In Proc. 4th IEEE International Smart Cities Conference (ISC2 2018), September 16-19, 2018, Kansas City, Missouri, USA

Short-Term PV Output Forecasts with Support Vector Regression Optimized by Cuckoo Search and Differential Evolution Algorithms

Musaed Alrashidi, Massoud Alrashidi, Manisa Pipattanasomporn, Saifur Rahman
Bradley Department of Electrical and Computer Engineering, Advanced Research Institute,
Virginia Tech, USA

***Abstract*—Renewable energy sources have gained momentum in electric power systems. Without the ability to precisely forecast their power production, integrating these sources to the electric power grid may affect grid stability. Support Vector Regression (SVR) is proven to be able to deal with nonlinearity in forecasting problems. However, determining the appropriate parameters for SVR is the key issue in attaining an accurate SVR forecasting model. The objective of this paper is to forecast the output power of solar photovoltaic (PV) systems using support vector regression, of which its parameters are optimized by Cuckoo Search (CS) and Differential evolution (DE) algorithms. Real-world solar data from the 6.4kW rooftop solar PV unit located at the Advance Research Institute (ARI) of Virginia Tech in Arlington, Virginia, are used as the basis of the forecast. Six input variables are used for model development, namely day, month, hour, global normal radiation, temperature and wind speed. Model performance is evaluated using statistical indicators. Results indicate that SVR with Radial Basis function optimized by CS and DE give the most accurate forecasts.**

Keywords—Solar power forecasting, photovoltaic system, artificial neural network, support vector regression, meta-heuristic optimization techniques

I. Introduction

Growing penetration of renewable resources have been observed over the past two decades and their presence in electrical power networks becomes inevitable. The increasing penetration of renewables is accompanied by some issues related to the power system operation and design, such as system protection, control, power quality and optimal power system operation [1]. Embracing high penetration of solar production has led to necessary adjustments in the power system operation, including the requirement of additional ancillary services to manage the variability in renewable energy output [2]. In fact, these ancillary services are very expensive and adding them may cancel out economic advantages of these renewable resources. These associated problems primarily occur due to the fact that PV is inherently an intermittent source [1]. Therefore, being able to accurately forecast solar PV output can help mitigate technical challenges in renewable energy integration [3].

Relevant forecasting topics in electric power systems address aspects of load, wind, solar and electricity markets forecasts [3]. In this study, forecasting PV solar power production is of focus. Several research papers have been published attempting to develop different models regarding solar forecasts using variety of machine learning algorithms. While some researchers developed models to estimate solar radiation, global or direct radiation [4][5][6], the others forecasted PV power output using different meteorological parameters with different time horizons [7][8][9].

Authors in [10], for instance, developed a multilayer feed-forward network which has back propagation (BPNN) to predict monthly global solar radiation for six cities in Iran using different weather variables for a six-year period. In addition, authors in [11] built two artificial neural network (ANN) to examine the global radiation and direct normal solar irradiance (DNI) in hourly manner in Salerno, Italy. In [12], PV power forecasting model was created based on a BPNN model to perform 24-hour ahead PV output forecast in Ashland, Oregon. The results showed good accuracy for forecasting the power output of photovoltaic systems.

Authors in [13] proposed a day-ahead PV output prediction algorithm utilizing support vector machine (SVM) and weather classification and results were promising. In [14], a support vector regression (SVR) model using serval weather variables, including cloudiness, was created to forecast the power production of a 1-MW photovoltaic power plant in Kitakyushu, Japan. The results of this study indicated that SVR with the cloudiness factor led to better accuracy. Furthermore, a short term solar irradiance was proposed by [15], where two different SVR models based on clearness index were used. In [16], the authors formulated models using linear least square regression and SVR with radial basis function for predicting solar generation. The outputs of this study indicated that SVR model with seven weather metrics is 27% more precise compared to other forecasted models which use only sky conditions for predictions. The study in [17] forecasted the power output of two wind farms and two PV power plants using machine learning algorithms, which were proved to be the most effective forecasting approaches to deal with non-linear relationship between weather parameters and power production.

The main drawback of applying SVR is that its model accuracy highly depends on the selected parameters, which are very difficult to determine properly. Metaheuristic optimization techniques, such as genetic algorithms (GA), simulated annealing (SA), immune algorithms (IA) and particle swarm optimization algorithm (PSO), have also been used aiming at selecting appropriate parameters. For instance, authors in [18] used GA to optimize SVM model parameters for electricity price forecasting. Furthermore, IA and SA were used in [19] and [20], respectively, to determine the best parameters to forecast Taiwan's annual electric load. The study [19] showed that the proposed models superior to other models such as ANN models.

Differential evolution (DE) and Cuckoo Search (CS) algorithms are recent heuristic optimization techniques. DE was proposed first in 1995 by Storn and Price [21] and CS was introduced by Yang and Deb in 2009 [22]. The main significant feature of DE and CS is that they are easy to implement [23]. Authors in [24] proposed SVR with DE model (DESVR) for the annual electric load forecasting of Beijing city in China. The results demonstrated that the DE algorithm has the ability to determine appropriate parameters of the SVR model and DESVR outperforms other models including SVR with default parameters and BPNN. Authors in [25] used SVR models optimized by various optimization algorithms, including PSO, GA and CS, to forecast short-term wind speed series. The results show that CS-SVR model had the best performance among other optimized SVR models.

Based on the above discussion, this paper proposes PV power forecasting models using SVR with radial basis (RB) and Linear kernel functions, optimized by CS or DE. BPNN is also used to forecast the solar power output for comparison. The total of five models are studied, namely: (i) SVR based on RB kernel function optimized by CS; (ii) SVR based on RB kernel function optimized by DE; (iii) SVR based on linear kernel function optimized by CS; (iv) SVR based on linear kernel function optimized by DE; and (v) BPNN model. Their outputs are compared to identify which model offers superior performance in forecasting PV output power. This comparison is measured using commonly used statistical indicators, including Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE) and Goodness of Fit (R^2).

The paper is organized as follow: Section II describes the framework of the forecasting models used in this study, together with the training and testing datasets and the selected forecasting models. Section III discusses the DE and CS algorithms. Section IV describes the criteria to evaluate model accuracy. Lastly, Section V compares and discusses PV output forecasts based on different forecasting models.

II. Methodology

In this section, the overall study is discussed, together with the datasets used, and the fundamental of back propagation neural network (BPNN) and support vector regression (SVR).

A. Study Framework

The framework of the proposed study is depicted in Fig. 1. The process is explained as follows:

(1) The training and testing datasets are initially normalized before conducting the training and testing processes.

(2) The SVR models with radial basis, where CS and DE algorithms are applied to determine the parameters, are established.

(3) Similarly, the SVR models with linear function, of which parameters are optimized using CS and DE, are established.

(4) The BPNN model with the selected number of layers and hidden neurons (nodes) are established.

(5) The actual and forecasted PV output power generated by the models are compared to evaluate the performance of the forecasting methods.

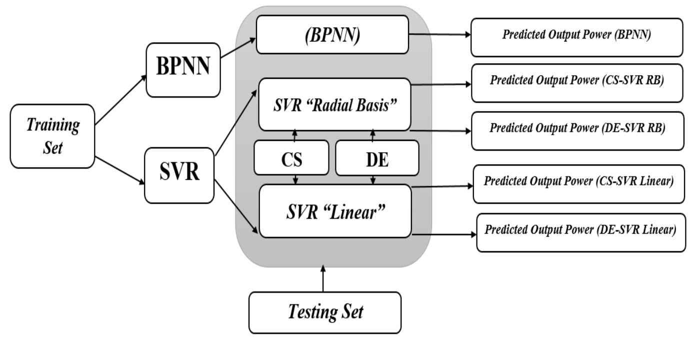

Fig. 1. Framework of the proposed study.

Each process is discussed in details in subsections below.

B. Training and Testing Datasets

The database used to develop the five PV output forecasting models are obtained from the 6.4kW rooftop PV system placed on the Virginia Tech Advanced Research Institute (VT-ARI) building in Arlington, Virginia, as shown in Fig. 2.

Fig. 2. The 6.4kW PV system at VT-ARI building, Arlington VA.

The PV output data in 5-minute intervals for the period of one year from January to December, 2015, are gathered from the unit. The 5-minute resolution dataset is transformed into an 1-hour resolution dataset for use in the models, together with corresponding weather parameters. The entire dataset (X) is divided into two subsets namely: the training dataset, x_{train} and the test dataset, x_{test}, such that $X = x_{train} \cup x_{test}$. In this study, eighty percent (80%) of the data is utilized as the training dataset, while the remaining is used as the testing dataset.

The measured PV output power in 5-minute intervals is exhibited in Fig. 3. A graphical representation of the corresponding ambient temperature over the same one-year period, in Fahrenheit, is shown in Fig. 4. Fig. 5 exhibits the change in PV output power with air temperature during August 10^{th}-20^{th}. It can be noticed that there is a correlation between PV power output and the air temperature.

In this study, six independent weather variables are used as the inputs to forecast the PV output (P_{out}). The selected variables are: month (M), day (D), hour (H), global normal irradiance

(*GNI*), ambient temperature (*T*) and wind speed (*W*). The forecasting model can be expressed as shown in Eq. (1):

$$P_{out} = f(M, D, H, GNI, T, W) \tag{1}$$

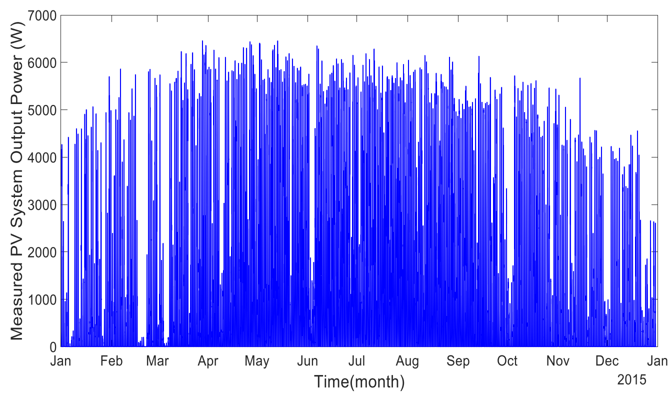

Fig. 3. Measured PV output power during a one-year period

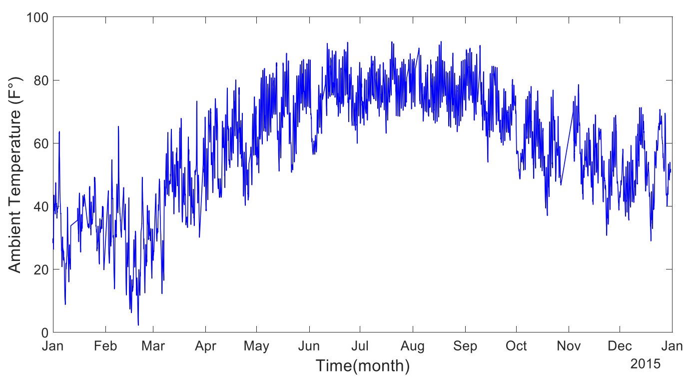

Fig. 4. Ambient temperature in degree Fahrenheit during a one-year period

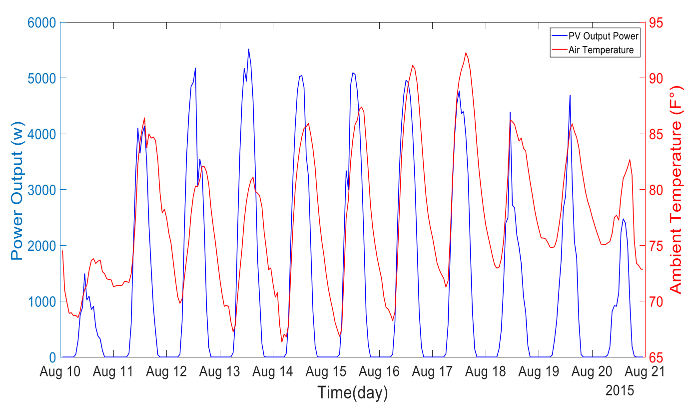

Fig. 5. Relative change in PV output power with temperature (Aug, 10th - 20th)

C. Input Data Normalization

Input data normalization plays a crucial rule in preparing the data before examining the forecasting models. The primary goal of data normalization is to reduce the possibility that features with high numerical values dominating those that have smaller numerical values [26]. In this study, month, day, hour, global solar radiation, temperature and wind speed data are normalized between 0 and 1 using Eq. (2).

$$x_i^n = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{2}$$

Where x_i is the actual data value; x_i^n is the normalized value; x_{max}, x_{min} are the maximum and minimum values corresponding to the actual dataset.

D. Back Propagating Neural Network

Artificial neural network (ANN) has been widely used for different forecasting applications. ANN inspired by the way of how human nervous systems interpret information. In a neural network, back propagation is one of the most popular artificial neural network methods used to conduct the learning process. Fig. 6 shows a multi-layer feed-forward neural network.

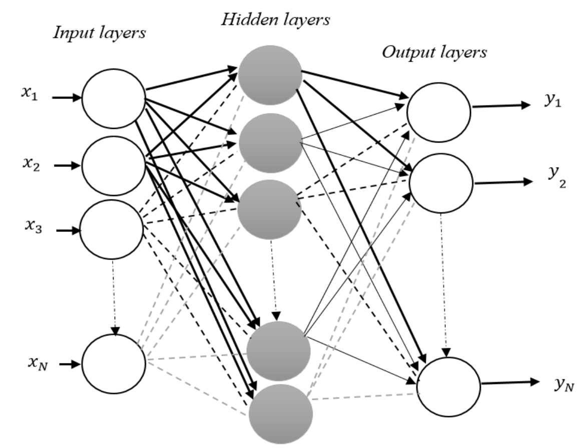

Fig. 6. Feed-forward neural network.

The ANN network is usually constructed with three layers: the input layer $[x_1, x_2, \dots, x_N]^T$, the hidden layer $[h_1, h_2, \dots, h_M]^T$ and the output layer $[y_1, y_2, \dots, y_L]^T$. The output of all nodes (neurons) in the hidden layer are calculated as follows:

$$z_j = \sum_{i=0}^{N} v_{ij} x_i \,, j = 1,2, \dots, M, i = 1,2, \dots, N \tag{3}$$

$$h_j = f(z_j)\,, \qquad j = 1,2, \dots, M \tag{4}$$

Where z_j is the activation value of the jth node in the hidden layer; v_{ij} is the weight connected between input i and hidden node j; h_j is output in the hidden layer; f is the transfer function of the neurons, usually a sigmoid function is used $f(x) = \frac{1}{1+\exp(-x)}$.

The output in the output layer can be calculated using the following steps:

$$z_l = \sum_{i=0}^{M} w_{jl} h_j, \quad l = 1,2, \dots, L\,, j = 1, 2, \dots, M \tag{5}$$

$$y_l = f(z_l), l = 1, 2, \dots, L \tag{6}$$

Where z_l is the activation value of the lth node in the output layer; w_{jl} is the weight connected between the hidden node j and the output node l; y_l is the output in the output layer; f is a sigmoid function.

The number of hidden layer nodes are selected by experimenting different numbers until the suitable number that provides the best training performance is attained. In this study, 12 hidden layer nodes are chosen as it results in the best forecasting accuracy.

E. Support Vector Regression

Support vector machine (SVM) is a supervised machine learning algorithm for classification applications. A trick known as the kernel trick is used to transfer the classes into a higher dimensional space where they can be separated linearly. The basic kernel includes linear, polynomial and radial basis functions [27]. Unlike SVM, support vector regression (SVR) conducts its classification based on the regression errors that are greater or less than a particular threshold as shown in Fig.7 [7].

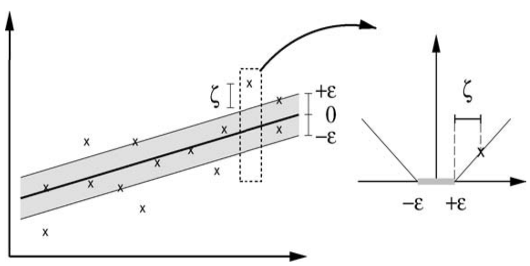

Fig. 7. The boundary margin for a linear SVR [7]

For a known set of training data (x_i, y_i), SVR aims at obtaining a linear function $f = w.x + b$ that solves the following optimization problem Eq.(7).

$$minimize \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*)$$
$$subject\ to \begin{cases} y_i - (w^T\phi(x_i) + b) \le \varepsilon + \xi_i{}^* \\ (w^T\phi(x_i) + b) - y_i \le \varepsilon + \xi_i \\ \xi_i, \xi_i^* \quad\quad \ge 0 \end{cases} \tag{7}$$

The constant $C > 0$ is the penalty factor; w is a normal vector; b, ξ_i *and* ξ_i^* are slack variables; ε is the problem threshold; and (x_i, y_i) is the training pair set.

The data behavior is sometime cannot be captured by applying only the linear regression. Therefore, the nonlinearity can be obtained by applying Kernel. The input space (training sup-space) is mapped into a higher dimensional space by function ϕ, which is the kernel trick $K(x_i, x_j) = \phi(x_i)^T(x_j)$. For kernel functions, radial basis (RB) and linear (linear) are used in this paper as shown in Eq.(8)-(9) [28]:

$$(RB): \quad K(x_i, x_j) = e^{-\gamma\left(\|x_i - x_j\|^2\right)} \tag{8}$$

$$(Linear): \quad K(x_i, x_j) = (x_i^T x_j) \tag{9}$$

Where γ $(Gamma)$ is the width of the RB function.

The selection of the two parameters, C and γ, of the SVR model is very important in improving the model forecasting accuracy. The parameter C controls the level of empirical risk of SVR. On the other hand, the parameter γ controls the width of the radial basis function [19]. Hence, both parameters influence the performance of SVR model. Researchers typically select these parameters based on their experience, previous knowledge [24] or using methods such as grid search [7]. In this study, both the penalty factor, C, and gamma, γ, are optimized by using Cuckoo Search (CS) and Differential Evolution (DE) algorithms. This is described in the next Section.

III. Intelligent Optimization Algorithms

The meta-heuristic optimization algorithms, including differential evolution (DE) and cuckoo search (CS), used in this study to optimize the SVR parameters are described as follows:

A. Differential Evaluation Algorithm

Differential evaluation algorithm (DE) is a meta-heuristics optimization technique and is a population-based algorithm which contains three major steps, namely: mutation, crossover and selection. The most significant feature that distinguishes DE from other Evolutionary Algorithms (EA), such as Genetic Algorithm (GA), is that DE's searching process depends on the information of current population distance and direction [24].

Details of the DE algorithm are as follows [24]:

Step 1: Initialize DE algorithm parameters which are population size (N), the mutation factor (F), the crossover rate (C), the maximum generations number g, the length of the generation (Chromosome) D and the upper (up) and lower (low) bounds of the searching space for parameters C and γ. In this study, N=100, F=0.5, C=0.9; and 100 is number of iterations. For the linear function, the up and low bounds for C are between [1,10,000] and for the RB function, the bounds are in the range of [1,10,000] and [0.01,3] for C and γ, respectively.

Step 2: Randomly generate an $N \times D$ matrix with a uniform distribution using the following formula:

$$X_{mn} = low[n] + rand() \times (Upp[n] - low[n]) \tag{10}$$

Where $m = 1, 2, \dots, N$; $n = 1, 2, \dots, D$; $rand()$ is a generated random number; and $Upp[n]$, $low[n]$ are the upper bound and lower bounds of the nth column, respectively.

Step 3: Evaluate the fitness values of all individuals in the population using the objective function. Mean Absolute Percentage Error $(MAPE)$ as in Eq. 16 is the objective function to be minimized in this study.

Step 4: Select two random vectors (X_b, X_c) to create a mutant vector X_a' from the following equality:

$$X_a' = X_a = F(X_b - X_c) \tag{11}$$

Where $a, b, c \in \{1, 2, \dots, N\}$ are randomly chosen and should be kept different from each other.

Step 5: Diversify the population by applying the crossover population through the following equation:

$$\begin{cases} X_b'(n) = X_a'(n) & if\ rand(n) \le C\ or\ n = randn(n) \\ X_b'(n) = X_a(n) & otherwise \end{cases} \tag{12}$$

Where n is the gene position of a chromosome; $randn(n)$ is also an integer number generated randomly within the range of $[1, D]$.

Step 6: Select the best offspring to represent the next generation based on their fitness values. In the case the fitness value of the offspring $f(R_{m,G})$ is better than the quality of parent's fitness value $f(X_{m,G})$, the offspring $R_{m,G}$ would move to the next generation; otherwise, the parent $X_{m,G}$ would continue.

$$X_{m,G+1} = \begin{cases} R_{m,G} & if\ f(R_{m,G}) < f(X_{m,G}) \\ X_{m,G} & otherwise \end{cases} \quad (13)$$

B. Cuckoo Search Algorithm

Cuckoo search (CS) is a meta-heuristic optimization algorithm stimulated from observing the life style of Cuckoo birds in their survival strategy along with the behavior of Lévy flight. Cuckoos laid their eggs in other birds' nests called host birds. If host birds discover cuckoo eggs, they may throw cuckoo eggs out or abandon their nests and live in a new location. In the CS algorithm, the egg in a certain nest represents a solution to the algorithm, while a new solution is represented by each cuckoo egg. A set of solution can be represented by a nest that has many eggs.

The CS algorithm is mainly idealized based on the following three assumptions [29]: (1) One egg is laid by each Cuckoo at a time and the nest to hatch is selected randomly; (2) The nests with high quality eggs (solutions) are continued to be the next Cuckoo generation; and (3) The available host nests are set to be fixed and the host bird probability to detect the Cuckoo egg is $p \in [0,1]$. In this case, the egg can be thrown by the host bird or the host bird leave the entire nest and rebuild a new nest somewhere else.

The Cuckoos' location is updated based on the Lévy flight, by using the following formula:

$$x_i^{(t+1)} = x_i^{(t)} + \partial \oplus levy(\beta) \quad (14)$$

Where $\partial > 0$ is the scale factor on which the direction and the step size depend. $\partial = 1$ is used in most cases; $\oplus$ means entry wise multiplication. Lévy flight is a random walk while the random steps are generated by a levy distribution:

$$levy{\sim}\mu = t^{(-1-\beta)} \quad ,(0 \le \beta \le 2) \quad (15)$$

Main steps of the CS algorithm are as follows [29]:

Step 1: Initialize the algorithm parameters: host nest (h), problem dimension (D), discovery probability by the host nest (p), number of iterations $(Iter)$, minimum and maximum step sizes of the random walk and the boundary of search space, lower (low) and upper (up) bound. In this study: h= 20, p=0.25 and 100 is number of iterations. For Linear function, the up and low bounds for C is between [1,10,000] and for RB function, the boundary are in the range of [1,10,000] and [0.01,3] for C and γ, respectively.

Step 2: Evaluate the fitness of each Cuckoo by using the objective function. Mean Absolute Percentage Error $(MAPE)$ as in Eq.16 is the objective function to be minimized in this study.

Step 3: A new position is generated for each Cuckoo in the next generation by using Lévy flight (Eq. 14).

Step 4: Compare the locations of new Cuckoos' generation with their parents. If the fitness of new generation $f(x_i^{(t+1)})$ is better than the previous generation $f(x_i^{(t)})$, the new Cuckoo $x_i^{(t+1)}$ is selected, otherwise, the parent $x_i^{(t)}$ would retain.

Step 5: Compare the probability of discovery p with a new number u generated uniformly from [0,1]. In case, $u > p$, the new nests are created and the worst nests are abandoned. These new nest locations should be evaluated and compared with the solution attained in Step 4. The optimal nest locations are selected accordingly.

Step 6: Check if the stopping conditions are satisfied; otherwise, the algorithm should return to step 2.

Step 7: The optimum nest position represents the optimal solution of the problem (best SVR parameters).

IV. Model Accuracy Criteria

The model accuracy is evaluated based on several statistical indicators: Mean Absolute Percentage Error $(MAPE)$, Root Mean Squared Error $(RMSE)$ and Goodness of Fit (R^2). They are expressed by the following equations [11]:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|f_i - y_i|}{y_i} \times 100\% \quad (16)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f_i)^2} \quad (17)$$

$$R^2 = \frac{\sum_{i=1}^{n}(f_i - \tilde{y})^2}{\sum_{i=1}^{n}(y_i - \tilde{y})^2} \quad (18)$$

where n is the number of dataset involved in the analysis; y_i is the actual value to be estimated; $\tilde{y}$ is the mean value of y_i; and f_i is the forecasted value.

MAPE determines the model accuracy, and RMSE measures the deviation between actual values and forecasted values [11]. R^2 is the proportion of variation of the forecasted values generated by a model and the sample data variation.

V. Results and Discussions

The proposed methods were applied to forecast PV power output at the Virginia Tech Advanced Research Institute (VT-ARI) building in Arlington, Virginia. In this study, the forecasting approaches used were as follows:

- SVR based on RB function with CS (CS-RB);
- SVR based on RB function with DE (DE-RB);
- SVR based on linear function with CS (CS-Linear);
- SVR based on linear function with DE (DE-Linear); and
- BPNN model

To conduct this study, both of the BPNN and SVR based kernel functions were implemented employing MATLAB R2017a and LIBSVM tools [30]. In the parameter selection process, input data were firstly normalized to minimize the numerical difficulties during searching the parameters. After that, CS and DE algorithms were applied to determine the best parameters of the SVR models. These parameters are $C\ and\ \gamma$ for RB function and C for linear function. During the algorithms, these parameters were evolved and their values were generated until the smallest testing MAPE values (Eq.16), which is the objective function in this study, were attained. These values became the optimal SVR parameters. Table I summarizes the optimal values of parameters $C\ and\ \gamma$ for this dataset with SVR based on RB and linear functions.

TABLE I. MODELS PARAMETERS FOR SVR MODELS WITH CS AND DE ($\varepsilon = 0.001$)

	C	**γ**
CS-RB	10000	0.143478
DE-RB	9945.612	0.143491
CS-Linear	1404.951	-
DE-Linear	1427.759	-
BPNN	Number of hidden Layers=1, hidden nodes=12 number of iterations=1000	

A multilayer perceptron (MLP) was selected for the BPNN model with the back propagation algorithm, while the Levenberg-Marquardt method was selected as the training function. By experimenting different configurations with MLP, one input layer, one output layer and one hidden layer are used since they provided the most accurate model performance. In this study, twelve neurons (nodes) were created for the hidden layer while the input and the output data were the same as those used in SVR models. Table I also summarizes the BPNN model configuration.

The SVR models with the optimal parameters and BPNN with the selected number of layers were then used to predict the PV power output. The performance of the proposed models was evaluated to determine how close the predicted data track the measured data. For this purpose, the output of the testing models was examined based on Mean Absolute Percentage Error ($MAPE$), Root Mean Squared Error ($RMSE$) and Goodness of Fit (R^2). Table II compares the performance indices of SVR and BPNN models.

TABLE II. PERFORMANCE INDICES OF THE TESTING DATASET

	MAPE (%)	RMSE (kW)	$\mathbf{R^2}$
CS-RB	8.67	0.137	0.9908827
DE-RB	8.68	0.137	0.9908823
BPNN	9.93	0.141	0.9904398
CS-Linear	17.65	0.235	0.973418
DE-Linear	17.65	0.235	0.973417

MAPE is expressed in percentage of the relative error between actual and forecasted values, while $RMSE$ is expressed in kW value. For example, the SVR radial basis with CS has RMSE of 0.137kW (or 2.7% of 5.04 kW the peak power of the testing set). The results indicate that the proposed two SVR models with RB function have the best performance compared to BPNN model and the two SVR models with linear function.

According to Table I, the SVR with RB, the two optimization algorithms, CS and DE, almost have the same performance in selecting the optimal SVR parameters. In term of forecasting with the chosen parameters (Table II), SVR with RB displays satisfactory performance. For the SVR with CS-RB and DE-RB models, MAPE is around 8.67%, which is the smallest MAPE among all forecasting models.

BPNN, on the other hand, has a better performance than the two SVR models with linear function and promising performance compared to RB models with MAPE of 9.93%.

Both CA and DE algorithms have almost the same performance with linear function; however, linear models performed the worst due to their limited capacity in dealing with nonlinearity in input data.

For better visualization of SVR models and BPNN model, measured PV output power is plotted against the predicted values, as shown in Fig. 8-10. These figures confirm that SVR models with RB perform the best in predicting the PV output power at the study site. BPNN shows also acceptable prediction accuracy in a way that can compete with the RB based SVR models.

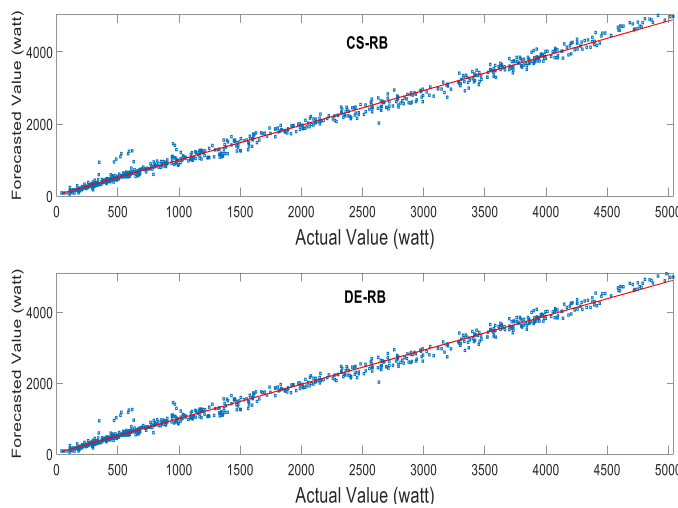

Fig. 8. Measured vs forecasted PV output using RB based SVR models with CS (top) and DE (bottom).

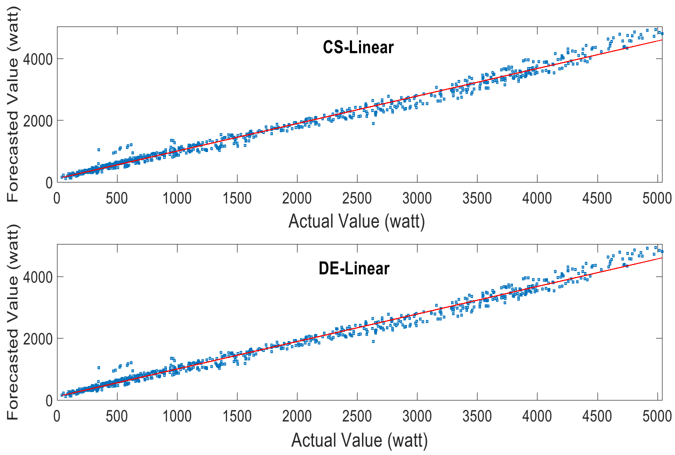

Fig. 9. Measured vs forecasted PV output using SVR Linear Basis Models with CS (top) and DE (bottom).

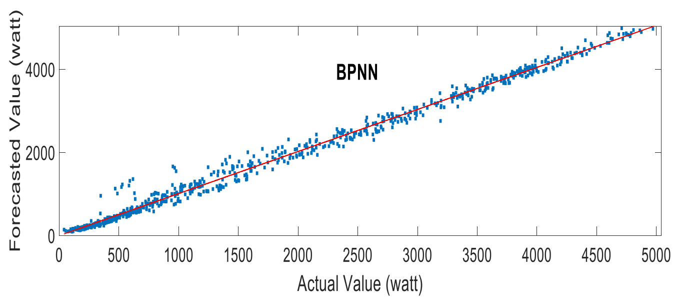

Fig. 10. Measured vs forecasted PV output using BPNN model.

Finally, for further visualization, Fig. 11 shows a comparative forecasting of PV power output by all models examined in this study for twenty days during October 6th-26th. Furthermore, Fig. 12 and Fig.13 exhibit the forecasting results on October 10th (high power output) and on November 22nd (low power output), respectively. From these figures, the radial basis SVR based kernel function models show the ability to track the actual PV power profile.

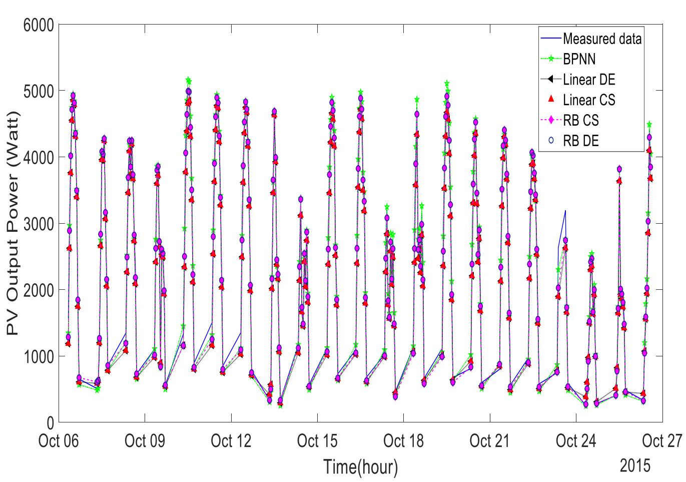

Fig. 11. Forecasting results of different models.

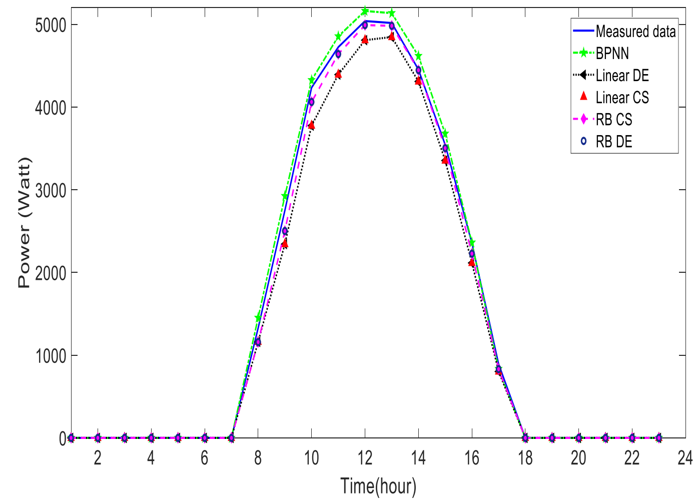

Fig. 12. Forecasting results on October 10th.

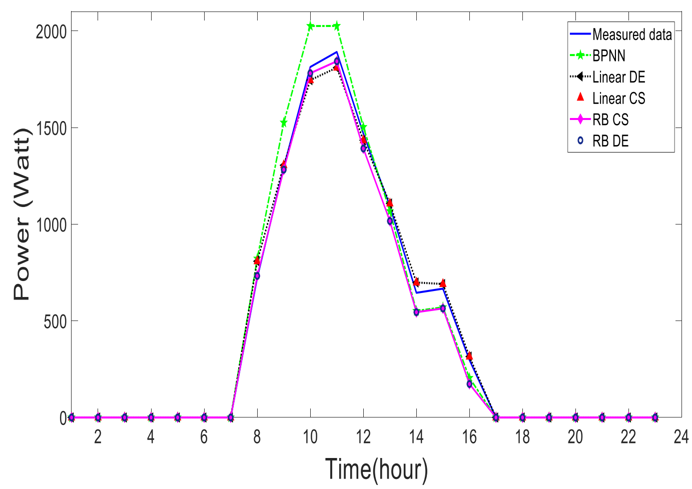

Fig. 13. Forecasting results on November 22nd.

VI. CONCLUSION

In this paper, SVR with radial basis and linear kernel functions models were examined to predict the PV output power of the rooftop PV unit at the Virginia Tech Advanced Research Institute (VT-ARI). The penalty factor (C) and kernel parameter (γ) of the SVR models with radial and linear functions were optimized using Cuckoo Search (CS) and differential evolution (DE) algorithms. Back propagation neural network was used also for comparison. Six input variables, including month, day, hour, global solar radiation, temperature and wind speed, were used as inputs to the models for predicting PV output power. The efficiency of all selected models was evaluated using Mean Absolute Percentage Error ($MAPE$), Root Mean Squared Error ($RMSE$) and Goodness of Fit (R^2).

Results indicate that the SVR with radial basis outperforms other models in forecasting PV power output at the tested site. CS and DE algorithms almost have the same performance and both result in high accuracy models which prove their ability to select SVR parameters. Furthermore, the BPNN model exhibits a good performance that can compare with the SVR radial basis models.

Acknowledgment

The authors, Musaed Alrashidi and Massoud Alrashidi, would like to thank Qassim University, Saudi Arabia, for the financial support in the form of funded educational scholarships.

References

[1] E. M. Sandhu and T. Thakur, "Issues, Challenges, Causes, Impacts and Utilization of Renewable Energy Sources -Grid Integration," *J. Eng. Res. Appl. www.ijera.com ISSN*, vol. 4, no. 1, 2014.

[2] "MIT Energy Initiative Symposium on Managing Large-Scale Penetration of Intermittent Renewables | Managing Large-Scale Penetration of Intermittent Renewables An MIT Energy Initiative Symposium," 2011.

[3] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-De-Pison, and F. Antonanzas-Torres, "Review of photovoltaic power forecasting," 2016.

[4] L. Olatomiwa *et al.*, "Potential of support vector regression for solar radiation prediction in Nigeria," *Nat Hazards*, vol. 77, pp. 1055–1068, 2015.

[5] Z. Ramedani, M. Omid, A. Keyhani, S. Shamshirband, and B. Khoshnevisan, "Potential of radial basis function based support vector regression for global solar radiation prediction," 2014.

[6] F. Wang, Z. Zhen, Z. Mi, H. Sun, S. Su, and G. Yang, "Solar irradiance feature extraction and support vector machines based weather status pattern recognition model for short-term photovoltaic power forecasting," *Energy Build.*, vol. 86, pp. 427–438, 2015.

[7] M. Abuella and B. Chowdhury, "Solar Power Forecasting Using Support Vector Regression." in *Proceedings of the American Society for Engineering Management 2016 International Annual Conference, 2016.*

[8] R. Xu, H. Chen, and X. Sun, "Short-term photovoltaic power forecasting with weighted support vector machine," *2012 IEEE Int. Conf. Autom. Logist.*, no. August, pp. 248–253, 2012.

[9] S. Leva, A. Dolara, F. Grimaccia, M. Mussetta, and E. Ogliari, "Analysis and validation of 24 hours ahead neural network forecasting of photovoltaic output power," *Math. Comput. Simul.*, vol. 131, pp. 88–100, 2017.

[10] A. Azadeh, A. Maghsoudi, and S. Sohrabkhani, "An integrated artificial neural networks approach for predicting global radiation," *Energy Convers. Manag.*, vol. 50, pp. 1497–1505, 2009.

[11] C. Renno, F. Petito, and A. Gatto, "Artificial neural network models for predicting the solar radiation as input of a concentrating photovoltaic system," 2015.

[12] L. Liu, D. Liu, Q. Sun, H. Li, and R. Wennersten, "Forecasting Power Output of Photovoltaic System Using A BP Network Method," *Energy Procedia*, vol. 142, pp. 780–786, 2017.

[13] J. Shi, W. J. Lee, Y. Liu, Y. Yang, and P. Wang, "Forecasting power output of photovoltaic systems based on weather classification and support vector machines," *IEEE Trans. Ind. Appl.*, vol. 48, no. 3, pp. 1064–1069, 2012.

[14] J. Gari da Silva Fonseca Jr, T. Oozeki, T. Takashima, G. Koshimizu, Y. Uchida, and K. Ogimoto, "Use of support vector regression and numerically predicted cloudiness to forecast power output of a photovoltaic power plant in Kitakyushu, Japan."

[15] H. Y. Cheng, C. C. Yu, and S. J. Lin, "Bi-model short-term solar irradiance prediction using support vector regressors," *Energy*, vol. 70, pp. 121–127, 2014.

[16] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting Solar Generation from Weather Forecasts Using Machine Learning."

[17] S. Sperati, S. Alessandrini, P. Pinson, and G. Kariniotakis, "The 'Weather intelligence for renewable energies' benchmarking exercise on short-term forecasting of wind and solar power generation," *Energies*, vol. 8, no. 9, pp. 9594–9619, 2015.

[18] L. M. Saini, S. K. Aggarwal, and A. Kumar, "Parameter optimisation using genetic algorithm for support vector machine-based price-forecasting model in National electricity market," *IET Gener. Transm. Distrib.*, vol. 4, no. 1, p. 36, 2010.

[19] W.-C. Hong, "Electric load forecasting by support vector model," *Appl. Math. Model.*, vol. 33, pp. 2444–2454, 2008.

[20] W.-C. Hong, "Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model," *Energy Convers. Manag.*, vol. 50, pp. 105–117, 2008.

[21] R. Storn and K. Price, "Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[22] X.-S. Yang and S. Deb, "Cuckoo Search via Levy Flights," 2010.

[23] I. Fister, D. Fister, and I. Fister, "A comprehensive review of cuckoo search: variants and hybrids," *Int. J. Math. Model. Numer. Optim.*, vol. 4, no. 4, p. 387, 2013.

[24] J. Wang, L. Li, D. Niu, and Z. Tan, "An annual load forecasting model based on support vector regression with differential evolution algorithm," *Appl. Energy*, vol. 94, pp. 65–70, 2012.

[25] J. Wang, Q. Zhou, H. Jiang, and R. Hou, "Short-Term Wind Speed Forecasting Using Support Vector Regression Optimized by Cuckoo Optimization Algorithm," vol. 2015, p. 619178, 2014.

[26] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification."

[27] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning," *The Mathematical Intelligencer*, vol. 27, no. 2. pp. 83–85, 2001.

[28] A. J. Smola and B. Sc Olkopf, "A tutorial on support vector regression *," *Stat. Comput.*, vol. 14, pp. 199–222, 2004.

[29] X.-S. Yang and S. Deb, "Engineering Optimisation by Cuckoo Search," pp. 1–17, 2010.

[30] C.-C. Chang and C.-J. Lin, "Libsvm," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.