Volume XXVI 2023
ISSUE no.1
MBNA Publishing House Constanta 2023

Scientific Bulletin of Naval Academy

SBNA PAPER • OPEN ACCESS

Photovoltaic systems forecast using machine learning algorithms and recurrent neural networks

To cite this article: A. Bâra and S-V Oprea, Scientific Bulletin of Naval Academy, Vol. XXVI 2023, pg. 178-185.

Submitted: 14.04.2023
Revised: 01.08.2023
Accepted: 18.08.2023

Available online at www.anmb.ro

ISSN: 2392-8956; ISSN-L: 1454-864X

doi: 10.21279/1454-864X-23-I1-022

Photovoltaic systems forecast using machine learning algorithms and recurrent neural networks

Adela BÂRA, Simona-Vasilica OPREA

Department of Economic Informatics and Cybernetics, Bucharest University of Economic Studies, Bucharest, Romania
Corresponding author: Simona-Vasilica OPREA, simona.oprea@csie.ase.ro

Abstract. The Photovoltaic (PV) systems are more present in the communities' landscape providing energy to the consumers, public buildings, municipalities, and industry, smoothen the electricity prices fluctuations and reducing the dependency on the public grid. They are reliable energy sources for boats and ships as some of the PV technologies are flexible and can be located on plane surfaces or even on the water surface especially when the ships dock at sea or at the seashore. However, the operation of PV systems depends on several weather factors, and it is important to predict their operation to manage the controllable load. Furthermore, it is essential to know if the PV systems generate in surplus or additional energy is required to cover the load. The surplus can be offered for local trading or aggregated and offered for centralized markets. Therefore, in this paper, we aim to predict the output of the PV systems using machine learning algorithms and recurrent neural networks (RNN), especially a multivariate Long Short-Term Memory (LSTM) model. Data extraction, feature engineering, and forecast of the PV power are depicted and the simulations are performed using 4 PV systems located in Constanta County. The results are assessed with prediction performance metrics such as RMSE, MAPE, etc.

Keywords: PV systems, forecast PV power, Long Short-Term Memory (LSTM), recurrent neural networks (RNN)

1. Introduction

The PV systems have started to become more frequent in the rural and urban communities as the technologies are getting cheaper [1]. Subsidies are given to the residential consumers to acquire up to 3kWh of PV systems per household and the top of the buildings (including blocks) will be the next target according to the Romanian decision makers. Currently, in Romania 2-3% of the load is covered by PV systems and 10-12% by the Wind Power Plants (WPP) located mainly in Dobrogea area. The total power for WPP sums up to 3,000 MW, dispatchable PV systems installed power is up to 612 MW and non-dispatchable PV systems installed power is around 780 MW. The PV systems will be installed on the blocks top roofs and their output will be shared among the apartments. Value sharing methods are essential to fairly share the cost and revenue the PV systems may bring as in case the load is lower than PV generation, the surplus is injected into the grid [2]. The surplus can be aggregated by an aggregator and offered to a supplier or grid operator as flexibility or on the market. The main disadvantage is related to the gap between the PV operating hours (with maximum output at noon) and peak load that usually takes place in the evening when most inhabitants return home [3], [4]. Furthermore, the PV systems do not generate at night. Therefore, storage facilities are essential to mind the gap between PV output and peak load. Load optimization is also a solution to maximize the PV usage [5]. Mainly the PV operation depends on the weather parameters: cloud cover, solar irradiation, intrinsic characteristics of the panels, location and site properties (azimuth, tilt, etc.) [6]. PV forecast is usually performed for home energy management system [7] or building management system [8], [9]. Furthermore, PV systems administrators have to notify the output and the forecast is therefore essential to improve notifications to the market and grid operators and avoid balancing costs. Numerous prediction models have been developed such as stochastic, artificial neural networks, hybrid methods [10], [11], including systems with battery-based storage facilities [12].

Thus, in this paper we proposed a methodology in three steps that consists of data extraction using Node-RED, feature engineering and the 15-min forecast itself using several standout Machine Learning (ML) algorithms (such as Random Forest, extreme Gradient Boosting, Light Gradient Boosting), Deep Neural Networks (DNN) and RNN (LSTM, Gated Recurrent Unit). Furthermore, we implemented the methodology into a PV dashboard application.

The paper is structured in several sections: introduction including related studies, methodology, results and conclusion. The simulations performed in this paper consider the input data of 4 PV systems with rated power between 0.5 and 2.97 MW, one located in Constanta County and three PV systems are situated in the North-West of Romania. In the next section, the proposed methodology steps will be depicted.

2. Methodology

The methodology consists of three steps: data extraction, feature engineering, and forecast the PV power.

2.1. Step 1 – Data extraction

The data sources consist of weather data collected from web API (Application Programming Interfaces), records of the PV power plants (inverters and smart meters) and records provided by the Distribution System Operator (DSO). Two different types of weather records are stored and related to the current conditions and forecasted values of the following parameters: temperature, cloud cover, humidity, wind speed and direction, ultraviolet index (UVI), pressure, dew point and precipitation. From the power plants, the readings of the inverters and the smart meters are collected at 15 minutes, preprocessed, and validated against missing values and outliers caused by errors. An extraction process automatically requests the data from these sources, transformed and stored them in a MySQL database. This process is implemented in Node-RED using *Pythonshell* nodes that run Python scripts to extract records from inverters, smart meters and weather API. The data is stored in a relational schema with *json* support due to the fact that the weather records are received as *json* documents.

The initial input is formed by merging the weather data and the power plant readings:

$$X = [X_w, X_{pp}] \tag{1}$$

Where X_w represents the weather records related to the previous and current conditions and X_{pp} represents the power generated by the inverters measured by the smart meter or by the grid operator.

2.2. Step 2 – Feature engineering

In this step, the initial input is prepared and enhanced with new attributes that model the PV generation. The factor with the greatest influence on generation is solar irradiation that can be determined from the cloud cover and pressure using the *irradiance.campbell_norman* method from the *PVLIB* Python library [13]. This method determined the three components of the solar irradiance (DNI - Direct Normal Irradiance, DHI - Diffuse horizontal irradiance, GHI - Global Horizontal Irradiance) from the extraterrestrial flux (which by default has a value of 1367 W/m^2), transmittance, and atmospheric pressure. Atmospheric transmittance is calculated based on the cloud cover using several approximations as described in [14]. The Effective Irradiance (EI) that is received by the PV panel depends on the tilt, azimuth, angle of incidence and albedo. It is calculated using the method *PVSystem.sapm_effective_irradiance* from the *PVLIB* library and it is added to the initial input.

Additionally, some aggregated values of the generated power are added to the input X to create a generation pattern under the following weather conditions: effective irradiance, UV index and Cloud Cover (CC). The new features represent the minimum, maximum, average and standard deviation of the generated power under the above-mentioned weather conditions:

$$X_{wag} = f_{ag}(X_w) | X_w \in \{EI, UVI, CC\} \quad (2)$$

Where f_{ag} represents the aggregation function (minimum, maximum, average or standard deviation) over the training period (*t*).

Also, to capture the power generated as a function of time, another four features are determined using the aggregation functions over each time interval (*t*) that corresponds to the quarter of an hour:

$$X_{tag} = f_{ag}(t) | t = \overline{1:96} \quad (3)$$

The last feature that is computed represent a combination of the effective irradiance and cloud cover that aims to strengthen the correlation between the generated power and these weather influencers:

$$X_{combi} = f_c(EI, CC) \quad (4)$$

Where f_c represents the concatenation function of the EI with CC.

The features are added to the initial input, and it becomes as following:

$$X = [X_w, X_{pp}, X_{wag}, X_{tag}, X_{combi}] \quad (5)$$

The output (target) of the ML algorithm is the generated power (P) in kW, so $y = P$.

To prepare the input and output for the ML training, a typical stage of preparation is applied that consists in standardization of the values and splitting of the data set into train (X_{train}, y_{train}) and test (X_{test}, y_{test}) sets, allocating 80% of the values for training and 20% for testing.

2.3. Step 3 – Forecast the PV power

For this step the following algorithms are used: Gradient Boosting (GB), Random Forest (RF), XGBoost (XGB), Light Gradient Boosting (LGB), Voting Regressor (VR), Deep Neural Network (DNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The hyperparameters of the GB, RF, XGB and LGB algorithms are similar, and their optimal values are obtained after a tuning process using *GridSearchCV* from *scikit-learn* library in Python. Therefore, the number of estimators is set to 200, maximum depth is between 5 and 10, learning rate is 0.05. The VR model is composed of three algorithms: GB, RF and XGB. The DNN architecture is obtained after the tuning process using validation curves while testing several configurations: three hidden layers with (128, 64, 64) neurons, activation function is set to the rectified linear unit (ReLU), optimizer is Adam with a learning rate of 0.001. After the first hidden layer a Dropout layer is added with dropout rate set to 0.2.

The first six methods do not need any further preparation of the input and are trained and tested on the corresponding data sets. For LSTM and GRU, the input and target data sets need to be transformed into 3D shapes with dimension formed by *n* features, *m* samples and *s* intervals corresponding to 96 daily intervals.

$$X3D \leftarrow X.reshape\ (m, 96, n);\ y3D \leftarrow y.reshape(m, 96) \quad (6)$$

For LSTM, two variants are configured as follows: LSTM3D with two LSTM layers with (128, 64) neurons followed by a Dropout layer with a rate of 0.2 and a Dense layer with 64 neurons; LSTMR with an initial LSTM layer with 128 neurons followed by a Repeat Vector layer and another LSTM layer with 64 neurons. A Dropout layer and a Dense layer are added with the same hyperparameters as LSTM3D. The GRU model is similar to LSTM3D, except for the fact that the LSTM layers are replaced by the GRU layers with the same hyperparameters. For all the recurrent models the optimizer is Adam with a learning rate of 0.001 and activation function is ReLU.

The models are trained on the last three months and forecast the PV output for the next 72 hours based on the weather forecast extracted from the API sources. The final PV forecast is obtained as an average of the predictions ($\widehat{PV_m}$) provided by all ML algorithms. Therefore, the PV output ($\widehat{PV}$) is:

$$\widehat{PV} = \frac{\sum_m \widehat{PV_m}}{M} \quad (7)$$

Where M is the total number of models (m).
The steps of the methodology are represented in Figure 1.

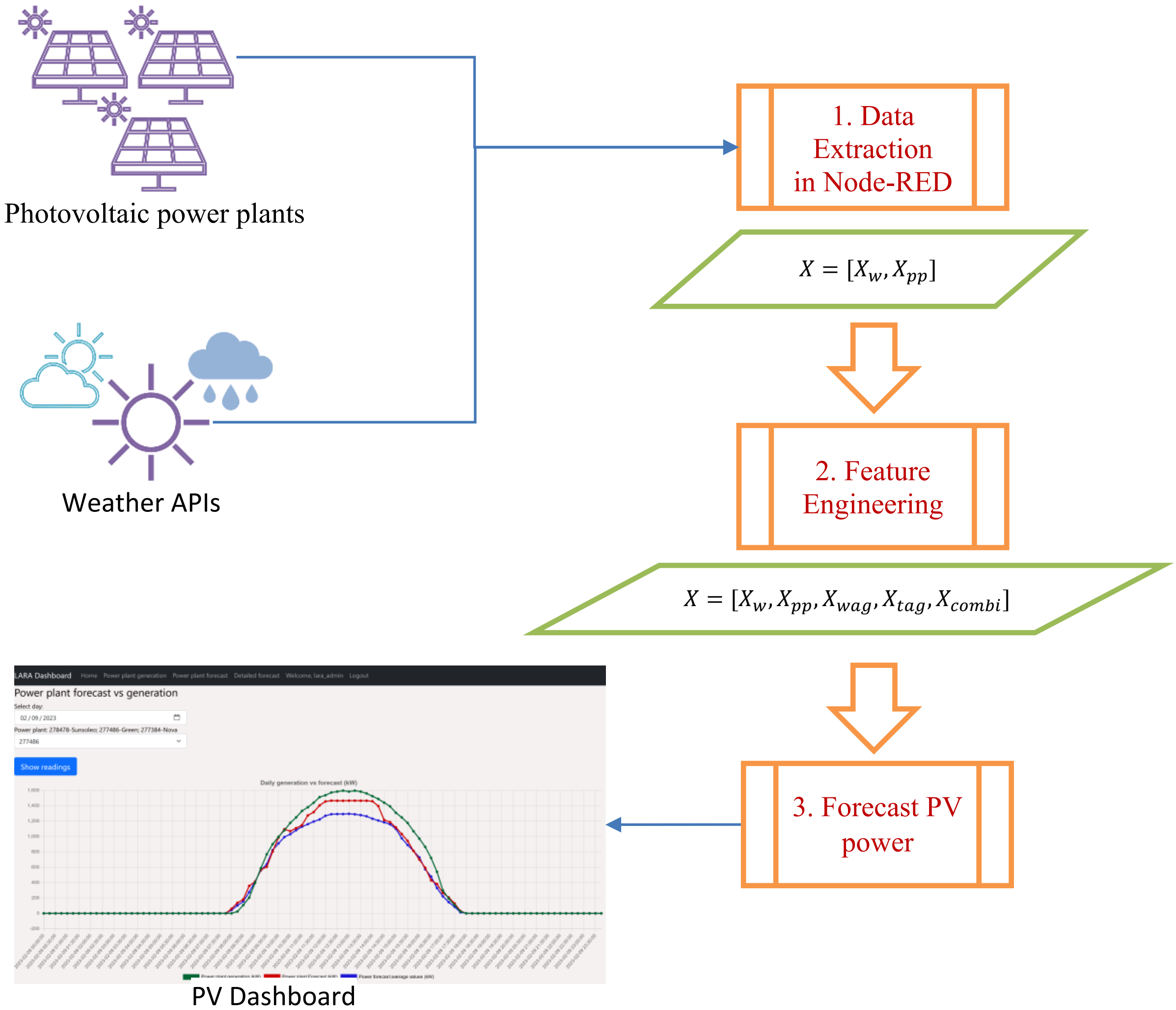

Figure 1. Steps of the proposed methodology for PV forecast.

3. Implementation and deployment of the PV Dashboard

The proposed methodology is implemented in Python, using Flask which is a Web Server Gateway Interface (WSGI) framework for web development. The web application, PV Dashboard consists in two sections for monitoring the PV generation and for forecasting the PV power. Each section provides tabular and graphical representations of the inverter and smart-meters readings and predicted power using ML algorithms. For example, in Figure 2 the forecasting section where several estimations are displayed compared with the generated power is shown. The following estimations are compared with the actual generation: *Power plant forecast* - obtained as an average of the predictions provided by all ML algorithms, including DNN, LSTM and GRU, *Average forecast* – obtained as an average of the ML models without DNN, LSTM and GRU, and three estimations obtained by calculating the PV power with *PVLIB* library only for single weather source: OpenWeather[1], Storm1 and Storm 2 respectively. Since Storm 1 and Storm 2 data are extracted from the API provided by Stormglass[2], their values may coincide for some locations depending on the location of the weather stations.

Figure 2. Forecasting section – comparison between generation and forecast.

As can be noticed from Figure 2, the estimations calculated directly from the weather data are far from the generated power, the closest values are obtained by the Power plant forecast and the Average forecast using ML models.

The tabular representation allows user to export the data in *excel* or *csv* files to be send to the DSO as daily notifications regarding the predicted power. In the following section, the results of the simulations using 4 PV systems with the installed power between 0.5 and 2.97 MW located one in Constanta County and the other three in the N-V part of Romania are depicted. Several metrics for the assessment of the prediction performance are calculated.

4. Results

The simulations were performed on four power plants in Romania with rated power between 0.5 and 2.97 MW. The first PV is located in Constanta County and has 0.5 MW installed power with 31 inverters. The data is collected directly from the inverters and validated with the DSO records. The other three PV systems are situated in the North-West of Romania and have between 2 and 2.97 MW

[1] https://openweathermap.org/api
[2] https://stormglass.io/

installed power. The data is collected from the smart meters that measure the entire PV generation and are validated with the DSO records. The ML algorithms are trained in the second step for each PV, and the results are centralized in Table 1, being averaged over the entire year 2022. The metrics used to evaluate the results are the following: Root-Mean Squared Error (*RMSE*), coefficient of determination (R^2) and Mean Absolute Percentage Error (*MAPE*).

Table 1. The results obtained in the training step, 01 January 2022 – 31 December 2022.

PV	RMSE	R^2	MAPE
PV1 – 0.5 MW	17.3	0.98	0.15
PV2 – 2.6 MW	58	0.97	0.59
PV3 – 2.97 MW	62	0.96	0.67
PV4 – 2.1 MW	87	0.92	0.88

After training and testing the models, the PV Dashboard was deployed into a testing environment and evaluated for two months, from the 1st of January 2023 to 28th of February 2023. The results are centralized in Table 2.

Table 2. The results obtained during the evaluation between 01 January 2023 – 28 February 2023.

PV	RMSE	R^2	MAPE
PV1 – 0.5 MW	21.1	0.95	0.19
PV2 – 2.6 MW	82.8	0.91	0.89
PV3 – 2.97 MW	86.04	0.90	0.92
PV4 – 2.1 MW	98.54	0.86	0.99

As it can be noticed, during the evaluation interval, the results proved that the algorithms are robust and that the overfitting is avoided. The algorithms are analysed, and their individual estimations are compared. In Figure 3 a comparison between the generated power (blue curve), averaged forecast ($\widehat{PV}$) and the estimations obtained by DNN (green), LSTM3D (purple curve - P_LSTM), LSTMR (light blue) and GRU (orange curve) for PPV2 are shown for two consecutive days (end of February and beginning of March 2022).

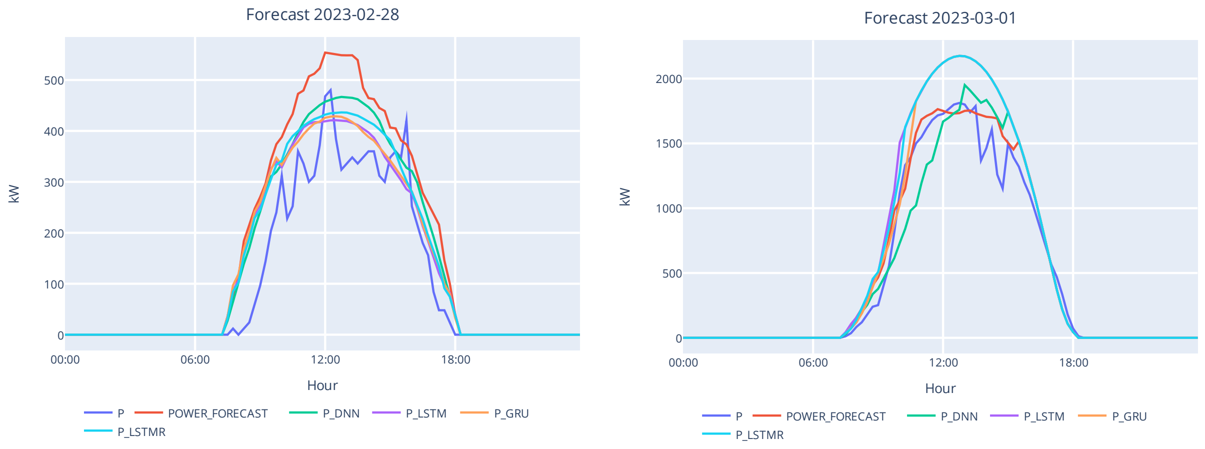

Figure 3. PV forecast using several ML models on two consecutive days for PPV2.

The differences between the estimations obtained by the ML models are small, demonstrating that the algorithms are reliable and that they can be used by the PPV's administrators to set up the notifications for the DSO, bid the forecasted quantities and manage the photovoltaic power plants.

5. Conclusion

There are numerous cases when PV forecast is important: for microgrids, for remoted areas or even for transportation, including sea transportation, for residential, commercial and industrial consumers, for PC systems administrators or owners to create notifications and bids, etc. The purpose of this paper is to propose a methodology to predict the PV output. The proposed methodology proved to be robust. Its application is already deployed in real operation. The simulations were performed using 4 PV systems with installed power between 0.5 and 2.97MW, one located in Constanta County and three located in the N-W part of Romania.

The implementation of the proposed methodology is performed as PV dashboard that offers tabular and graphical representation of the results. It monitors the operation of the PV systems and provides 15-min forecast for the next 72 hours.

Acknowledgement

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS-UEFISCDI, project number PN-III-P4-PCE-2021-0334, within PNCDI III.

References

[1] D. de São José, P. Faria, and Z. Vale, "Smart energy community: A systematic review with metanalysis," *Energy Strateg. Rev.*, 2021, doi: 10.1016/j.esr.2021.100678.

[2] A. Kulmala, M. Baranauskas, A. Safdarian, J. Valta, P. Jarventausta, and T. Bjorkqvist, "Comparing Value Sharing Methods for Different Types of Energy Communities," 2021. doi: 10.1109/ISGTEurope52324.2021.9640205.

[3] S. Preda, S. V. Oprea, A. Bâra, and A. Belciu, "PV forecasting using support vector machine learning in a big data analytics context," *Symmetry (Basel).*, 2018, doi: 10.3390/sym10120748.

[4] S. V. Oprea, A. Bâra, Ş. Preda, and O. B. Tor, "A smart adaptive switching module architecture using fuzzy logic for an efficient integration of renewable energy sources. A case study of a RES System located in Hulubeşti, Romania," *Sustain.*, 2020, doi: 10.3390/su12156084.

[5] S. Prajapati and E. Fernandez, "Residential load management with gray water recycling to maximize rooftop solar PV usage," *Energy Sources, Part A Recover. Util. Environ. Eff.*, 2019, doi: 10.1080/15567036.2019.1687618.

[6] M. Pierro *et al.*, "Progress in regional PV power forecasting: A sensitivity analysis on the Italian case study," *Renew. Energy*, 2022, doi: 10.1016/j.renene.2022.03.041.

[7] S. V. Oprea and A. Bâra, "Mind the gap between PV generation and residential load curves: Maximizing the roof-top PV usage for prosumers with an IoT-based adaptive optimization and control module," *Expert Syst. Appl.*, vol. 212, p. 118828, Feb. 2023, doi: 10.1016/J.ESWA.2022.118828.

[8] W. El-Baz, P. Tzscheutschler, and U. Wagner, "Day-ahead probabilistic PV generation forecast for buildings energy management systems," *Sol. Energy*, 2018, doi: 10.1016/j.solener.2018.06.100.

[9] S. P. Durrani, S. Balluff, L. Wurzer, and S. Krauter, "Photovoltaic yield prediction using an irradiance forecast model based on multiple neural networks," *J. Mod. Power Syst. Clean Energy*, 2018, doi: 10.1007/s40565-018-0393-5.

[10] H. J. Lu and G. W. Chang, "A Hybrid Approach for Day-Ahead Forecast of PV Power Generation," in *IFAC-PapersOnLine*, 2018, vol. 51, no. 28. doi: 10.1016/j.ifacol.2018.11.774.

[11] G. B. M. A. Litjens, E. Worrell, and W. G. J. H. M. van Sark, "Assessment of forecasting

methods on performance of photovoltaic-battery systems," *Appl. Energy*, 2018, doi: 10.1016/j.apenergy.2018.03.154.

[12] C. L. Nge, I. U. Ranaweera, O. M. Midtgård, and L. Norum, "A real-time energy management system for smart grid integrated photovoltaic generation with battery storage," *Renew. Energy*, 2019, doi: 10.1016/j.renene.2018.06.073.

[13] W. F. Holmgren, C. W. Hansen, and M. A. Mikofski, "pvlib python: a python package for modeling solar energy systems," *J. Open Source Softw.*, 2018, doi: 10.21105/joss.00884.

[14] H. C. Kim and E. E. Hofmann, "Evaluation and derivation of cloud-cover algorithms for calculation of surface irradiance in sub-Antarctic and Antarctic environments," *Antarct. Sci.*, 2005, doi: 10.1017/S0954102005002518.