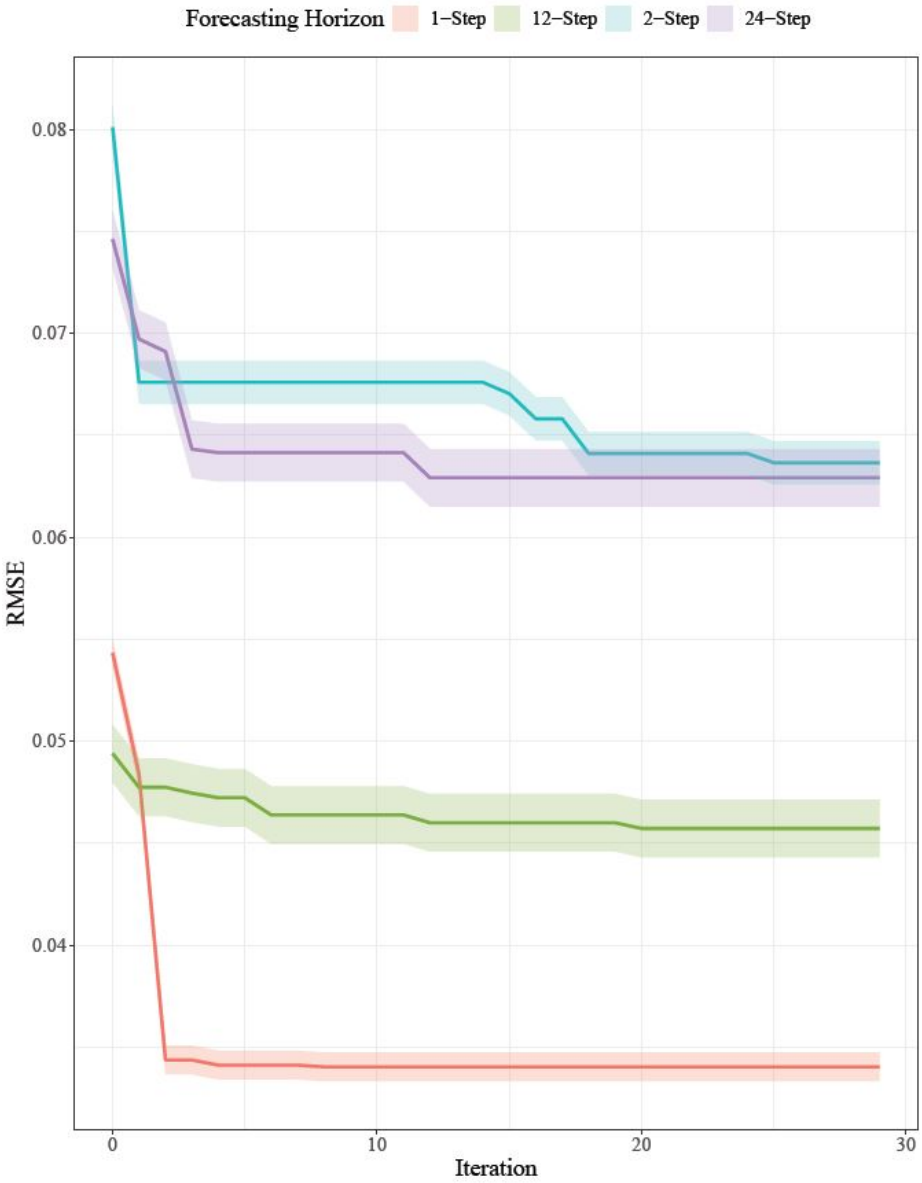


Figure 7: The convergence curves of the proposed HYDEREL model for legend horizons of LA dataset.

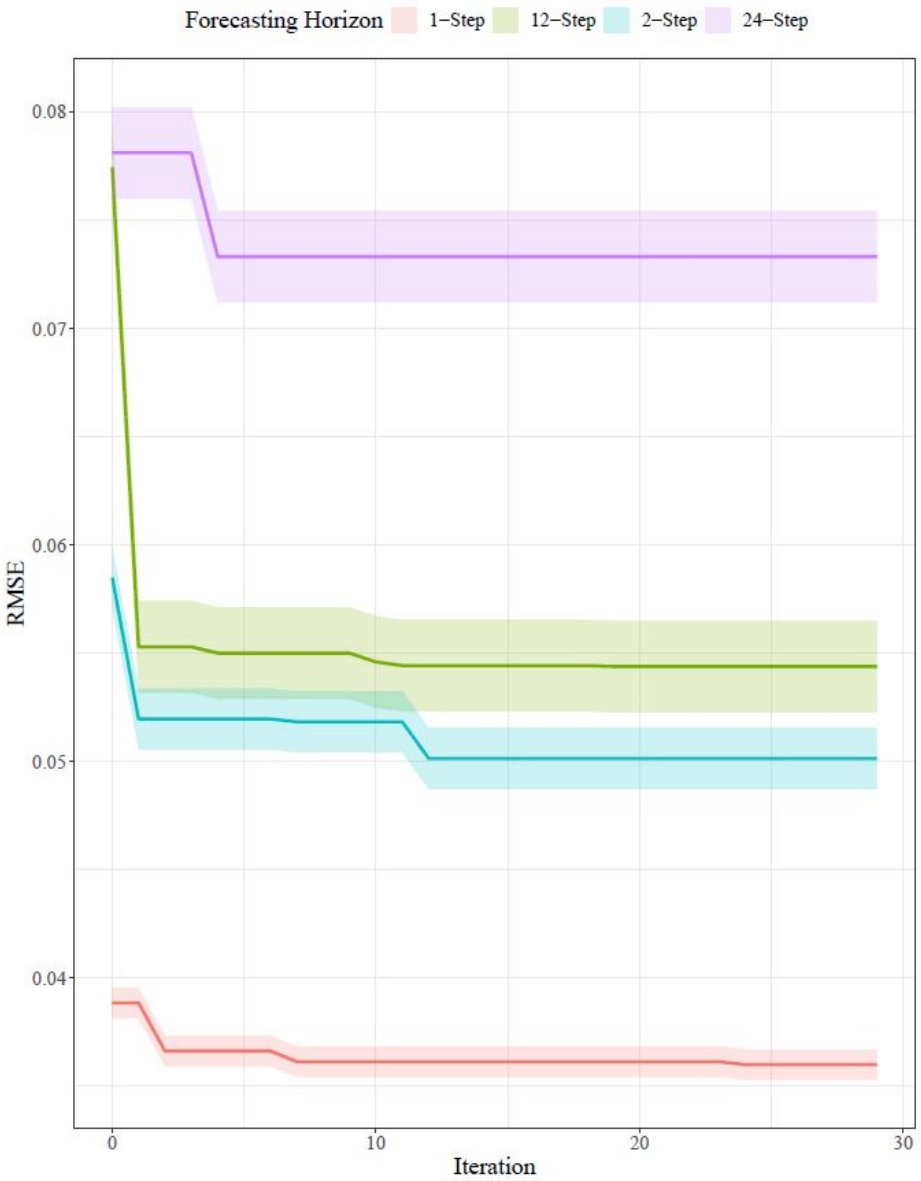


Figure 8: The convergence curves of the proposed HYDEREL model for legend horizons of Phoenix dataset.

If a forecasting algorithm can match the actual and predicted points optimally, it indicates the obvious strength of that method. As can be seen from Figs. 5 and 6, our proposed HYDEREL algorithm matches well the blue dots that represent the actual data points and the red dots that represent the predicted data points for each of the next four different time-steps. The important point should be noticed from Tables 2 and 3, as well as Figs. 5 and 6, which show that as the time-step increases, the prediction made by the our proposed HYDEREL model becomes more and more difficult, and in this case, our algorithm has the least amount of error metric values in comparison to other twelve benchmark models. In Figs. 8 and 7, the diagrams generated by the proposed HYDEREL algorithm using the next four different forecasting time-steps are demonstrated. The obvious highlight from these diagrams refers to the high convergence speed of the HYDEREL model in order to find the optimal solution, which is due to the use of two efficient operators including chaotic circle map and OBL in the proposed MWOA algorithm. In summary, this work demonstrates that by applying two efficient chaotic circle map and OBL operators in the proposed MWOA algorithm, the hyperparameters of the deep LSTM–CNN models are perfectly optimized, and also, the deep RL Q-learning algorithm selects the best subset of optimal solutions for solar irradiance GHI forecasting, which indicates the strength performance of our proposed HYDEREL model.

## 5. Conclusion

In this research work, a hybrid model called HYDEREL is proposed based on three main components including PMI input selection strategy, the optimized deep LSTM–CNN models, and deep RL algorithm for predicting solar irradiance. To validate the efficacy of the hybrid HYDEREL model, the GHI data points obtained from two separate solar stations which are near to the cities of Phoenix and Los Angeles are utilized. Meanwhile, in

different time-step scenarios, several well-known and recently published algorithms are fairly compared with our proposed framework. The experimental findings show that our proposed HYDEREL model can increase the forecasting performance noticeably and indicate that the proposed model is also more robust than other compared benchmark models.

## References


[1] A. Jäger-Waldau, "Pv status report 2011. research, solar cell production and market implementation of photovoltaics. jrc scientific and technical reports, eur 24807 en," *Institute for Energy/Joint Research Centre/European Commission. Office for Official Publications of the European Communities, Luxembourg. Available at http://ie. jrc. ec. europa. eu. Accessed Mar*, 2012.

[2] T. Sharma and P. Balachandra, "Model based approach for planning dynamic integration of renewable energy in a transitioning electricity system," *International Journal of Electrical Power & Energy Systems*, vol. 105, pp. 642–659, 2019.

[3] J. M. Alemany, B. Arendarski, P. Lombardi, and P. Komarnicki, "Accentuating the renewable energy exploitation: Evaluation of flexibility options," *International Journal of Electrical Power & Energy Systems*, vol. 102, pp. 131–151, 2018.

[4] Z. W. Geem and Y. Yoon, "Harmony search optimization of renewable energy charging with energy storage system," *International Journal of Electrical Power & Energy Systems*, vol. 86, pp. 120–126, 2017.

[5] H. Golmohamadi, R. Keypour, B. Bak-Jensen, and J. R. Pillai, "A multi-agent based optimization of residential and industrial demand response aggregators," *International Journal of Electrical Power & Energy Systems*, vol. 107, pp. 472–485, 2019.

[6] S. Rosiek, J. Alonso-Montesinos, and F. Batlles, "Online 3-h forecasting of the power output from a bipv system using satellite observations and ann," *International Journal of Electrical Power & Energy Systems*, vol. 99, pp. 261–272, 2018.

[7] F. Barbieri, S. Rajakaruna, and A. Ghosh, "Very short-term photovoltaic power forecasting with cloud modeling: A review," *Renewable and Sustainable Energy Reviews*, vol. 75, pp. 242–263, 2017.

[8] M. Saffari, M. Khodayar, S. M. J. Jalali, M. Shafie-khah, and J. P. Catalão, "Deep convolutional graph rough variational auto-encoder for short-term photovoltaic power forecasting," in *2021 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, 2021, pp. 1–6.

[9] 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.

[10] S. M. J. Jalali, S. Ahmadian, M. K. Noman, A. Khosravi, S. M. S. Islam, F. Wang, and J. P. Catalão, "Novel uncertainty-aware deep neuroevolution algorithm to quantify tidal forecasting," *IEEE Transactions on Industry Applications*, vol. 58, pp. 3324–3332, 2022.

[11] P. Arora, S. M. J. Jalali, S. Ahmadian, B. K. Panigrahi, P. Suganthan, and A. Khosravi, "Probabilistic wind power forecasting using optimised deep auto-regressive recurrent neural networks," *IEEE Transactions on Industrial Informatics*, vol. 1, pp. 1–10, 2022.

[12] S. M. J. Jalali, S. Ahmadian, M. Khodayar, A. Khosravi, M. Shafie-khah, S. Nahavandi, and J. P. Catalão, "An advanced short-term wind power forecasting framework based on the optimized deep neural network models," *International Journal of Electrical Power  Energy Systems*, vol. 141, p. 108143, 2022.

[13] 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.

[14] 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.

[15] S. M. J. Jalali, M. Khodayar, A. Khosravi, G. J. Osório, S. Nahavandi, and J. P. Catalão, "An advanced generative deep learning framework for probabilistic spatio-temporal wind power forecasting," in *2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)*. IEEE, 2021, pp. 1–6.

[16] M. Khodayar, M. Saffari, M. Williams, and S. M. J. Jalali, "Interval deep learning architecture with rough pattern recognition and fuzzy inference for short-term wind speed forecasting," *Energy*, vol. 254, p. 124143, 2022.

[17] S. Ahmadian, S. M. J. Jalali, S. M. S. Islam, A. Khosravi, E. Fazli, and S. Nahavandi, "A novel deep neuroevolution-based image classification method to diagnose coronavirus disease (covid-19)," *Computers in biology and medicine*, vol. 139, p. 104994, 2021.

[18] S. Ahmadian, M. Ahmadian, and M. Jalili, "A deep learning based trust- and tag-aware recommender system," *Neurocomputing*, vol. 488, pp. 557–571, 2022.

[19] M. Ahmadian, M. Ahmadi, and S. Ahmadian, "A reliable deep representation learning to improve trust-aware recommendation systems," *Expert Systems with Applications*, vol. 197, p. 116697, 2022.

[20] M. Ahmadian, M. Ahmadi, S. Ahmadian, S. M. J. Jalali, A. Khosravi, and S. Nahavandi, "Integration of deep sparse autoencoder and particle swarm optimization to develop a recommender system," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 2524–2530.

[21] A. K. Yengikand, M. Meghdadi, S. Ahmadian, S. M. J. Jalali, A. Khosravi, and S. Nahavandi, "Deep representation learning using multilayer perceptron and stacked autoencoder for recommendation systems," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 2485–2491.

[22] 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. 52, pp. 54–65, 2022.

[23] P. Kumari and D. Toshniwal, "Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting," *Applied Energy*, vol. 295, p. 117061, 2021.

[24] A. Ryu, M. Ito, H. Ishii, and Y. Hayashi, "Preliminary analysis of short-term solar irradiance forecasting by using total-sky imager and convolutional neural network," in *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*. IEEE, 2019, pp. 627–631.

[25] L. Mou, C. Zhou, P. Zhao, B. Nakisa, M. N. Rastgoo, R. Jain, and W. Gao, "Driver stress detection via multimodal fusion using attention-based cnn-lstm," *Expert Systems with Applications*, vol. 173, p. 114693, 2021.

[26] B. Nakisa, M. N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, "Automatic emotion recognition using temporal multimodal deep learning," *IEEE Access*, vol. 8, pp. 225 463–225 474, 2020.

[27] S. M. J. Jalali, S. Ahmadian, A. Khosravi, M. Shafie-khah, S. Nahavandi, and J. P. Catalao, "A novel evolutionary-based deep convolutional neural network model for intelligent load forecasting," *IEEE Transactions on Industrial Informatics*, vol. 17, pp. 8243–8253, 2021.

[28] S. Ahmadian, S. M. J. Jalali, S. Raziani, and A. Chalechale, "An efficient cardiovascular disease detection model based on multilayer perceptron and moth-flame optimization," *Expert Systems*, vol. 39, p. e12914, 2022.

[29] S. M. J. Jalali, M. Ahmadian, S. Ahmadian, R. Hedjam, A. Khosravi, and S. Nahavandi, "X-ray image based covid-19 detection using evolutionary deep learning approach," *Expert Systems with Applications*, vol. 201, p. 116942, 2022.

[30] S. Raziani, S. Ahmadian, S. M. J. Jalali, and A. Chalechale, "An efficient hybrid model based on modified whale optimization algorithm and multilayer perceptron neural network for medical classification problems," *Journal of Bionic Engineering*, vol. 1, pp. 1–18, 2022.

[31] 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.

[32] 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 (ICONIP)*. Springer, 2020, pp. 147–155.

[33] B. Nakisa, M. N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, "Long short term memory hyperparameter optimization for a neural network based emotion recognition framework," *IEEE Access*, vol. 6, pp. 49 325–49 338, 2018.

[34] 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.

[35] S. M. J. Jalali, A. Khosravi, P. M. Kebria, R. Hedjam, and S. Nahavandi, "Autonomous robot navigation system using the evolutionary multi-verse optimizer algorithm," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 1221–1226.

[36] 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.

[37] S. M. J. Jalali, G. J. Osório, S. Ahmadian, M. Lotfi, V. M. Campos, M. Shafie-khah, A. Khosravi, and J. P. Catalão, "New hybrid deep neural architectural search-based ensemble reinforcement learning strategy for wind power forecasting," *IEEE Transactions on Industry Applications*, vol. 58, pp. 15–27, 2021.

[38] S. K. Perepu, B. S. Balaji, H. K. Tanneru, S. Kathari, and V. S. Pinnamaraju, "Reinforcement learning based dynamic weighing of ensemble models for time series forecasting," *arXiv preprint arXiv:2008.08878*, 2020.

[39] H. Liu, C. Yu, H. Wu, Z. Duan, and G. Yan, "A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting," *Energy*, vol. 202, p. 117794, 2020.

[40] W. Zhang, Q. Chen, J. Yan, S. Zhang, and J. Xu, "A novel asynchronous deep reinforcement learning model with adaptive early forecasting method and reward incentive mechanism for short-term load forecasting," *Energy*, vol. 236, p. 121492, 2021.

[41] C. Chen and H. Liu, "Dynamic ensemble wind speed prediction model based on hybrid deep reinforcement learning," *Advanced Engineering Informatics*, vol. 48, p. 101290, 2021.

[42] V. Kosana, K. Teeparthi, S. Madasthu, and S. Kumar, "A novel reinforced online model selection using q-learning technique for wind speed prediction," *Sustainable Energy Technologies and Assessments*, vol. 49, p. 101780, 2022.

[43] E. Snieder, R. Shakir, and U. Khan, "A comprehensive comparison of four input variable selection methods for artificial neural network flow forecasting models," *Journal of Hydrology*, vol. 583, p. 124299, 2020.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] S. Albawi, T. Abed Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*. IEEE, 2017, pp. 1–8.

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

[47] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *Journal of computational design and engineering*, vol. 5, no. 4, pp. 458–472, 2018.

[48] A. H. Gandomi and A. R. Kashani, "Evolutionary bound constraint handling for particle swarm optimization," in *2016 4th International Symposium on Computational and Business Intelligence (ISCBI)*. IEEE, 2016, pp. 148–152.

[49] M. Sun, C. Feng, and J. Zhang, "Multi-distribution ensemble probabilistic wind power forecasting," *Renewable Energy*, vol. 148, pp. 135–149, 2020.

[50] C. Feng, M. Sun, and J. Zhang, "Reinforced deterministic and probabilistic load forecasting via $q$-learning dynamic model selection," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1377–1386, 2019.

[51] J. Zhang, Z. Tan, and Y. Wei, "An adaptive hybrid model for short term electricity price forecasting," *Applied Energy*, vol. 258, p. 114087, 2020.

[52] X. Zhang, J. Wang, and Y. Gao, "A hybrid short-term electricity price forecasting framework: Cuckoo search-based feature selection with singular spectrum analysis and svm," *Energy Economics*, vol. 81, pp. 899–913, 2019.

[53] J. Wang, W. Yang, P. Du, and T. Niu, "Outlier-robust hybrid electricity price forecasting model for electricity market management," *Journal of Cleaner Production*, vol. 249, p. 119318, 2020.

[54] C.-J. Huang, Y. Shen, Y.-H. Chen, and H.-C. Chen, "A novel hybrid deep neural network model for short-term electricity price forecasting," *International Journal of Energy Research*, 2020.

[55] S. Zhou, L. Zhou, M. Mao, H.-M. Tai, and Y. Wan, "An optimized heterogeneous structure lstm network for electricity price forecasting," *IEEE Access*, vol. 7, pp. 108 161–108 173, 2019.

[56] M. Sengupta, Y. Xie, A. Lopez, A. Habte, G. Maclaurin, and J. Shelby, "The national solar radiation data base (nsrdb)," *Renewable and Sustainable Energy Reviews*, vol. 89, pp. 51–60, 2018.