Arabian Journal for Science and Engineering (2023) 48:14823–14836
https://doi.org/10.1007/s13369-023-07822-9

RESEARCH ARTICLE-ELECTRICAL ENGINEERING

Optimized Machine Learning-Based Forecasting Model for Solar Power Generation by Using Crow Search Algorithm and Seagull Optimization Algorithm

Shashikant Kaushaley[1] · **Binod Shaw**[1] · **Jyoti Ranjan Nayak**[2]

Received: 22 August 2022 / Accepted: 27 February 2023 / Published online: 3 May 2023

Abstract

Forecasting Solar Power is an important aspect for power trading companies. It helps in energy bidding, planning, and control. The challenge in forecasting is to predict nonlinear data, which can be fulfilled by the computation technique and machine learning model. ML models have high accuracy for time-series forecasting, but their accuracy is poor for nonlinear forecasting. To enhance the ML model for nonlinear prediction, an optimization algorithm is used for training. This paper presents how the computation technique is incorporated into the machine learning model and compared it with the conventional model. CSA-ANN and SOA-ANN models are developed and forecast solar power for a-day-ahead, three-day-ahead, and a week-ahead solar power generation by considering time, irradiation, and temperature as input parameters for the model. The models are compared with ANN, DE-ANN, and PSO-ANN since these models are widely used. Upon comparison, ANN gives the best result for short-term prediction but is unable to predict midterm and long-term predictions, whereas this problem is overcome by SOA-ANN, which is done by changing its training algorithm, and its performance is measured via statistical parameters such as MAE, MSE, MAPE, and R^2. The percentage improvement of SOA-ANN is obtained with these statistical parameters as 6.54%, 16.05%, 1.67%, and 3.61%. Hence, SOA-ANN gives best result as compared to other models.

Keywords Solar power generation forecasting (SPGF) · Machine learning (ML) · Artificial neural network (ANN) · Optimized artificial neural network (OANN) · Crow search algorithm (CSA) · Seagull optimization algorithm (SOA)

Abbreviations

ANN	Artificial neural network
ARIMA	Auto-regressive integrated moving average
ARMA	Autoregressive moving average
ARMAX	Autoregressive integrated moving average with exogenous inputs
BP	Backpropagation
CSA	Crow search algorithm
CSA-ANN	Crow search algorithm-based artificial neural network
CSPTCL	Chhattisgarh State power transmission company limited
DE	Differntial evolution
DE-ANN	Differential evolution-based artificial neural network
ESN	Echo state network
GA	Genetic algorithm
MAE	Mean absolute error
MAPE	Mean absolute percentage error
ML	Machine learning
MSE	Mean square error
OANN	Optimized artificial neural network
PSO	Particle swarm optimization
PSO-ANN	Particle swarm optimization-based artificial neural network
R^2	Co-relation of determination
RNN	Recurrent neural network
SOA	Seagull optimization algorithm
SOA-ANN	Seagull optimization algorithm-based artificial neural network
SPG	Solar power generation
SPGF	Solar power generation forecasting

✉ Shashikant Kaushaley
shashikantkaushaley20@gmail.com

[1] National Institute of Technology Raipur, Raipur, Chhattisgarh, India

[2] Vignan's Institute of Information Technology, Andhra Pradesh, India

SVM	Space vector machine

1 Introduction

In the last two decades, electricity demand is kept on growing, due to development in science and technology. Technology has changed every single aspect of lives such as ease of access to information, saving time, ease of mobility, better communication, cost and efficiency of the system, innovation in many fields, and artificial intelligence, i.e., in the present scenario technology is solely dependent on electricity, hence technology and electricity are proportional to each other. An increase in electricity demand is a burden to the power sector companies since generation has to meet demands. Hence, power sector companies are forced to look for solutions and one of the possible solutions is to rely on renewable energy sources such as solar and wind. Both these sources are naturally dependent and intermittent. To rely on renewable energy sources, the power sector company has to do forecasting based on past data. Forecasting has many benefits such as scheduling, planning, coordination among generating stations and devices, maximum utilization of power generating plants, minimizing risk in bidding, reliable and economic operation for the grid and interconnected system, etc. Solar energy is highly nonlinear since it is nature dependent and intermittent. A fast and accurate model is required to forecast the forecasted parameter, which can handle nonlinear data. In the literature, many researchers have created forecasting models for linear data, time-series data, and nonlinear data, but they are not accurate, and hence, it will always be a part of research for the researcher.

Solar power generation (SPG) depends on meteorological factors such as temperature, irradiation, pressure, wind, module temperature, direction, and humidity [1] Some of these factors are taken for prediction purposes. In real time, unpredicted solar power affects the reliability, scheduling, and stability of the system [2, 3], thus forecasting helps to minimize these effects. Based on the availability of data, forecasting is classified into three categories such as *short-term forecasting* useful for unit commitment, generation control, and energy trading, *and medium-term and long-term forecasting* useful for future planning, and handling the operation of grid and transmission systems. [4]. Various statistical forecasting models have been developed by the researcher such as autoregressive moving average (ARMA) [5], autoregressive integrated moving average (ARIMA) [6], autoregressive integrated moving average with exogenous inputs (ARMAX) [7], and statistical time series model [8] which supports linear data, but it does not give promising results for nonlinear data. Various methods have been presented in the literature to handle nonlinear data such as optimization algorithms, machine learning, fuzzy-based system, and hybridization of the algorithms. Machine learning can learn and adapt uncertainties in the system which is useful for prediction applications. This ability of ML has shown great performance in image recognition, signal recognition, stock market trading, etc. [9–11]. Various ML models have been developed for forecasting irradiation and power such as ANN [12, 13], SVM-based estimation and evaluation of solar irradiation [14, 15], and power generation based on the satellite image and SVM [16, 17], some deep learning algorithm [18] which is also presented for power generation forecasting with integrated wavelet [19], ensemble earning approach and autoencoder-based ELM is used for feature learning [20, 21]. Some researchers also presented hybrid models to improve the forecasting model such as fuzzy integrated with ANN [22], in contrast to the ARMA model time delay neural network shows enhanced performance, and hybridization of these two concludes stable and accurate results [5]. ML itself is a good approach for solving problems, but as the usage of ML increases in various applications, some new problem arises in ML such as the possibility of high error, algorithm selection, data acquisition, underfitting, and overfitting. To solve those problems advancement in ML takes place such as the hybrid model, backpropagation (BP) [23] Algorithm in which error is propagated back to minimize the error, recurrent neural network (RNN) [24] is widely used in various application because of its internal memory, Echo State Network (ESN) [25] in which dynamic neurons are used. Apart from this, the performance of ML can be enhanced by tuning ML parameters such as weights and bias or algorithm parameters. One of the approaches is by using an optimization algorithm.

Optimization algorithms have been used in many engineering problems because it finds the best possible solution for a particular problem. Various optimization algorithms have been developed by the researcher, and a review of different optimization algorithms is illustrated in [26], which consists of seventeen optimization algorithms based on biology-based algorithms, five physics-based optimization algorithms, and two geography-based algorithms. A comparison of six different optimization algorithms on the performance of the cam-follower mechanism is illustrated in [27]. In this work, crow search optimization algorithm (CSA) and seagull optimization algorithm (SOA) [28, 29] are used to train the model to get optimized weight values with minimum error and to reduce training time. Highlights of this paper are as follows:

- Two models are developed for prediction, ANN and optimized-ANN (OANN).
- Pseudocode is presented for ANN, CSA-ANN, and SOA-ANN.
- Two different approaches are presented for incorporating optimization algorithms in machine learning.

- A-day-ahead, three-day-ahead, and week-ahead prediction are presented in the paper.
- Percentage improvement of SOA-ANN is presented for midterm solar power generation forecasting.

The work is arranged as follows: Sect. 1. represents Introduction which includes a literature survey to complete this work, Sect. 2 represents the optimization algorithm used in the work, Sect. 3 represents the prediction model ANN and OANN, and Sect. 4 represents Statistical performance to evaluate the performance of the prediction model. Section 5 represents the result and discussion and at last conclusion and reference.

2 Optimization Algorithm

2.1 Crow Search Algorithm

CSA was introduced by *Alireza Askar Zadeh* in 2016. It is a nature-inspired population-based algorithm and works on the intelligence of crows in search of food. Generally, crows steal food from other birds' hidden places. Apart from stealing, it takes precautions so that other crows or birds can't find their hidden place.

This intelligence is developed as an optimization algorithm for many engineering applications. Implementation of the crow search optimization algorithm involves four stages as per their intelligence; they live in groups, memorize hidden places, search other hidden places for food, and protect food from others. These four stages in the optimization process are the initialization of crows, their memory locations, updating the position of crows in search of food as per their memory, and preventing it from thievery as expressed in Eqs. (1), (2), (3), and (4).

$$\text{Crows}: \; X = \begin{bmatrix} x_1^1 & x_2^1 & \cdots x_d^1 \\ \vdots & \vdots & \ldots \vdots \\ x_1^N & x_2^N & \cdots x_d^N \end{bmatrix} \tag{1}$$

$$\text{Memory}: \; m = \begin{bmatrix} m_1^1 & m_2^1 & \cdots & m_d^1 \\ \vdots & \vdots & \ldots & \vdots \\ m_1^N & m_2^N & \cdots & m_d^N \end{bmatrix} \tag{2}$$

where 'N' and 'd' are group size and dimension (or decision variables) over a search space such that the size of vector X and m are [N x d]. Each row in vector X represents a solution to the problem and vector m represents the memory of each crow in which food is hidden; at the initial state, they have no experience; hence, the position of the hidden place for food is initialized. The fitness function is obtained by substituting design variables in the objective function. Now the ith crows try to steal the food from observing the jth crow from its memory and jth crow tries to fool ith crow by changing the location of the food on the territory by knowing the intention of the crow and the ith crow changes its position and memory as:

$$\text{Update Position}: \; X^{i,iter+1} = \begin{cases} X^{i,iter} + r_i + fl^{i,iter} * \left(m^{j,iter} - X^{i,iter}\right) & r_j \geq AP^{j,iter} \\ \text{a random position} & \text{otherwise} \end{cases} \tag{3}$$

$$\text{Update Memory}: \; m^{i,iter+1} = \begin{cases} X^{i,iter} & f\left(x^{i,iter+1}\right) is\ better\ than\ f\left(m^{i,iter}\right) \\ m^{i,iter} & otherwise \end{cases} \tag{4}$$

where 'r_i' and 'r_j' are two random numbers ranging between [0,1], which decides the crow to protect their food from other crows by comparing with the awareness probability 'AP'whose value is fixed at 0.1, which enhances the intensification or diversification if the values of 'AP' is low and high. 'fl'is the flight length whose value is 2.

2.2 Seagull Optimization Algorithm

Seagulls are omnivorous seabirds that eat insects, reptiles, earthworms, fishes, etc. their scientific name is Larade and found all over the planet. They are intelligent and uses their feet to attract hidden earthworm under the ground by producing a sound like rain, and they also attract fish with bread crumbs. This intelligence helps them to find food while migrating. The attacking and migrating behavior of seagulls are modeled for optimization problems for solving various engineering problems. For mathematical modeling, the optimization work in two phases exploration and exploitation phase is inspired by the behavior of seagulls during migration and attack.

Exploration (Migration of seagulls) Exploration is carried out in three phases to move from one position to another position in the search space.

(1) *Avoiding collision of seagulls*: A variable 'A' is introduced to avoid collision between seagulls in search of the new position of seagulls (i.e., search agents.)

$$\overrightarrow{C_s} = A * \overrightarrow{P_s(x)} \tag{5}$$

where 'A' represents the movement behavior of the search agent in the search space, '$\overrightarrow{P}_s$' represents the current position of the seagull, 'x' indicates the current iteration and '$\overrightarrow{C_s}$' represents the position of the search agent which does not collide with another search agent.

$$A = f_c - \left(x * \left(\frac{f_c}{\text{itermax}} \right) \right) \tag{6}$$

where 'f_c' is set to 2, which linearly decreases from 'f_c' to 0 and is employed to control the frequency of variable 'A'.

(2) *Movement toward best neighbor direction*: After successfully avoiding the collision between neighboring search agents, the search agents are moving toward the best neighbor.

$$\overrightarrow{M_s} = B * \left(\overrightarrow{P_{bs}(x)} - \overrightarrow{P_s(x)} \right) \tag{7}$$

$$B = 2 * A^2 * rd \tag{8}$$

where '$\overrightarrow{P_s}$' represents the position of the search agent and '$\overrightarrow{P_{bs}}$' represents the position of the best search agent and '$\overrightarrow{M_s}$' represents the position of the search agent moving toward the best search agent. To balance the exploration and exploitation phase, a variable 'B' is introduced and randomized with a variable '*rd*' which lies in the range of [0,1].

(3) *Remain close to the best search agent*: The closeness of the search agent toward the best search agent is represented by the '$\overrightarrow{D_s}$', due to which the search agent updates the position.

$$\overrightarrow{D_s} = \left| \overrightarrow{C_s} + \overrightarrow{M_s} \right| \tag{9}$$

Exploitation (Attacking behavior of seagulls) Exploitation refers to utilizing their knowledge, experience, and history during their search process. Seagulls attack their prey in a spiral manner which can be modeled in x, y, and z planes; they can change their speed and angle of attack continuously.

$$x' = r * \cos(k) \tag{10}$$

$$y' = r * \sin(k) \tag{11}$$

$$z' = r * k \tag{12}$$

$$r = u * e^{kv} \tag{13}$$

where 'k' is a random number that lies between [0,2π], '*u*' and '*v*' are constant to define their spiral shape whose value is 1, '*e*' is the base of the natural logarithm and '*r*' is the radius of each turn of the spiral. Finally, the search agents update their positions concerning the best search agent and can be modeled as Eq. (14).

$$\overrightarrow{P_s}(x) = \left(\overrightarrow{D_s} * x' * y' * z' \right) + \overrightarrow{P_{bs}}(x) \tag{14}$$

where $\overrightarrow{P_s}(x)$ saves the best solution and updates the position of the search space.

3 ANN and OANN

A computational model for neural networks was first introduced in 1943 [30] and perceptron-based first artificial neural networks is introduced in 1957. ANN consists of three layers input layer, an output layer, and a hidden layer. The structure of ANN used in this work is portrayed in Fig. 1 in which the input layer consists of three nodes each node represents solar parameters such as time, solar irradiation, and temperature. The hidden layer consists of five nodes with a sigmoidal activation function, and the output layer consists of a single node with a sigmoidal activation function. Weights between the input layer to the hidden layer are represented by W_{21} and between the hidden layer to the output, layer are represented by W_{32}, bias in the hidden layer and output layer are not considered in this work. The work presented in this paper consists of two parts. In the first part, ANN architecture is trained and tested with the simple architecture as portrayed in Fig. 1; in the second part, weights W_{21} and W_{32} are updated by an optimized algorithm as discussed in section II; the reason for approaching the second part of the work is to train the network in such a way that its performance is enhanced. The important part of the work is how the network is trained. In

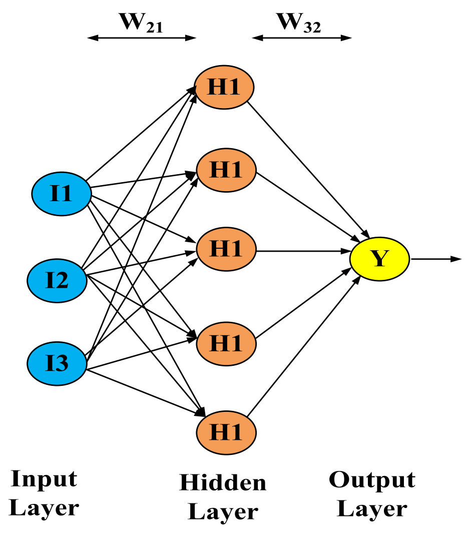

Fig. 1 Architecture of ANN

this section, training process with a conventional approach and with an optimized approach is illustrated.

The conventional approach of Training ANN For training, any network first and foremost important is the collection of data from a valid source. In this work, the data are collected from Chhattisgarh State Power Transmission Company Limited (CSPTCL) from February 2019 to December 2019. Hourly data are collected and out of which only useful data are taken, i.e., the point where the output is obtained (i.e., nonzeros data are considered, during night time output data are zero hence it is excluded in the training part) These data are normalized between 0 and 1 for proper handling of data. These data consist of input and output in which time, solar irradiance, and temperature are taken as input parameters (i.e., $X_{\mathrm{i}} = [3 \times 1]$) and solar power is taken as target data (i.e., $X_{\mathrm{t}} = [1 \times 1]$); now randomly initialize the value of the weights and the network is ready for training. Error and hidden layer values are calculated mathematically as:

$$\text{Input to the hidden layer is : } \mathrm{IH}_1 = \mathrm{W}_{21} - \mathrm{X}_{\mathrm{i}} \tag{15}$$

$$\text{The output of the hidden layer is : } \mathrm{OH}_1 = \frac{1}{1 + e^{-\mathrm{IH}_1}} \tag{16}$$

$$\text{Input to output layer is : } \mathrm{IY} = \mathrm{W}_{32} - \mathrm{O}_{\mathrm{H1}} \tag{17}$$

$$\text{The output of the output node is } Y = \frac{1}{1 + \mathrm{e}^{-\mathrm{IY}}} \tag{18}$$

$$\text{Error } E = Y - X_t \tag{19}$$

$$W_{32,\,\mathrm{new}} = W_{32,\,\mathrm{old}} + \mathrm{alpha} * \mathrm{E} * \mathrm{delta} * \mathrm{O}'_{\mathrm{H1}} \tag{20}$$

$$W_{21,\,\mathrm{new}} = W_{21,\,\mathrm{old}} + \mathrm{alpha} * \left(\left(\mathrm{E} * \mathrm{delta} * \mathrm{W}'_{32}\right) * \mathrm{del}_{\mathrm{H1}}\right) * \mathrm{X}_{\mathrm{i}} \tag{21}$$

$$\mathrm{del}_{\mathrm{H1}} = \mathrm{OH}_1 (1 - \mathrm{OH}_1) \tag{22}$$

$$\mathrm{del}_Y = Y(1 - Y) \tag{23}$$

where Eqs. (15)–(18) are the mathematical calculation to obtain the output from ANN architecture and Eqs. (20) and (21) are used to update the weights. Error is calculated with Eq. (19) and the derivative of the output layer and hidden layer is obtained by Eqs. (22) and (23). The net weight is obtained in two ways, i.e., by taking the average of the weights obtained by all the input samples or by selecting one of the weights obtained by all input samples, whose mean square error is minimum. In this work, second approach is used to calculate net weight, not the average weight. The total size of samples for input is 3000 × 3; i.e., 3000 samples are considered, and the size of the output is 3000 × 1.

Pseudo code for training conventional ANN

```
Normalize input (Xi) and output data (Xt)and initialize maximum iteration as itermax.
Randomly initialize weight($W_{21}$=[5x3] rows represent the number of neurons in the hidden layer and column
represents the number of neurons in the input layer. $W_{32}$=[1x5])
for i=1:length(input)
    $Xi^i$ considers 1 data set from a total sample
    $Xt^i$
    for iter=1:itermax
        evaluate all the parameters from equations (15)-(19)
        update weight values from equation (20)-(21) up to the itermax
        save (weights values, error, and output)
        *Note: this gives weights value for one sample of input and output data at minimum error
    end
    save weight values for all samples.
    *Note: if the sample size is SS=3000, then the total weight size is 3000 times the actual weight size.
    *$W_{21}$=5x3 with all samples it will be $W_{21,i}$=15000x3 or 5x9000 as per user program (where
    i=1,2,..length(input))
end
$W_{21}$ = min(MSE) for( $W_{21,iter}$ ) is the net weight values
$W_{32}$= min(MSE) for( $W_{32,iter}$ ) is the net weight values
Evaluate the training error with final weight values (i.e. net weight values);
Evaluate error measurements with testing data and predict the output.
```

3.1 OANN

The objective of using ML or deep learning algorithms is to develop a random function approximation or to find a relation between the variables. It uses an algorithm to train the network attributes by minimizing the loss function. Various algorithms have been reported in the literature to minimize the loss such as gradient descent [31], backpropagation [32], stochastic gradient descent method [33], conjugate gradient descent [34], steepest descent algorithm[35], and Rprop algorithm [36]. The advantage of these algorithms is easy implementation, ease to understand, continuous learning, wide application, and efficiency, but these methods have certain demerits such as high computation time except the Rprop algorithm, veering off due to frequent updates, the convergence rate very slow, sensitive to noise and irregularities, may trap in local minima. To overcome these challenges, optimization-based ML is used. The process of incorporating the optimization algorithm in ML is discussed via pseudocode and a generalized structure is given in the form of flowchart as shown in Fig. 2**.** is to implement any optimization algorithm in ML.

ANN training with optimization algorithm The most important factor in any optimization problem is the objective function. In this work, mean square error (MSE) is taken as an objective function as expressed in Eq. (24). The number of populations is taken as 100 and 50 iterations. This is the benefit of the optimization algorithm where the minimum value of error is achieved with a smaller number of iterations. In the conventional approach, number of iterations is taken as 10,000 to reach minimum error. Here the net weight is considered by taking one of the weights obtained from all the samples whose error is less by considering all input samples the same as the weight calculation for ANN.

$$\text{Objective Function} = \frac{1}{\text{N}}\sum_{\text{i}=1}^{\text{N}}\left(A_i - P_i\right)^2 \tag{24}$$

Pseudo code for CSA-ANN

```
Initialize optimization algorithm parameter flight length (fl), awareness probability (AP), number of population (np), design variables, the maximum number of iterations (tmax)
Randomly generate weight matrix and memory matrix same as weights. $W_{21}$=rand[(np*5)x3] and $W_{32}$=rand[(np*1)x5].
Load solar data (training input and target)
Start loop-1 for the number of samples
Assign a value U which contains a set of inputs i.e. U=3x1.
Start loop 2 for the number of populations
Calculate fitness function.
End loop-2
Start loop-3 for the maximum number of iterations
Start loop 4 for the number of populations
 Check if the condition for the AP
Update the weights using equation (3))
else
Update the weights using equation (3)
End if condition
Calculate fitness function.
End loop -4
Start loop 5 for the number of populations
Compare fitness function.
Update memory.
End loop -5
End loop-3
Save weight values
End loop-1
Select one weight values whose mean square error is minimum for input samples.
```

Pseudo code for SOA-ANN

```
Initialize optimization algorithm parameters u, v, number of population (np), design variables (d1, d2), upper bound, lower bound, the maximum number of iterations (tmax)
Randomly generate weight matrix and memory matrix same as weights. $W_{21}$=rand[(np*5)x3] and $W_{32}$=rand[(np*1)x5].
Load solar data (training input and target)
Start loop-1 for np
w1=W21{i} and w2=W32{i} i=1,2,..np
Start loop 2 for the number of samples
Predict output with ANN architecture.
End loop-2
Evaluate fitness function.
End loop-1
Calculate the best fitness function and store weights value correspondingly.
Start loop-3 for a maximum number of iterations
Start loop 4 for the number of populations
Calculate all variables (5)-(13) and update the positions (14)
End loop-4
Bound weight values if it exceeds the range.
% Calculate new fitness value
Start loop-5 for np
w1=W21{i} and w2=W32{i} i=1,2,..np
Start loop 6 for the number of samples
Predict output with ANN architecture.
End loop-6
Evaluate fitness function.
End loop-5
Calculate the newly best fitness function and store the value of the weights correspondingly.
Start loop-7 for np
Compare the new fitness function with the old fitness function and update it in the old fitness variable
End loop-7
Evaluate best fitness value after comparison and save weight values
End loop 3
Select final weight values whose fitness function is minimum.
```

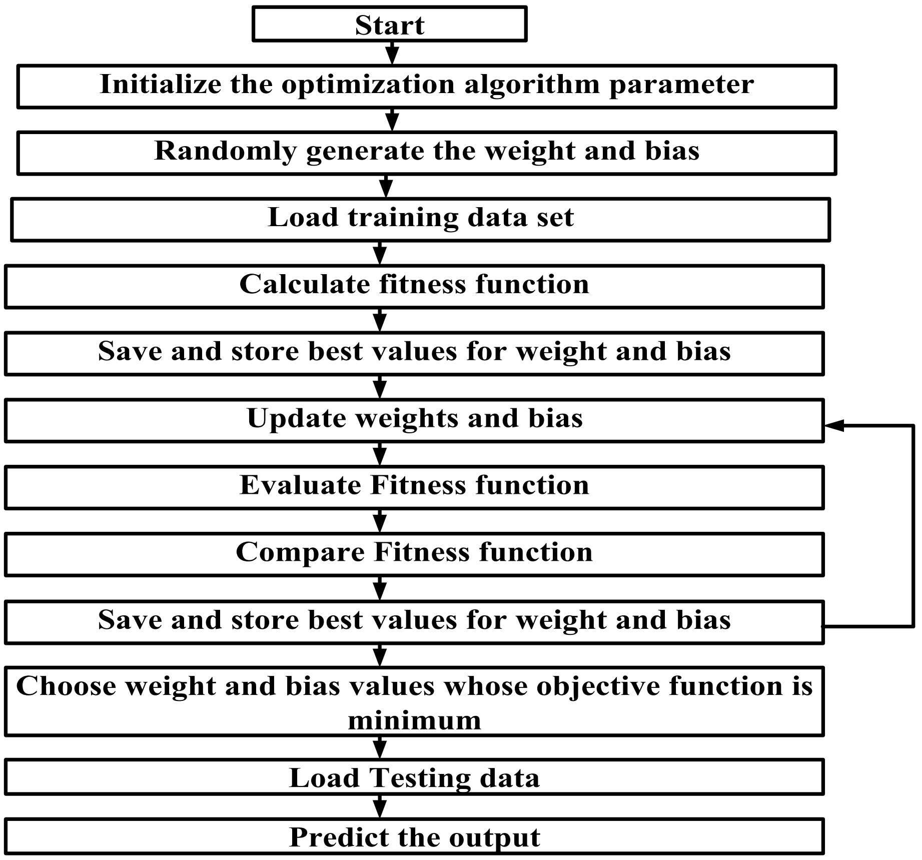
Start
Initialize the optimization algorithm parameter
Randomly generate the weight and bias
Load training data set
Calculate fitness function
Save and store best values for weight and bias
Update weights and bias
Evaluate Fitness function
Compare Fitness function
Save and store best values for weight and bias
Choose weight and bias values whose objective function is minimum
Load Testing data
Predict the output

Fig. 2 Flowchart of OANN

The optimization algorithm has many advantages such as it can solve complex problems, is robust in nature, has fast convergence, and provides a global solution. The advantages of the optimization algorithm incorporated with ML give good results as reported in the literature in many applications such as [38] illustrates DE-based ANN for time series forecasting, [38] illustrates nonlinear-based forecasting done for a very short time interval, PSO-based ANN is illustrated in [39], [40], [41], GA-based ANN is illustrated in [42], and the comparison between GA-ANN and PSO- ANN is illustrated in [43], and the results show that GA-ANN has higher accuracy over PSO-ANN and BP algorithm. This paper demonstrates the comparison between ANN, CSA-ANN, and SOA-ANN along with existing methods such as DE-ANN and PSO-ANN.

4 Statistical Measures

To predict the closeness to actual data statistical measures are used. Mean square error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and correlation of regression (R^2) [4] are used to evaluate the performance of the model; these four performance indices are mathematically represented in Eqs. (25)–(28). If the sample size is large and the prediction depends on one single value (in this work weights are the important parameter for both ANN and OANN), then the reduction in MSE is more advantageous for performance measures. If it is done in the simulation part, then integral time absolute error (ITAE) and integral time square error (ITSE) are used for performance measures. In general, the value of MSE is bigger than MAE. Co-relation of determination (R^2) gives the regression line ranges between 0 and 1. If its value is near one, it exhibits strong linear relation. MAPE is associated with the accuracy of the predicted data. A lesser value represents more accuracy.

$$\mathrm{MAE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} |A_{\mathrm{i}} - P_{\mathrm{i}}| \tag{25}$$

$$\mathrm{MSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} (A_i - P_i)^2 \tag{26}$$

$$\mathrm{MAPE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}} \left|\frac{A_i - P_i}{P_i}\right| * 100 \tag{27}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (A_i - P_i)^2}{\sum_{i=1}^{N} (A_i - \mathrm{mean(P_i)})^2} \tag{28}$$

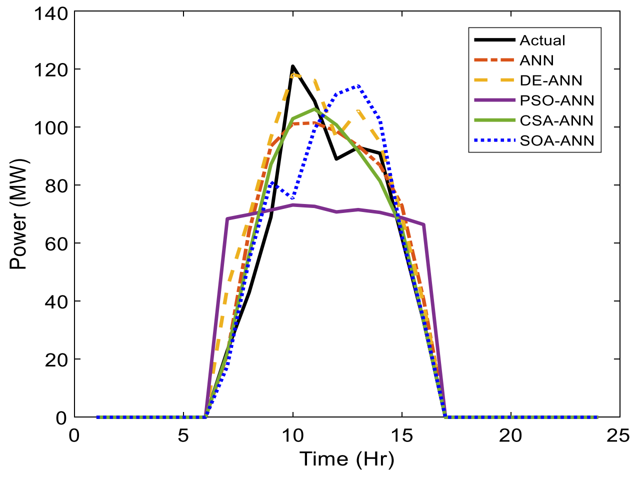

Fig. 3 Comparison of forecasted model output with actual data for a-day-ahead prediction

5 Results and Discussion

Solar power generation forecasting for Chhattisgarh State Power Transmission Company Limited (CGPTCL) is done for a short term (i.e., a-day-ahead) and midterm (i.e., three-day-ahead and a week-ahead prediction) (*Note: Short-term is used for the prediction for an hour ahead, a-day-ahead and two-day-ahead prediction, midterm prediction involves for a 3–15 days ahead, i.e., a week or two-week-ahead prediction*). The data used in the work are collected from CSPTCL from Feb 2019 to Dec 2019. Hourly data is considered for the forecasting model in which time, temperature, and irradiation is taken as input. The data are normalized between 0 and 1, for proper handling of data, and at last, the data are de-normalized to their original value for comparison with the actual data. The model used in the work is ANN and optimized ANN model. Two optimization-based forecasting model is developed one is CSA-ANN and the other is SOA-ANN. These models are compared with conventional ANN along with existing methods for validation of work such as DE-ANN and PSO-ANN, for the short-term and midterm

Fig. 4 Comparison of forecasted model output with actual data for three-day-ahead prediction

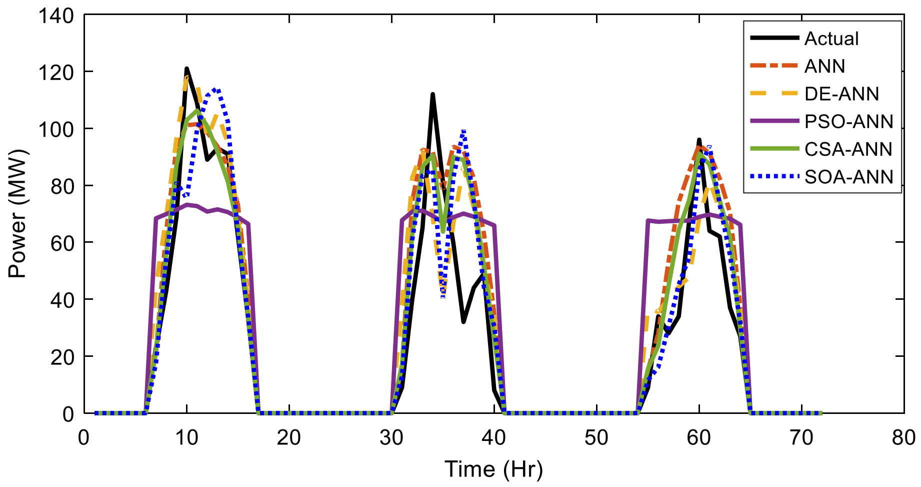

Fig. 5 Comparison of forecasted model output with actual data for a week-ahead prediction

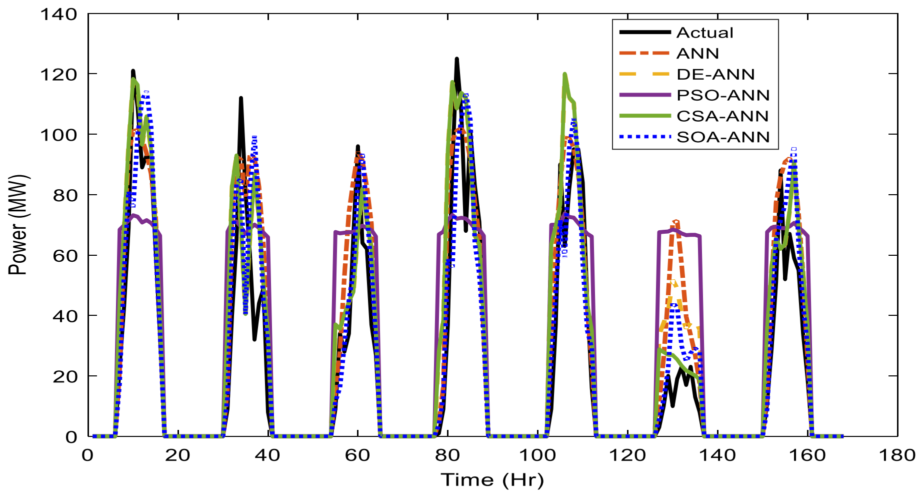

Table 1 Performance analysis for a-day-ahead prediction

Models	MAE	MSE	MAPE	R^2
ANN	**4.8026**	**78.8124**	**0.1554**	**0.9539**
DE-ANN	5.1495	96.5364	0.1742	0.9436
PSO-ANN	10.8075	363.6080	0.3624	0.7875
CSA-ANN	6.3995	129.4635	0.2936	0.9243
SOA-ANN	5.9080	170.0193	0.2044	0.9006

Bold values represent the best values, based on which model performance is measured/evaluated

Table 2 Performance analysis for 3-day-ahead prediction

Models	MAE	MSE	MAPE	R^2
ANN	7.6154	**203.8768**	**0.2994**	**0.8266**
DE-ANN	8.1312	237.8581	0.3341	0.7987
PSO-ANN	11.8647	454.2524	0.4068	0.6138
CSA-ANN	7.6599	231.6082	0.5601	0.8030
SOA-ANN	**7.1511**	214.9188	0.3242	0.8172

Bold values represent the best values, based on which model performance is measured/evaluated

Table 3 Performance analysis for a week-ahead prediction

Models	MAE	MSE	MAPE	R^2
ANN	7.5692	204.3490	0.3235	0.8163
DE-ANN	9.1062	276.1053	0.3795	0.7518
PSO-ANN	13.5229	580.7694	0.4584	0.4778
CSA-ANN	7.6633	273.3283	0.4634	0.7543
SOA-ANN	**6.5406**	**171.5412**	**0.3181**	**0.8458**

Bold values represent the best values, based on which model performance is measured/evaluated

duration. The results associated with short-term prediction are shown in Fig. 3, and midterm prediction is shown in Figs. 4 and 5.

Each Figs. 3–5 represents the comparison of forecasted data with the actual data which is shown in the legends in which 'Actual' represents the actual data to which the comparison is made. 'ANN' represents forecasted data based on the ANN model, 'DE-ANN' represents the forecasted data on the differential evolution-based ANN model, 'PSO-ANN' represents the forecasted data on the particle swarm optimization-based ANN model, 'CSA-ANN' represents the forecasted data based on crow search algorithm-based ANN model and last 'SOA-ANN' represents the forecasted data based on SOA-based ANN model.

From Figs. 3 and 4, it is observed that the ANN, DE-ANN, and SOA-ANN are compatible with each other, but the other method's PSO-ANN performance is poor and gets saturation point, whereas the CSA-ANN can follow the path of the actual data (i.e., the slope of actual data matches with the slope of CSA-ANN forecasted data.). This can be observable with a statistical comparison between the models as tabulated in Tables 1 and 2, and the good results are shown in bold letters. For predicting a day, ANN performance is the best and DE-ANN is close to the ANN result. But as the prediction of the day increases, the ANN and DE-ANN show poor performance as compared to SOA-ANN.

From Fig. 5, it is observed that during bad weather, i.e., on the 6th day (120–144 h), ANN gives poor performance,

Table 4 Percentage improvement of SOA-ANN over ANN

Models	MAE	MSE	MAPE	R^2
SOA-ANN (%)	**6.54**	**16.05**	**1.67**	**3.61**

Bold values represent the best values, based on which model performance is measured/evaluated

Table 5 Trained weight values of the forecasting model

Model	W21			W32
ANN	− 4.3961	− 0.0242	− 0.3756	− 4.1420
	− 5.3350	− 3.7606	1.2344	− 7.7827
	− 2.3999	5.0129	6.4149	4.9361
	− 3.2917	0.7139	3.1321	2.0041
	1.6386	1.3233	0.5491	− 5.7450
DE-ANN	3.2956	− 0.8057	− 1.1899	0.2251
	1.9809	1.4915	− 0.4964	2.0392
	1.5727	− 1.6336	− 1.5735	− 2.8978
	− 1.7455	4.0728	− 3.3164	0.2511
	− 0.4482	− 4.4539	2.6194	− 2.1136
PSO-ANN	1.5285	0.7425	0.0103	− 0.1936
	0.4771	1.6048	0.8490	0.7474
	0.2124	− 0.4013	0.8790	− 0.3993
	0.6191	− 0.5181	1.3118	− 0.4951
	− 0.5846	1.0002	0.8089	− 0.0990
CSA-ANN	− 2.6088	0.5356	1.7944	1.4729
	− 4.0655	0.0362	3.8292	3.8960
	− 6.4193	− 0.2250	2.0937	− 6.5431
	− 0.4764	1.6169	− 0.4085	− 1.7236
	− 0.4332	− 5.9533	− 1.2065	− 6.9649
SOA-ANN	4.8357	− 0.6254	− 0.7010	2.8178
	− 4.3207	− 4.8357	1.6910	− 4.8357
	− 4.8357	4.8357	− 4.8357	− 4.3835
	− 1.9649	− 4.8357	− 4.8357	− 4.8357
	4.3037	− 4.8357	− 2.1809	− 4.1760

whereas the OANN model can predict bad weather days also. It is also observed that on that particular day the SOA-ANN is very close to actual data. It is therefore concluded that SOA-ANN can predict low as well as high data. Still, we can't conclude the performance of the model, the conclusion can be reached by analyzing its statistical parameters.

The performance of the forecasting model for prediction is evaluated based on statistical parameters. The statistical parameters used in the work for comparison are MAE, MSE, MAPE, and R^2. These four parameters are significantly used in forecasting for evaluation. The minimum value of MAE, MSE, MAPE, and maximum value for R^2 represents the best performance for the forecasting model. So, a comparison is made for the forecasting model with this statistical parameter for short-term and midterm ahead prediction which is shown in tabulation format. Table 1, represents the comparison for an a-day-ahead prediction. Table 2 represents the comparison for three-day-ahead predictions, and Table 3 represents the comparison for a week-ahead prediction.

From Table 1, it is observed that ANN exceeds the performance of both OANN models. It can be concluded that for short-term prediction ANN gives better results as compared with other models.

From Table 2, it is observed that ANN exceeds the performance of both OANN models, but its performance over MAE is poor in comparison with SOA-ANN. The percentage improvement of SOA-ANN is **6.5%,** which is calculated by using Eq. (28).

From Table 3, it is observed that SOA-ANN outperforms ANN and CSA-ANN. The percentage improvement of SOA-ANN based on the statistical parameters is tabulated in Table 4, which is obtained by substituting values in Eq. (28).

ANN accuracy is high for short-term prediction which is its advantage but its performance deteriorates for midterm forecasting. To increase its performance the training algorithm is modified via a global optimization algorithm. Its performance is tabulated in Tables 2 and 3, which guarantees the results.

$$\text{Improvement\%} = \frac{|\text{ANN}_{\text{metric}} - \text{SOA}_{\text{metric}}|}{\text{ANN}_{\text{metric}}} * 100 \tag{28}$$

On observing both figures and tables, it can be said that not all optimization algorithm-based forecasting model improves the performance of conventional machine learning-based prediction model.

The weights obtained after training of the forecasted model are shown in tabular format in Table 5. These trained weights can be used for this particular problem only because the weights are obtained for a particular dataset (i.e., time, irradiation, and temperature in sequence).

The average training time for the model's ANN, DE-ANN, PSO-ANN, CSA-ANN, and SOA-ANN is obtained as 445.4128, 177.5161, 128.2633, 75.6305, and 53.1360 s. It is observed that the training algorithm for SOA-ANN is the least as compared to other models. Hence, it can be concluded that the OANN model is fast as compared to the conventional model.

6 Conclusion

SPGF is important for power sector companies in terms of energy bidding, scheduling, etc. to support power sector Companies, an optimization-based machine learning model is developed, i.e., CSA-ANN and SOA-ANN. ML model has high accuracy for time series data but for nonlinear data its accuracy is low, to enhance the ML model Optimization technique is used. This work helps the researcher to incorporate computation techniques into the machine learning model. CSA-ANN and SOA-ANN models are developed and compared with the conventional methods, i.e., ANN, and the optimization-based model is compared with DE- and PSO-based ANN model since these two global optimization algorithms are widely used. The trained weight values are shown in Table 5. The developed models were performed to predict SPG for a day ahead, three day ahead, and a week ahead. By comparing the results, it is concluded that the ANN model predicts better than OANN models for short-term duration, i.e., a-day-ahead prediction. For midterm prediction, SOA-ANN gives the best result as compared to other ML models. ANN performance is enhanced or its limitations/demerits are overcome by changing its training algorithm. Incorporating the optimization algorithm in the training section enables it to forecast midterm also. The percentage improvement of SOA-ANN in comparison with the ANN model is obtained as 6.54%, 16.05%, 1.67%, and 3.61% in terms of MAE, MSE, MAPE, and R^2. Hence, it can conclude that the ANN performance can be enhanced by an optimization algorithm and the overall performance of SOA-ANN is good as compared to the other methods.

References

1. Das, U.K., et al.: Forecasting of photovoltaic power generation and model optimization: A review. Renew. Sustain. Energy Rev. **81**, 912–928 (2017). https://doi.org/10.1016/j.rser.2017.08.017
2. Strzalka, A.; Alam, N.; Duminil, E.; Coors, V.; Eicker, U.: Large scale integration of photovoltaics in cities. Appl. Energy **93**, 413–421 (2012). https://doi.org/10.1016/j.apenergy.2011.12.033
3. Woyte, A., et al.: "Voltage fluctuations on distribution level introduced by photovoltaic systems. IEEE J. Mag. **21**(1), 202–209 (2017)
4. Sahu, R.K.; Shaw, B.; Nayak, J.R.; Shashikant,: Short/medium term solar power forecasting of Chhattisgarh state of India using modified TLBO optimized ELM. Eng. Sci. Technol. an Int. J. **24**(5), 1180–1200 (2021). https://doi.org/10.1016/j.jestch.2021.02.016
5. Ji, W.; Chee, K.C.: Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN. Sol. Energy **85**(5), 808–817 (2011). https://doi.org/10.1016/j.solener.2011.01.013
6. Pieri, E.; Kyprianou, A.; Phinikarides, A.; Makrides, G.; Georghiou, G.E.: Forecasting degradation rates of different photovoltaic systems using robust principal component analysis and ARIMA. IET Renew. Power Gener. **11**(10), 1245–1252 (2017). https://doi.org/10.1049/iet-rpg.2017.0090
7. Li, Y.; Su, Y.; Shu, L.: An ARMAX model for forecasting the power output of a grid-connected photovoltaic system. Renew. Energy **66**, 78–89 (2014). https://doi.org/10.1016/j.renene.2013.11.067
8. Prema, V.; Uma Rao, K.: "Development of statistical time series models for solar power prediction." Renew Energy **83**, 100–109 (2015). https://doi.org/10.1016/j.renene.2015.03.038
9. Decencière, E., et al.: TeleOphta: Machine learning and image processing methods for teleophthalmology. Irbm **34**(2), 196–203 (2013). https://doi.org/10.1016/j.irbm.2013.01.010
10. Sun, Y.; Babu, P.; Palomar, D.P.: Majorization-minimization algorithms in signal processing, communications, and machine learning. IEEE Trans. Signal Process. **65**(3), 794–816 (2017). https://doi.org/10.1109/TSP.2016.2601299
11. Dash, R.; Dash, P.K.: A hybrid stock trading framework integrating technical analysis with machine learning techniques. J. Financ. Data Sci. **2**(1), 42–57 (2016). https://doi.org/10.1016/j.jfds.2016.03.002
12. Mathioulakis, E.; Panaras, G.; Belessiotis, V.: Artificial neural networks for the performance prediction of heat pump hot water heaters. Int. J. Sustain. Energ. **37**(2), 173–192 (2018). https://doi.org/10.1080/14786451.2016.1218495
13. Mubiru, J.; Banda, E.J.K.B.: Estimation of monthly average daily global solar irradiation using artificial neural networks. Sol. Energy **82**(2), 181–187 (2008). https://doi.org/10.1016/j.solener.2007.06.003
14. Chen, J.L.; Li, G.S.: Evaluation of support vector machine for estimation of solar radiation from measured meteorological variables. Theor. Appl. Climatol. **115**(3–4), 627–638 (2014). https://doi.org/10.1007/s00704-013-0924-y
15. Chen, J.L.; Bin Liu, H.; Wu, W.; Xie, D.T.: Estimation of monthly solar radiation from measured temperatures using support vector machines: A case study. Renew. Energy **36**(1), 413–420 (2011). https://doi.org/10.1016/j.renene.2010.06.024
16. Jang, H.S.; Bae, K.Y.; Park, H.; Sung, D.K.: Solar power prediction based on satellite images and support vector machine. IEEE Trans. Sustain. Energy **7**(3), 1255–1263 (2016). https://doi.org/10.1109/TSTE.2016.2535466
17. Zeng, J.; Qiao, W.: Short-term solar power prediction using a support vector machine. Renew. Energy **52**, 118–127 (2013). https://doi.org/10.1016/j.renene.2012.10.009

18. AlKandari, M.; Ahmad, I.: Solar power generation forecasting using ensemble approach based on deep learning and statistical methods. Appl. Comput. Inform. (2019). https://doi.org/10.1016/j.aci.2019.11.002
19. Mishra, M.; Byomakesha Dash, P.; Nayak, J.; Naik, B.; Kumar Swain, S.: Deep learning and wavelet transform integrated approach for short-term solar PV power prediction. Meas. J. Int. Meas. Confed. **166**, 108250 (2020). https://doi.org/10.1016/j.measurement.2020.108250
20. Mohammed, A.A.; Aung, Z.: Ensemble learning approach for probabilistic forecasting of solar power generation. Energies (2016). https://doi.org/10.3390/en9121017
21. Long, H.; Zhang, C.; Geng, R.; Wu, Z.; Gu, W.: A combination interval prediction model based on biased convex cost function and auto-encoder in solar power prediction. IEEE Trans. Sustain. Energy **12**(3), 1561–1570 (2021). https://doi.org/10.1109/TSTE.2021.3054125
22. Melin, P.; Mancilla, A.; Lopez, M.; Mendoza, O.: A hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting. Appl. Soft Comput. J. **7**(4), 1217–1226 (2007). https://doi.org/10.1016/j.asoc.2006.01.009
23. Van Ooyen, A.; Nienhuis, B.: Improving the convergence of the back-propagation algorithm. Neural Netw. **5**(3), 465–471 (1992). https://doi.org/10.1016/0893-6080(92)90008-7
24. Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Phys. D Nonlinear Phenom. **404**, 132306 (2020). https://doi.org/10.1016/j.physd.2019.132306
25. Rodan, A.; Tiňo, P.: Minimum complexity echo state network. IEEE Trans. Neural Netw. **22**(1), 131–144 (2011). https://doi.org/10.1109/TNN.2010.2089641
26. Behera, S.; Sahoo, S.; Pati, B.B.: A review on optimization algorithms and application to wind energy integration to grid. Renew. Sustain. Energy Rev. **48**, 214–227 (2015). https://doi.org/10.1016/j.rser.2015.03.066
27. Abderazek, H.; Yildiz, A. R.; and Mirjalili, S. "Comparison of recent optimization algorithms for design optimization of a cam-follower mechanism," *Knowledge-Based Syst.*, vol. 191, p. 105237, 2020, doi: https://doi.org/10.1016/j.knosys.2019.105237.
28. Dhiman, G.; Kumar, V.: Knowledge-based systems seagull optimization algorithm : Theory and its applications for large-scale industrial engineering problems. Knowl. Based Syst. **165**, 169–196 (2019). https://doi.org/10.1016/j.knosys.2018.11.024
29. Dhiman, G., et al.: 2021 "MOSOA: A new multi-objective seagull optimization algorithm." Expert Syst. Appl. **167**, 114150 (2020). https://doi.org/10.1016/j.eswa.2020.114150
30. McCulloch, W.S.; Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**(4), 115–133 (1943). https://doi.org/10.1007/BF02478259
31. Verma, S.; Thampi, G.T.; Rao, M.: ANN based method for improving gold price forecasting accuracy through modified gradient descent methods. IAES Int. J. Artif. Intell. **9**(1), 46–57 (2020). https://doi.org/10.11591/ijai.v9.i1.pp46-57
32. Singh, B.K.; Verma, K.; Thoke, A.S.: Adaptive gradient descent backpropagation for classification of breast tumors in ultrasound imaging. Procedia Comput. Sci. **46**(2014), 1601–1609 (2015). https://doi.org/10.1016/j.procs.2015.02.091
33. Vijayalakshmi, K.; Vijayakumar, K.; Nandhakumar, K.: Prediction of virtual energy storage capacity of the air-conditioner using a stochastic gradient descent based artificial neural network. Electr. Power Syst. Res. **208**, 107879 (2022). https://doi.org/10.1016/j.epsr.2022.107879
34. Chattopadhyay, S.; Chattopadhyay, G.: Conjugate gradient descent learned ANN for Indian summer monsoon rainfall and efficiency assessment through Shannon-Fano coding. J. Atmos. Solar Terr. Phys. **179**, 202–205 (2018). https://doi.org/10.1016/J.JASTP.2018.07.015
35. Santosh Kumar, G.; Rajasekhar, K.: Performance analysis of Levenberg-Marquardt and steepest descent algorithms based ANN to predict compressive strength of SIFCON using manufactured sand. Eng. Sci. Technol., Int. J. **20**(4), 1396–1405 (2017). https://doi.org/10.1016/j.jestch.2017.07.005
36. Saputra, W.; Tulus, T.; Zarlis, M.; Sembiring, R.W.; Hartama, D.: Analysis resilient algorithm on artificial neural network backpropagation. J. Phys.: Conf. Series (2017). https://doi.org/10.1088/1742-6596/930/1/012035
37. Behera, S.P.H.S.: Time series forecasting using differential evolution-based ANN modelling scheme. Arab. J. Sci. Eng. **45**(12), 11129–11146 (2020). https://doi.org/10.1007/s13369-020-05004-5
38. Hu, Y.; Chen, L.: A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and differential evolution algorithm. Energy Convers. Manage. **173**, 123–142 (2018). https://doi.org/10.1016/j.enconman.2018.07.070
39. Asadnia, M.; Chua, L. H. C.; Qin, X. S.; Asce, A. M.; & Talei, A. (2014). Improved particle swarm optimization – based artificial neural network for rainfall-runoff modeling. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000927.
40. Adhikari, R.; & Agrawal, R. K. (n.d.). Effectiveness of PSO based neural network for seasonal time series forecasting. pp.231–244.
41. Riaz, S.; Khan, A.; Riaz, S.; & Khan, A. (2007). Short Term load forecasting using particle swarm optimization based ANN approach
42. Patra, S. K. (2008). Short term load forecasting using neural network trained with genetic algorithm & particle swarm optimization Sanjib Mishra. pp. 606–611. https://doi.org/10.1109/ICETET.2008.94
43. Tonnizam, E.; Roohollah, M.; Faradonbeh, S.; Jahed, D.: An optimized ANN model based on genetic algorithm for predicting ripping production. Neural Comput. Appl. **28**(s1), 393–406 (2017). https://doi.org/10.1007/s00521-016-2359-8