Prediction of Photovoltaic Power Using Nature-Inspired Computing

Miroslav Sumega(✉), Anna Bou Ezzeddine, Gabriela Grmanová,
and Viera Rozinajová

Faculty of Informatics and Information Technologies,
Slovak University of Technology in Bratislava,
Ilkovičova 2, 842 16 Bratislava, Slovakia
miro.sumega@gmail.com
{anna.bou.ezzeddine,gabriela.grmanova,viera.rozinajova}@stuba.sk

Abstract. Prediction of photovoltaic (PV) energy is an important task. It allows grid operators to plan production of energy in order to secure stability of electrical grid. In this work we focus on improving prediction of PV energy using nature-inspired algorithms for optimization of Support Vector Regression (SVR) models. We propose method, which uses different models optimized for various types of weather in order to achieve higher overall accuracy compared to single optimized model. Each sample is classified by Multi-Layer Perceptron (MLP) into some weather class and then model is trained for each weather class. Our method achieved slightly better results compared to single optimized model.

Keywords: Firefly Algorithm · Optimization · Support Vector Regression

1 Introduction

Renewable sources of energy are increasingly involved in total energy production. One of the most important sources of renewable energy is solar radiation. PV panels are used in order to obtain energy from solar radiation. However this type of energy can be unstable, resulting in large fluctuations of energy production which might cause instability of electrical grid. Therefore, it is necessary to predict the output of these power plants so that grid operators can plan power generation or effectively regulate the grid to ensure its stability.

Various approaches to prediction of PV energy are used. According to [1] there are 3 types of approaches: physical, statistical and hybrid. Physical approaches use technical parameters of PV power plants and weather forecasts. Statistical approaches use only data from the past, which contains information about weather and production of PV power plant. Statistical methods are further subdivided into regression and artificial intelligence methods, which are able to use these data to create prediction models. Hybrid approaches combine previous approaches to the ensembles to improve prediction.

Y. Tan et al. (Eds.): ICSI 2020, LNCS 12145, pp. 25–36, 2020.
https://doi.org/10.1007/978-3-030-53956-6_3

Artificial intelligence methods are powerful in predicting PV power production, but their accuracy is highly dependent on their hyperparameter setting. The hyperparameter setting can be done in various ways, either manually or by using algorithms capable of finding and evaluating different hyperparameter settings. A group of algorithms used for hyperparameter setting is called nature-inspired algorithms. These algorithms are able to avoid local minima and find global minimum. Many of these algorithms use large amounts of agents representing specific solutions. Firefly algorithm (FA), particle swarm optimization (PSO) and genetic algorithm (GA) are some of nature-inspired algorithms.

In this paper artificial intelligence approach is used. We are using SVR for predictions. In order to increase prediction accuracy we are using FA to optimize SVR hyperparameters. We also classify each sample with MLP and we train multiple SVR models, one for each weather type.

2 Related Work

There are many different approaches to prediction of PV power in the literature.

Multiple SVR models were used for different weather types, which were obtained with SOM and LVQ, were used in [16]. In [9] comparison of ANN, kNN, SVM and MLR was done. Simple parameter optimization was performed for each algorithm. Multiple weather types were also used in [12]. In [6] ten different optimized machine learning algorithms were used for predicting. Various algorithms were also used in [14], specifically FFNN, SVR and RT. Parameter optimization was done for each algorithm. Classifying weather into weather types with SOM was used in [2]. For each weather type one model of RBF network was trained. SVR and ensemble of NN were used in [11]. They also used CFS for feature selection. In [10] they compared accuracy of SVR to accuracy of physical model. In [13] they used GBDT with Taylor formula for predictions and compared it to original data and prediction of optimized SVM with RBF kernel. Different approach was used in [3]. They used v-SVR with parameter optimization. In order to achieve best results, model was retrained each night. MARS was used for predictions in [8], where it was compared to multiple different algorithms. In [7] ELM, ANN and SVR were used. MLP, LSTM, DBN and Auto-LSTM neural networks along with physical P-PVFM model were used in [5] for prediction of PV power production of 21 power plants. In [15] multiple physical and SVR models were used for various time intervals for 921 power plants. SVR was optimized using GridSearch to obtain higher accuracy.

3 Firefly Algorithm

Firefly algorithm [17] is a metaheuristic inspired by firefly behaviour in nature. Idea of this algorithm is that each firefly represents one solution of optimized problem. All fireflies move towards other fireflies they see according to movement equation. Since fireflies represent solutions, change of position of firefly also means change of solution.

We use firefly algorithm to optimize SVR models. In our case firefly represents model hyperparameters which change when firefly moves. SVR hyperparameters we optimized using FA are C, ϵ, γ and tolerance for stopping criterion.

We chose FA because its parameters α, β_0 and γ, which are described in following subsection, allow great control over optimization process. Several experiments were performed to find the best setting of those parameters.

3.1 Movement Equation

In our implementation, each firefly moves according to following equation:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha \epsilon_i^t \delta^t \tag{1}$$

where x_i^{t+1} is new position of a firefly, x_i^t is actual position of a firefly. Attractivity coefficient β_0 determines how fast fireflies move towards each other. Visibility coefficient γ is used to change perceived attractivity of fireflies. α is random movement coefficient, which decreases with every generation, ϵ_i^t is vector of random numbers representing random movement of firefly and δ^t is vector of coefficients used for changing range from which random movement is generated.

4 Methods of Prediction

We are using two methods of prediction. Both of our methods are based on SVR [4], which is regression method based on an idea of Support Vector Machine. SVR utilizes hyperplane that maximizes margins of tolerance for data points while tolerating some error. In case data are not linear, SVR also uses kernel functions to transform them to linear feature space.

First method is single SVR model optimized on entire training data set. Second method (Fig. 1) is based on multiple models of SVR with each model optimized on specific weather class. Weather class is a numerical representation of weather type (sunny, cloudy, etc.). We use combination of clustering and classification to obtain weather classes.

4.1 Weather Classes Discovery

First step to discover weather classes is obtaining of initial weather class labels from training data set. We are using agglomerative clustering to obtain labels. Each cluster that agglomerative clustering discovers is considered a unique weather class. Before clustering is started, we specify the number of initial classes it should discover. Initial classes represent the first division of samples according to weather. After initial weather classes are discovered, we train MLP classifier so we can use it to classify new samples.

With weather classes discovered, we use FA to optimize one SVR model for each class. Then accuracy of SVR for each weather class is compared to accuracy of the first method on that class. After accuracy of models of all weather classes

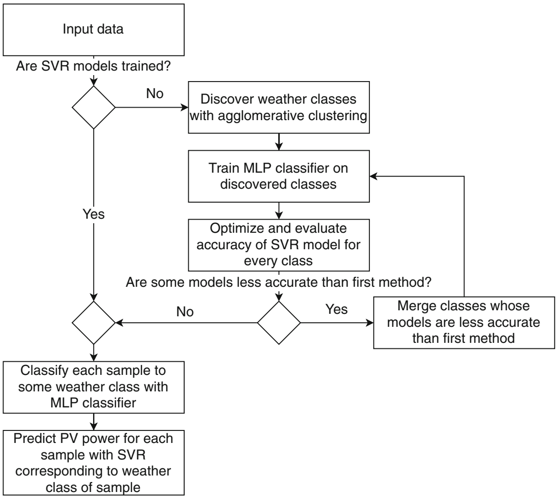
Input data
Are SVR models trained?
No
Discover weather classes with agglomerative clustering
Train MLP classifier on discovered classes
Yes
Optimize and evaluate accuracy of SVR model for every class
Are some models less accurate than first method?
No
Yes
Merge classes whose models are less accurate than first method
Classify each sample to some weather class with MLP classifier
Predict PV power for each sample with SVR corresponding to weather class of sample

Fig. 1. Diagram showing how method based on weather classes works.

is checked, all classes whose SVR performed worse compared to the first method are merged. Obtained weather class should be more similar to the whole training data set than any of the classes that were merged together, therefore model optimized for this new class should perform more similarly to the first method.

After merging, SVR model is optimized for new weather class and MLP classifier is retrained. Then accuracy of prediction on all classes is checked again. Because of merging of classes and retraining of classifier, some samples might be classified into different classes than before. This might cause that models of some classes, which were better before merging, are now worse that the first method. Then merging of classes and optimization will happen again. This process of optimization, accuracy evaluation and merging of classes repeats in a cycle while there are at least two weather classes to merge. When cycle ends, we have final weather classes and SVR models, which we can now use for prediction.

4.2 Use of Multiple Weather Classes for Prediction

We use MLP classifier to obtain weather classes. This classifier can predict probability that sample belongs to a specific weather class. We use these probabilities to improve accuracy of a prediction according to the following equation:

$$X = \sum_{i=0}^{n} p_i x_i \tag{2}$$

where X is final prediction, x_i is prediction if sample belongs to weather class i, p_i is probability of sample belonging to weather class i and n is a number of weather classes. When predicting, we first obtain probabilities of sample belonging to specific weather classes. Then for each weather class we make prediction with its SVR model and multiply it by probability of sample belonging to that weather class. Sum of all augmented predictions is considered as the final prediction.

4.3 Bias Correction

Since machine learning models might be slightly biased if not trained perfectly, we decided to use simple bias correction for all models in order to decrease prediction error. To perform bias correction, we first evaluate Mean Bias Error according to Eq. 8 on validation data set. Then from simple equation:

$$coef = 1 - \frac{MBE}{R} \tag{3}$$

where *MBE* and *R* are described in Sect. 6.1, we obtain bias coefficient which we use to correct bias of prediction. This correction is performed by multiplicating predicted values with obtained bias coefficient.

5 Data

In our experiments we used data set from University of Queensland[1]. This data set has one minute resolution, but we aggregated it to higher resolution depending on what series of experiments we were performing. Data sets contain following attributes: air temperature, humidity, wind speed and direction, insolation, power production in watts (W) and timestamp.

For the first and second series of experiments we used data from UQ Centre from 1.1.2014 to 31.12.2017 and aggregated it to hourly resolution. Time interval of data we used was 5 am to 7 pm. Data from years 2014 and 2015 were used for training, data from year 2016 were used for validation and data from year 2017 were used for testing.

In order to compare our results to [11], we used the same subset of data, therefore data were only from years 2013 and 2014 from 7 am to 5 pm and we aggregated it to 5 min resolution. In case of insolation and power we used addition to aggregate them. Other attributes were aggregated as mean hourly values. Training data were from year 2013. As validation data we chose every other day from year 2014 starting with 2nd January. Test data were chosen in the same way as validation data, however it started with 1st January.

In all experiments training and validation data were used in optimization process and test data were used to evaluate accuracy of optimized models.

5.1 Data Preprocessing

We transformed production to kilowatts (kW) and we also extracted minute (for third series of experiments), hour, day, month and year for each sample

[1] https://solar-energy.uq.edu.au/research/open-access-data.

from timestamp. We also derived weather changes in last hour for the first and second series of experiments and in last 55 min for third series of experiments. We also scaled power production and all attributes used for prediction.

For each sample we also used hourly production from last 3 h for the first and second series of experiments and last 6 h for third series of experiments. In some experiments we used also production from the most similar sample in entire data set if the first method was used or only in specific weather class if second method was used. We checked for similar production only in samples where absolute hour difference between original and similar sample is not larger than 1. This difference in case of months was set to 2. We decided to use those limits because production difference between those limits is not too large.

6 Experiments

We made three series of experiments. In the first series we focused on finding a good setting of FA. In the second series we used our methods of prediction to predict hour ahead PV production and in the third series we compared our approach with existing approach.

6.1 Evaluation Metrics

To evaluate accuracy of both of our methods of prediction, we use Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and their percentage transformations: normalized RMSE (nRMSE) and Mean Relative Error (MRE). Because we are using simple bias correction, we also use Mean Bias Error (MBE) to obtain bias. Following are formulas used for calculation of errors:

$$MAE = \frac{1}{N} * \sum_{i=1}^{N} |x_i - y_i| \tag{4}$$

$$MRE = 100\% * \frac{MAE}{R} \tag{5}$$

$$RMSE = \sqrt{\frac{1}{N} * \sum_{i=1}^{N} (x_i - y_i)^2} \tag{6}$$

$$nRMSE = 100\% * \frac{RMSE}{R} \tag{7}$$

$$MBE = \frac{1}{N} * \sum_{i=1}^{N} (x_i - y_i) \tag{8}$$

where x_i is predicted value, y_i is real value, N is number of samples and R is computed as a difference between maximal and minimal power production in training data set. In case of predictions for one hour ahead we used largest value in training data set where $R = 21856.645$ kW and in case of predictions

Table 1. Experiments to find good settings of FA.

α	β_0	γ	RMSE	MAE	Best generation	Scattering
0	0.2	0	2229.0	1258.8	1	None
0	0.2	1	2579.3	2230.5	0	Very small
0	0.2	2	1983.5	1382.0	0	Very small
1	0	0	2043.7	1397.7	18	Medium
1	**1**	**0**	**1920.5**	**1221.7**	**19**	**Very small**
1	2	0	1972.1	1258.8	6	Small
1	1	1	1921.9	1223.5	30	Very small
1	1	2	1958.3	1316.8	29	Big
1	2	1	1931.4	1230.5	15	Small
1	2	2	2061.5	1451.5	27	Big

for 55–60 min ahead interval we used largest value in entire dataset (training, validation and test) $R = 1150.27$ kW because same approach was used in solution with which we compare our methods.

In all experiments metrics RMSE and MAE are in kW and metrics nRMSE and MRE are in %.

6.2 Experiments with Settings of Firefly Algorithm

In this series of experiments we tried various settings of FA to find the most suitable setting we could use in further experiments. We investigated the impact of parameters α, β and γ described in Subsect. 3.1 on the speed of finding the best solution in that run (column *Best generation*) and how scattered were fireflies after last generation. This series of experiments were performed on first method of prediction which used only current weather to forecast hour ahead production. For each experiment we used 15 fireflies and 30 generations.

We must note that data used in these experiments were later slightly changed and therefore model performances are slightly different compared to other experiments. However we did not run these experiments again because we could use the results to decide which setting is most suitable for further experiments.

We can see in Table 1 that when $\alpha = 0$ firefly algorithm was not able to find good hyperparameter settings of SVR model, but very small scattering was achieved. We assume that this is because there was no random movement, therefore fireflies moved directly towards each other. We can also see that scattering is smaller when value of γ is smaller. This happens because smaller values of γ mean better visibility. When $\beta_0 = 0$ movement is completely random because β_0 controls attractivity of fireflies. Otherwise there does not seem to be any significant influence of β_0 on optimization.

We chose settings where $\alpha = 1$, $\beta_0 = 1$ and $\gamma = 0$ for further use because of the small spread after last generation and also because best solution was not found too late nor too early.

Table 2. Results of experiments with first method. In column *Used attributes* value 1 represents only current weather, value 2 represents weather change in last hour, value 3 represents measured production in 3 previous hours and value 4 represents measured production from most similar sample. Last row (in italic) is SVR with default parameters and best attributes.

Used attributes	Single model		Single model with bias correction	
	RMSE (nRMSE)	MAE (MRE)	RMSE (nRMSE)	MAE (MRE)
1	1968.2 (9.01)	1256.7 (5.75)	1954.2 (8.94)	1267.7 (5.80)
1, 2	2043.5 (9.35)	1354.0 (6.19)	2037.9 (9.32)	1379.9 (6.31)
1, 3	1511.1 (6.91)	948.1 (4.34)	1510.1 (6.91)	947.7 (4.34)
1, 4	1916.3 (8.77)	1220.6 (5.58)	1906.6 (8.72)	1237.1 (5.66)
1, 2, 3	1477.2 (6.76)	932.9 (4.27)	1475.0 (6.75)	934.8 (4.28)
1, 2, 4	1959.5 (8.97)	1380.5 (6.31)	1967.6 (9.00)	1403.3 (6.42)
1, 2, 3, 4	**1464.0 (6.70)**	**888.7 (4.07)**	**1461.9 (6.69)**	**890.2 (4.07)**
1, 2, 3, 4	*1498.3 (6.86)*	*952.9 (4.36)*	*1496.1 (6.84)*	*956.9 (4.38)*

6.3 Experiments with Hour Ahead Prediction

In this series of experiments we used various features for prediction of hour ahead production of PV power. We grouped those features into four sets: current weather, weather change in last hour, power production for last three hours, power production from most similar sample in the past.

Single Model Experiments. In order to decide which attributes are most suitable for second method, we evaluated accuracy of the first method on multiple combinations of attributes. We include prediction with and without bias correction for comparison. For each experiment we used 15 fireflies and 50 generations.

Best results in Table 2 were obtained when previous power production was used. Using weather changes also improved results when it was used along with current weather and previous production. However, when used only with current weather, trained model was less accurate. Similar production improved accuracy in most cases except one, where MAE of trained model was higher compared to model trained on same attributes but without similar production.

In case of bias correction, we evaluated every model with and without bias correction. We noticed that when using bias correction, RMSE tends to be smaller compared to RMSE without bias correction, however MAE tends to increase slightly. We think this happened because bias correction flattened high errors, but increased overall error.

We have added SVR with default hyperparameters and best attributes to show that optimization helped us to improve results. It is best seen when comparing the best model (in bold) with default SVR on MAE metric. Other models were less accurate than default, but it is because of attributes.

Table 3. The attributes current weather and previous production were used. Initial number of weather classes was 2. No classes merged.

Method variations	RMSE (nRMSE)	MAE (MRE)
Basic	1500.8 (6.87)	914.7 (4.19)
Bias correction	1497.1 (6.85)	918.9 (4.20)
Multiclass	1498.9 (6.86)	914.1 (4.18)
Multiclass with bias correction	1495.3 (6.84)	918.3 (4.20)

Table 4. The attributes current weather, weather change and previous production were used. Initial number of weather classes was 5 and after merging 3.

Method variations	RMSE (nRMSE)	MAE (MRE)
Basic	1486.0 (6.80)	896.2 (4.10)
Bias correction	1484.6 (6.79)	900.1 (4.12)
Multiclass	1480.5 (6.77)	892.1 (4.08)
Multiclass with bias correction	1479.1 (6.77)	896.0 (4.10)

Multiple Model Experiments. In Tables 3, 4 and 5 are the best results of the experiments with the second method for each attribute combination. We used three best attribute combinations from experiments with first method. For each used combination of attributes we evaluated accuracy without any improvements, with bias correction, with multiclass prediction (Subsect. 4.2) and with combination of bias correction and multiclass prediction. For each experiment we used 10 fireflies and 20 generations.

We can see in Tables 4 and 5 that RMSE slightly increased compared to the single model experiments (Table 2) and in Table 3 that RMSE decreased. However in all cases MAE decreased.

Increase of RMSE means that some deviations from real values are larger compared to the first method and decrease of MAE means that overall deviations are smaller. Increase of RMSE might have happened because optimization of models for specific weather class did not achieve global optimum. Other reason might be that model could not be more accurate on given class because samples in a class were too different due to merging of classes.

Regarding improvement of accuracy of second method, we noticed that both bias correction and usage of multiple classes for prediction decreased RMSE. However bias correction increases MAE. Best results for MAE were achieved with usage of multiple classes, however combination of bias correction and multiple classes achieved smallest RMSE.

Bias correction has probably flattened high errors, but increased overall error as in single model experiments. We think multiclass predictions improved accuracy because it took into consideration that samples might be misclassified.

Table 5. The attributes current weather, weather change, previous and similar production were used. Initial number of weather classes was 25 and after merging 8.

Method variations	RMSE (nRMSE)	MAE (MRE)
Basic	1474.4 (6.75)	874.3 (4.00)
Bias correction	1473.1 (6.74)	878.7 (4.02)
Multiclass	1471.5 (6.73)	872.3 (3.99)
Multiclass with bias correction	1470.2 (6.73)	876.5 (4.01)

Table 6. Comparison of our solution with solution from [11]. Values of MAE and MRE for NN ensemble and SVR are taken from compared article. Single model represents first method and Multiclass with bias correction represents second method.

Method of prediction	MAE	MRE
NN ensemble	100.2	8.71
SVR	107.4	9.34
Single model	100.4	8.73
Multiclass with bias correction	102.0	8.87

6.4 Comparison with Existing Solution

In this series of experiments, we compared the best solutions of both our methods to the best solution from [11]. They also used data from University of Queensland, but from years 2013 and 2014 and from multiple buildings.

In order to obtain most accurate results, we tried to replicate data used in the mentioned solution. However we were not able to fully reproduce data they used and therefore results might have been slightly different as if data were identical.

They made predictions for every 5-min interval for next hour. We compared our solutions to theirs only on the last interval (55–60 min ahead). For experiments we used feature sets combinations for both methods where highest accuracy was acquired when predicting for one hour ahead. For both methods the best combination was current weather, weather changes, previous and similar production. In both experiments we used 10 fireflies and 20 generations.

In Table 6 we can see that first method has performance similar to ensemble of neural networks, but outperformed their SVR. Difference is that in our method SVR is optimized using FA and SVR from [11] does not seem to be optimized. Also we did not use same features. That might have caused better performance.

We can see that second method performed worse than first method. This probably happened because we had to change the application of second method due to high computational complexity of SVR. Instead of optimizing for various numbers of weather classes, models were only trained with optimal parameters obtained from the first method on the same data. Then we optimized models for best number of weather classes. As a result the optimal number of weather classes might not have been used. Another reason might be that models were not optimized enough to perform better.

7 Conclusion

In this paper, we proposed approach to prediction of PV power based on classifying samples into different weather classes and using FA to optimize model for each weather class.

We compared this approach to single SVR model optimized on entire training data set. Experiments show that our approach tends to decrease MAE compared to single model. We also compared our methods with [11]. We achieved similar accuracy with both of our methods, however the second method performed worse than expected. This is probably caused by the fact that we did not utilize FA optimization fully when comparing with [11]. From this and comparison of optimized and unoptimized single SVR model we conclude that optimization has visible impact on accuracy of predictions and we recommend using it.

Our approach has proven to have potential, however it still needs improvements. It might be improved by changing merging of weather classes from one large class into several smaller classes to avoid the problem of merging of two too different classes. Also optimization of classifier could result in more accurate assignment to classes and therefore better performance.

In the future we might also try different optimization algorithms to compare them with FA, however we do not expect any significant improvements from using different optimization algorithm.

Acknowledgements. This work was supported by the project "Knowledge-based Approach to Intelligent Big Data Analysis" - Slovak Research and Development Agency under the contract No. APVV-16-0213 and by the project "International Centre of Excellence for Research of Intelligent and Secure Information-Communication Technologies and Systems - phase II", No. ITMS: 313021W404, cofinanced by the European Regional Development Fund.

References

1. Antonanzas, J., Osorio, N., Escobar, R., Urraca, R., Martinez-de Pison, F., Antonanzas-Torres, F.: Review of photovoltaic power forecasting. Solar Energy **136**, 78–111 (2016). https://doi.org/10.1016/J.SOLENER.2016.06.069
2. Chen, C., Duan, S., Cai, T., Liu, B.: Online 24-h solar power forecasting based on weather type classification using artificial neural network. Solar Energy **85**(11), 2856–2870 (2011). https://doi.org/10.1016/J.SOLENER.2011.08.027
3. De Leone, R., Pietrini, M., Giovannelli, A.: Photovoltaic energy production forecast using support vector regression. Neural Comput. Appl. **26**(8), 1955–1962 (2015). https://doi.org/10.1007/s00521-015-1842-y
4. Drucker, H., Burges, C.J., Kaufman, L., Smola, A.J., Vapnik, V.: Support vector regression machines. In: Advances in Neural Information Processing Systems, pp. 155–161 (1997)
5. Gensler, A., Henze, J., Sick, B., Raabe, N.: Deep learning for solar power forecasting - an approach using AutoEncoder and LSTM neural networks. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics SMC 2016 - Conference Proceedings, pp. 2858–2865. IEEE, October 2017. https://doi.org/10.1109/SMC.2016.7844673

6. Hossain, M.R., Oo, A.M.T., Ali, A.B.M.S.: Hybrid prediction method for solar power using different computational intelligence algorithms. Smart Grid Renew. Energy **04**(01), 76–87 (2013). https://doi.org/10.4236/sgre.2013.41011
7. Hossain, M., Mekhilef, S., Danesh, M., Olatomiwa, L., Shamshirband, S.: Application of extreme learning machine for short term output power forecasting of three grid-connected PV systems. J. Clean. Prod. **167**, 395–405 (2018). https://doi.org/10.1016/j.jclepro.2017.08.081
8. Li, Y., He, Y., Su, Y., Shu, L.: Forecasting the daily power output of a grid-connected photovoltaic system based on multivariate adaptive regression splines. Appl. Energy **180**, 392–401 (2016). https://doi.org/10.1016/j.apenergy.2016.07.052
9. Long, H., Zhang, Z., Su, Y.: Analysis of daily solar power prediction with data-driven approaches. Appl. Energy **126**, 29–37 (2014). https://doi.org/10.1016/j.apenergy.2014.03.084
10. Nageem, R., Jayabarathi, R.: Predicting the power output of a grid-connected solar panel using multi-input support vector regression. Procedia Comput. Sci. **115**, 723–730 (2017). https://doi.org/10.1016/j.procs.2017.09.143
11. Rana, M., Koprinska, I., Agelidis, V.G.: Univariate and multivariate methods for very short-term solar photovoltaic power forecasting. Energy Convers. Manag. **121**, 380–390 (2016). https://doi.org/10.1016/J.ENCONMAN.2016.05.025
12. Shi, J., Lee, W.J., Liu, Y., Yang, Y., Wang, P.: Forecasting power output of photovoltaic systems based on weather classification and support vector machines. In: IEEE Transactions on Industry Applications, vol. 48, pp. 1064–1069, October 2012. https://doi.org/10.1109/TIA.2012.2190816
13. Sun, X., Zhang, T.: Solar power prediction in smart grid based on NWP data and an improved boosting method. In: 2017 IEEE International Conference on Energy Internet (ICEI), pp. 89–94. IEEE, April 2017. https://doi.org/10.1109/ICEI.2017.23
14. Theocharides, S., Makrides, G., Georghiou, G.E., Kyprianou, A.: Machine learning algorithms for photovoltaic system power output prediction. In: 2018 IEEE International Energy Conference (ENERGYCON), pp. 1–6. IEEE, June 2018. https://doi.org/10.1109/ENERGYCON.2018.8398737
15. Wolff, B., Kühnert, J., Lorenz, E., Kramer, O., Heinemann, D.: Comparing support vector regression for PV power forecasting to a physical modeling approach using measurement, numerical weather prediction, and cloud motion data. Solar Energy **135**, 197–208 (2016). https://doi.org/10.1016/j.solener.2016.05.051
16. Yang, H.T., Huang, C.M., Huang, Y.C., Pai, Y.S.: A weather-based hybrid method for 1-day ahead hourly forecasting of PV power output. IEEE Trans. Sustain. Energy **5**(3), 917–926 (2014). https://doi.org/10.1109/TSTE.2014.2313600
17. Yang, X.: Firefly algorithm, stochastic test functions and design optimisation. Int. J. Bio-Inspired Comput. **2**(2), 78–84 (2010)