

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Forecasting of Photovoltaic Solar Power Production Using LSTM Approach

Fouzi Harrou, Farid Kadri and Ying Sun

Abstract

Solar-based energy is becoming one of the most promising sources for producing power for residential, commercial, and industrial applications. Energy production based on solar photovoltaic (PV) systems has gained much attention from researchers and practitioners recently due to its desirable characteristics. However, the main difficulty in solar energy production is the volatility intermittent of photovoltaic system power generation, which is mainly due to weather conditions. For the large-scale solar farms, the power imbalance of the photovoltaic system may cause a significant loss in their economical profit. Accurate forecasting of the power output of PV systems in a short term is of great importance for daily/hourly efficient management of power grid production, delivery, and storage, as well as for decision-making on the energy market. The aim of this chapter is to provide reliable short-term forecasting of power generation of PV solar systems. Specifically, this chapter presents a long short-term memory (LSTM)-based deep learning approach for forecasting power generation of a PV system. This is motivated by the desirable features of LSTM to describe dependencies in time series data. The performance of the algorithm is evaluated using data from a 9 MWp grid-connected plant. Results show promising power forecasting results of LSTM.

Keywords: forecasting, deep learning, LSTM, solar power production

1. Introduction

Solar energy becomes one of the most promising sources for generating power for residential, commercial, and industrial applications [1, 2]. Solar photovoltaic (PV) systems use PV cells that convert solar irradiation into electric power. Renewable energy sources, in particular photovoltaic (PV) energy, has been progressively increased in recent years because of its advantages of being plentiful, inexhaustible, clean energy and environmentally friendly [3–5]. As one of the most popular renewable energy sources, solar energy has the advantages of abundant resources, no pollution, free use, and no transportation [6–8]. This greatly accelerated the installation of solar photovoltaic (PV) systems around the world.

Reliable and precise forecasting plays an important role in enhancing power plant generation based on renewable energy sources such as water, wind, and sun [9]. One of the most sustainable and competitive renewable energy sources is

solar photovoltaic (PV) energy which is becoming nowadays more attracting than ever before [3]. The main crucial and challenging issue in solar energy production is the volatility intermittent of PV system power generation due to mainly to weather conditions. In particular, a variation of the temperature and irradiance can have a profound impact on the quality of electric power production. A drop of more than 20% of power PV production can be observed in real PV energy plants. This fact usually limits the integration of PV systems into the power grid. Hence, accurately forecasting the power output of PV modules in a short-term is of great importance for daily/hourly efficient management of power grid production, delivery, and storage, as well as for decision-making on the energy market [10].

Precise forecasting of solar energy is important for photovoltaic (PV) based energy plants to facilitate early participation in energy auction markets and efficient resource planning [11]. Numerous methods have been reported in the literature for PV solar power forecasting. These methods can be classified into four classes: (i) statistical approaches based on data-driven formulation to forecast solar time series by using historical measured data, (ii) machine learning techniques, in particular, deep learning approaches based artificial neural network, (iii) physical models based on numerical weather prediction and satellite images, and (iv) hybrid approaches which are the combination of the above methods. In [12], a combined approach merging seasonal autoregressive integrated moving average (SARIMA), random vector functional link neural network hybrid model and discrete wavelet transform has been introduced for forecasting short-term solar PV power production. It has been shown that the combined models provide improved forecasting results compared to individuals ones. In [13], Gradient boosted regression trees approach has been used to predict solar power generation for 1–6 h ahead. It has been that this approach outperforms the simpler autoregressive models. In [14], a model combining seasonal decomposition and least-square support vector regression has been designed to forecast power output. This approach demonstrated good forecasting capacity compared to the autoregressive integrated moving average (ARIMA), SARIMA, and generalized regression neural network. In [15], a multi-variate ensemble forecast framework integrating ensemble framework with neural predictors and Bayesian adaptive combination is proposed for forecasting PV output power.

Most conventional solar power forecasting approaches are limited in uncovering the correlation of the limited data but are not able to deep correlation and uncover implicit and relevant information. With the huge data from the modern power system, the use of conventional approaches is not suited for guaranteeing precise forecasting. Recently, deep learning (DL) approaches have emerged as powerful machine learning tools that enable complicated pattern recognition and regression analysis and prediction applications [16–18]. DL approaches are becoming increasingly popular due to their good capacity in describing dependencies in time series data. Deep Learning is the result of the concatenation of more layers into the neural network framework. Over the past few decades, many deep learning models have been proposed including Boltzmann machines, Deep Belief Networks (DBN) and Recurrent Neural Networks (RNNs) [19]. RNN is a type of neural networks that exploits the sequential nature of input data. RNNs are used to model time-dependent data, and they give good results in the time series data, which have proven successful in several applications domains [3, 20, 21]. Long Short-Term Memory Networks (LSTM) is a type of RNNs that is able to deal with remembering information for much longer periods of time [22]. It is also considered as one of the most used RNN models for time series data predictions, which is perfectly suited

to PV solar power production forecasting problems. In this chapter, we applied the LSTM model to accurately forecast short-term photovoltaic solar power. The effectiveness of this approach is tested based on power output data collected from a 9 MWp grid-connected plant.

The next section introduces the core idea behind the LSTM model and how it can be designed and implemented. Then, Section 3 presents the results of solar photovoltaic power forecasting using the LSTM model. Lastly, conclusions are offered in Section 4.

2. Deep learning and forecasting of PV power production

Over the last decades, many studies have been dedicated to forecasting problems in several application domains. Recurrent Neural Networks (RNNs) have been successfully used in machine learning problems [23]. These models have been proposed to address time-dependent learning problems [22]. **Figure 1** shows the basic concept of RNNs; a chunk of a neural network, A, looks at some input x_t and outputs a value h_t . It should be noted that RNNs are suited to learn and extract temporal information [24]. A general formula for RNN hidden state h_t given an input sequence $\mathbf{x} = (x_1, x_2, \dots, x_t)$:

$$h_t = \begin{cases} 0, & t = 0 \\ \varphi(W_{x_t}, x_t), & otherwise \end{cases} \quad (1)$$

where φ is a non-linear function. The update of recurrent hidden state is realized as:

$$h_t = g(W_{x_t} + uh_{t-1}) \quad (2)$$

where g is a hyperbolic tangent function (*tanh*).

Generally, it is not easy to capture long term time dependencies in time series when using recurrent neural networks. To bypass this limitation, Long Short-Term Memory Networks (LSTM) models were designed. LSTM is an extended version of RNN that are effectively capable to handle time dependency in data [22]. These models are flexible and efficient to describe time-dependent data, and they demonstrated success in several applications. LSTM is one of the most used RNN models for time series data predictions, which is perfectly suited to the PV forecasting problems [22]. Next, we present a basic overview of LSTM and how it can be designed and implemented.

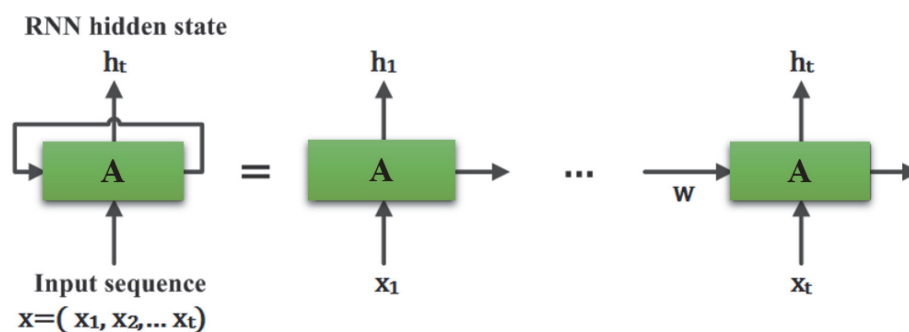


Figure 1.
 Basic illustration of RNN.

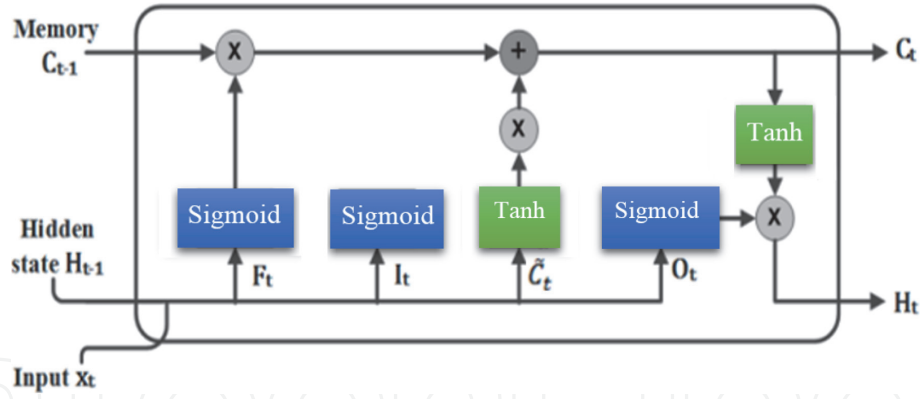


Figure 2.
Illustration of LSTM unit.

2.1 Long short-term memory (LSTM) models

The Long Short-Term Memory (LSTM) is a variant of the Recurrent Neural Networks (RNN) that is capable of learning long term dependencies. LSTM models were initially proposed by Hochreiter and Schmidhuber [4] and were improved and popularized by many other researchers [4–6, 9]. LSTM models have an excellent ability to memorize long-term dependencies, are developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM models over ANNs models, hidden Markov models and other sequence learning methods in several application domains.

A common LSTM model is composed of cell blocks in place of standard neural network layers. These cells have various components called the input gate, the forget gate and the output gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell [5]. **Figure 1** shows the basic structure of RNN-LSTM.

From **Figure 2**, the RNN-LSTM has two input features at each time, which include the current time step input X_t (input vector) and the hidden state of the previous time step H_{t-1} (previous input vector). The output is computed by the fully connected layer with its activation function (e.g., \tanh , sigmoid, Softmax, and Adam). Therefore, the output of each gate can be obtained through logical operation and nonlinear transformation of input.

Let us denote the input time series as X_t , the number of hidden units as h , the hidden state of the last time step as H_{t-1} , and the output time series as H_t . The mathematical relationship between inputs and outputs of the RNN-LSTM can be described as follows.

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \quad (3)$$

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f) \quad (4)$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \quad (5)$$

$$\tilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c) \quad (6)$$

$$C_t = F_t \circ C_{t-1} \circ \tilde{C}_t \quad (7)$$

$$H_t = O_t \circ \tanh(C_t) \quad (8)$$

where

- I_t, F_t, O_t are input gate, forget gate, and output gate respectively, W_{xi}, W_{xf}, W_{xo} and W_{hi}, W_{hf}, W_{ho} are weight parameters and b_i, b_f, b_o are bias

parameters. All these gates have the same dimensions and the same equations just with different parameters. They are called gates because the activation function transforms the element values between ranges $([0, 1], [-1, 1])$. The input gate defines how much of the newly computed state for the current input you want to let through. The forget gate defines how much of the previous state you want to let through. The output gate defines how much of the internal state you want to expose to the external network (higher layers and the next time step).

- \tilde{C}_t is the candidate memory cells, W_{xc} , W_{hc} are weight parameters and b_c is a bias parameter. LSTM model needs to compute the candidate memory cell \tilde{C}_t , its computation is similar to the three gates (input, forget and output gates), but using a \tanh function as an activation function with a value range between $[-1, 1]$.
- C_t is the memory cells, \odot is an operator that expresses element-wise multiplication. The computation of the current time steps memory cell C_t combines the information of the previous time step memory cells (C_{t-1}) and the current time step candidate memory cells (\tilde{C}_t), and controls the flow of information through forgetting gate and input.
- H_t is the hidden states, we can control the flow of information from memory cells to the hidden state H_t through the output gate. The \tanh function ensures that the hidden state element value is between $[-1, 1]$. It should be noted that when the output gate is approximately 1, the memory cell information will be passed to the hidden state for use by the output layer; and when the output gate is approximately 0, the memory cell information is only retained by itself.

2.2 Proposed approach

The proposed approach in this chapter aims to forecast solar power production. This methodology is based on the LSTM deep-learning model. **Figure 3** summarizes the main steps of the proposed methodology. The proposed approach includes four key steps (**Figure 3**):

- i. Collect the SCADA data from the PV system.
- ii. Pre-process and clean data by removing outliers and imputing missing values.
- iii. Normalize the original data.
- iv. Train, validate and test the LSTM model. Various statistical indicators are used to quantify the accuracy of the developed model. Lastly, the designed LSTM model can be used for power production forecasting.

2.3 Metrics for evaluating the forecasting models

To assess the forecasting performance, numerous statistical indicators have been proposed in the literature including root mean square error (RMSE), mean absolute error (MAE), coefficient of determination (R^2), and mean absolute percentage

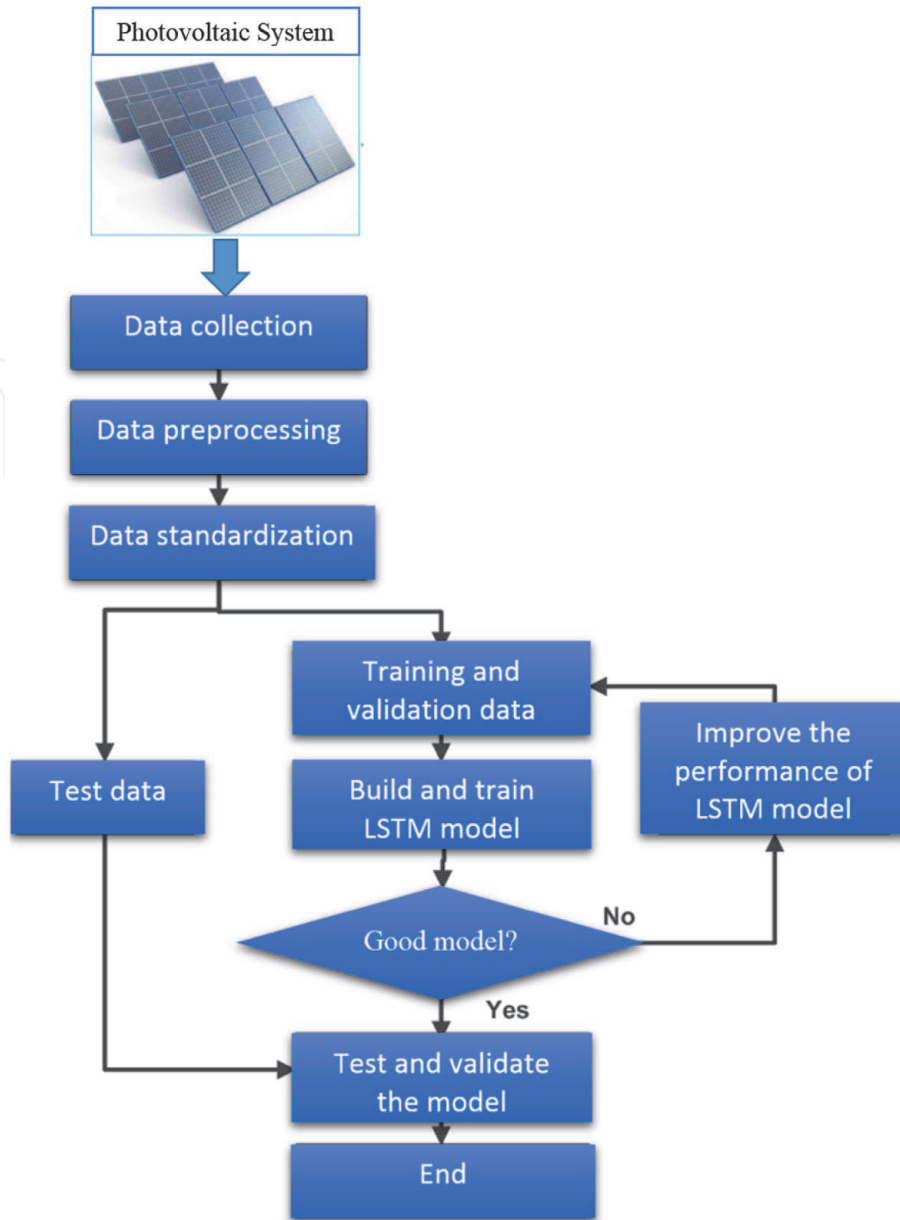


Figure 3.
Schematic block of the proposed forecasting method.

error (MAPE). In this study, we used R^2 and MAPE, which are frequently to evaluate the forecasting accuracy:

$$MAE = \frac{1}{n} \sum |\hat{x} - x| \quad (9)$$

$$RMSE = \sqrt{\frac{\sum (\hat{x} - x)^2}{n}} \quad (10)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{\sum_{i=1}^N (x_i - \bar{x})^2} \quad (11)$$

where x are the measured values, \hat{x} are the corresponding forecasted values by the LSTM model and n is the number of measurements.

2.4 Implementation steps

Essentially, the LSTM model can be designed and implemented in four main steps. At first, define the LSTM model and train it, then fit the LSTM model, and

lastly, the trained LSTM model is used for forecasting. **Table 1** summarizes the main steps (partial codes) performed in designing the LSTM model.

2.5 Enhance LSTM models performance

The key factors impacting the accuracy of the LSTM model are not only the amount of training data but also the architecture of the network, hyper-parameters and the utilized optimizers. Accordingly, the performance of LSTMs can be enhanced by acting on the following elements.

- *Activation functions*: activation functions an important role in determining the final response of the neural network. Two families of functions are distinguished: linear and nonlinear functions. The output of the linear activation functions is linearly proportional to the inputs and is not limited between any ranges. They are more suited than a step function because they permit obtaining multiple outputs, not just binary output (i.e., yes and no). On the other hand, nonlinear activation functions are the most frequently utilized because they are flexible and permit obtaining nonlinear output and they are confined within a range. For instance, Sigmoid, Softmax, and Rectified Linear Unit (ReLU) activation functions permit rescaling the data to values in the interval $[0, 1]$, while Hyperbolic Tangent (tanh) activation functions rescale the data within $[-1, 1]$.
- *Optimizer*: in the training phase of the LSTM model, optimization algorithms are used for minimizing its error rate. The performance of an optimizer is generally characterized by convergence speed and generalization (the efficiency of the model on new datasets). The commonly used optimizers include Adaptive Moment Estimation (Adam) or Stochastic Gradient Descent (SGD) [25, 26].
- *Dropout*: it is a well-known stochastic regularization procedure applied to avoid overfitting and further enhance the prediction capacity of RNN models [27]. More details about dropout techniques can be found in [26–28].
- *Epochs and batches*: the number of epochs and batch are two important parameters when constructing deep learning models. It has been shown in the

Steps	Action
Step 1: Define LSTM network	<pre>from keras.layers.recurrent import LSTM from keras.models import Sequential from keras.layers.core import Activation, Dense, Dropout model = Sequential() model.add(LSTM(units=nb_neural, return_sequences=True, input_shape=(Xtrain.shape[1], 1)))</pre>
Step 2: Compile the LSTM network	<pre>model.compile(loss="mse", optimizer="adam", metrics=[rmse, 'mae', Rsquare])</pre>
Step 3: Fit the LSTM network	<pre>history = model.fit(Xtrain, ytrain, batch_size=batch_size, epochs=num_epochs, validation_data=(Xval, yval), verbose=2)</pre>
Step 4: Forecasting	<pre>ypred = model.predict(Xtest)</pre>

Table 1.
 Partial codes used for building the LSTM network.

literature that good results can be achieved when using large epochs and small batch sizes.

- *Weight regularization*: another way to avoid overfitting and improves model performance is called weight regularization. This approach imposes constraints on the RNN weights within nodes to allow the network to maintain the weights small. Several penalizing or regularization approaches are commonly used in the literature based on L1 or L2 vector norm penalty.

3. Results and discussion

This study is based on real data collected from January 2018 to December 2018 every 15 min from a 9 MWp grid-connected plant. **Figure 4** shows the hourly distribution of PV power production day from January 2018 to December 2018.

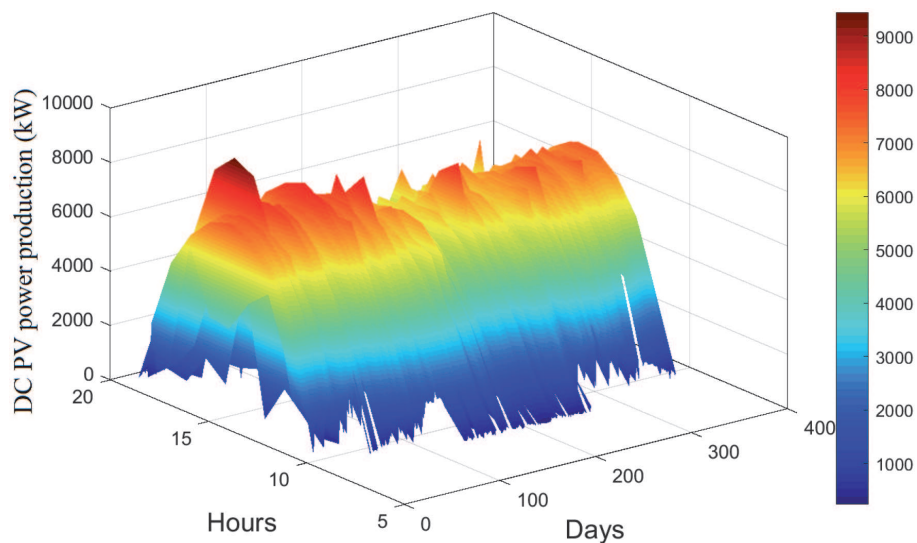


Figure 4.
PV power production per hour for each day from January 2018 to December 2018.

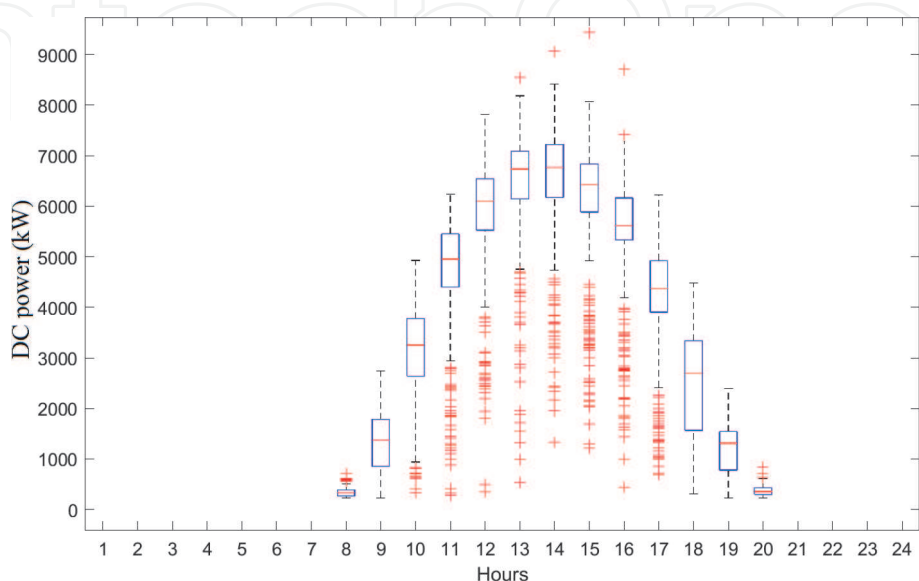


Figure 5.
Distribution of DC power output in the daytime.

From **Figure 4**, the solar PV power production reaches, every day, its maximum at the mid-day and falls to zero over the night.

For the data in **Figure 4**, the box plots showing the distribution of DC power generation in the daytime are displayed in **Figure 5**. One can see that the maximum power production is achieved around mid-day.

The monthly cumulative DC power generated by the inspected PV system from January 2018 to December 2018 is displayed in **Figure 6**. The highest and lowest monthly cumulative power are respectively achieved in March (6450.056 MW) and in October (4655.524 MW).

Figure 7 shows the monthly distribution of DC power production during the monitored period. **Figure 7** shows that the produced DC power is relatively high in January, February, and March. Also, it can be noticed that the production was relatively low from June to September (**Figure 7**).

To investigate the interactions between the DC power and meteorological factors (i.e., inclined irradiance 27, ambient temperature, and wind velocity) a Pearson correlation heatmap is displayed in **Figure 8**. From **Figure 8**, one can see that there

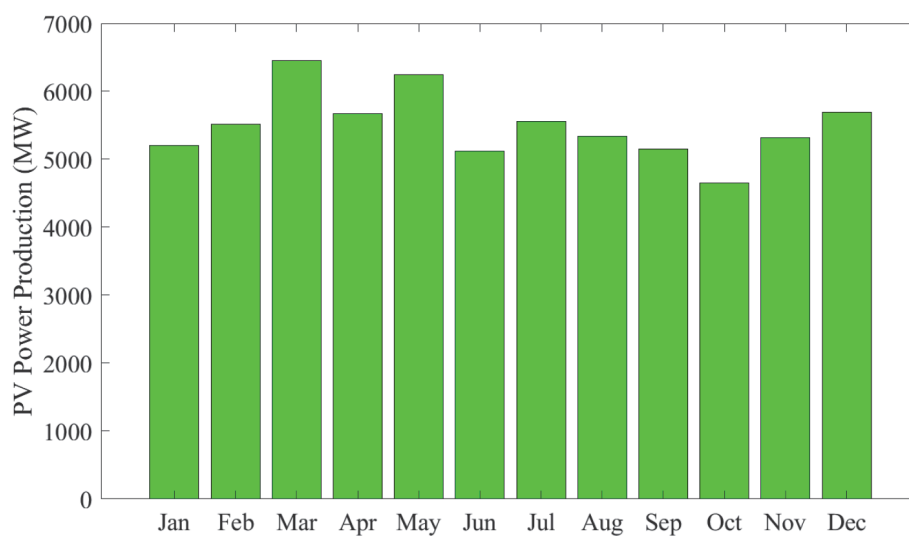


Figure 6.
Monthly total DC power produced from January 2018 to December 2018.

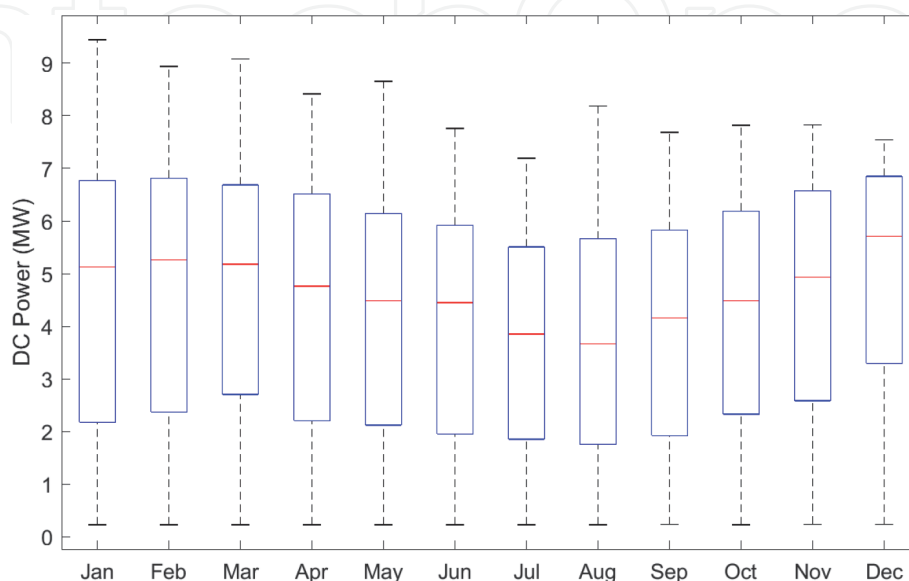


Figure 7.
Monthly distribution of DC power output.

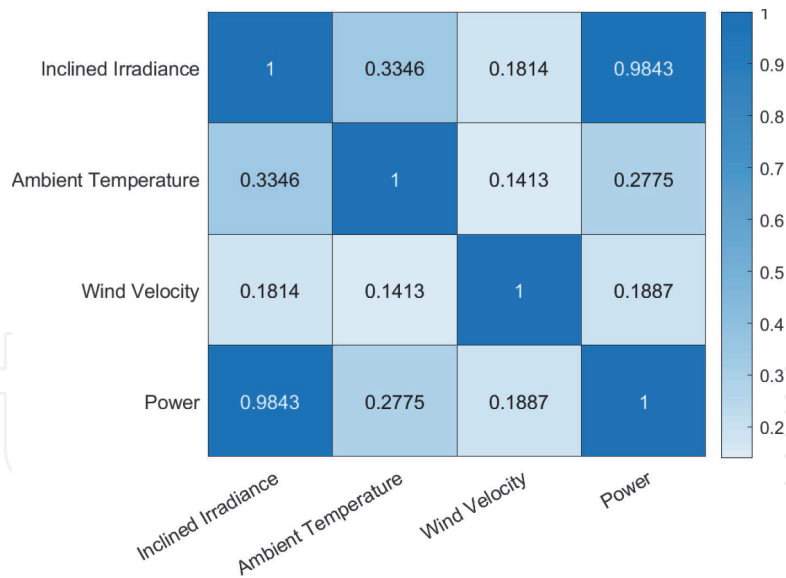


Figure 8. Heatmap of the correlation matrix of data: inclined irradiance 27, ambient temperature, wind velocity, and power.

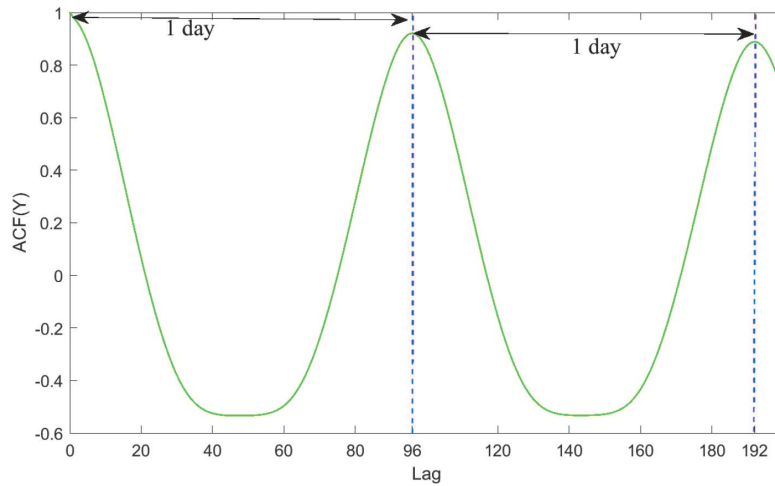


Figure 9. ACF DC power measurements.

Models	Parameter	Value
	Number of LSTM layers	1
	learning rate	0.001
	Loss function	Cross-entropy
	Number of hidden units	200
	Optimizer	Adam
	Number of training epochs	100

Table 2. Parameters in LSTM model.

is a high correlation between solar irradiance and power production. It should be noted that DC Power has a low correlation with wind velocity.

Figure 9 shows the autocorrelation function (ACF) plot of the data shown in power generation data. A seasonality of 24 h can be seen from the ACF plot of PV power data, the time difference between two maximum in the ACF (**Figure 9**). In particular, this seasonality is mainly due to the variation of solar irradiance.

The LSTM model has been constructed and then used for forecasting. Data were split into training and testing datasets (90% and 10% respectively). Parameters of the constructed LSTM are presented in **Table 2**.

The evolution of the loss function and RMSE in the function of the number of iterations is displayed respectively in **Figures 10** and **11**. **Figures 10** and **11** indicate the convergence of the loss function and RMSE when the number of epochs is around 60.

Once the LSTM model has been constructed based on training data, it will be employed to forecast future values of power production. We attempt now to test

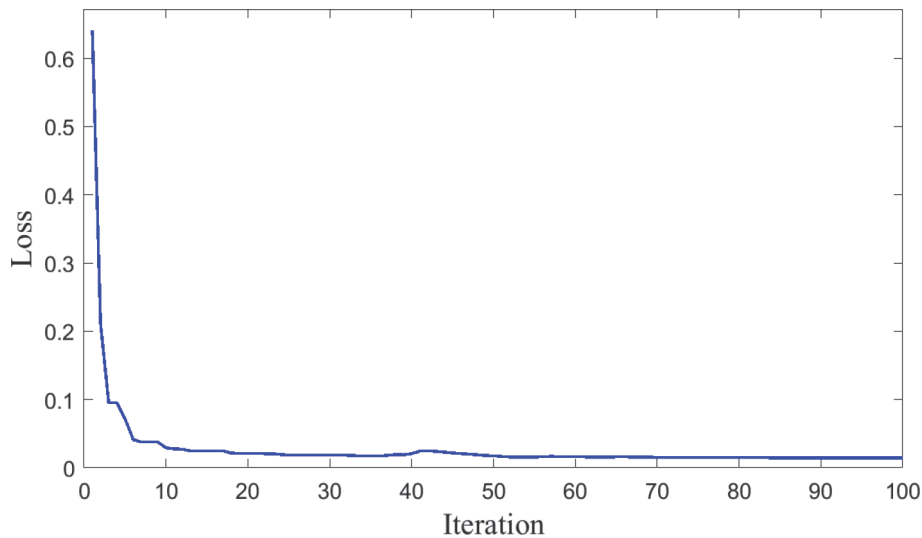


Figure 10.
Evolution of LSTM loss function during training stage.

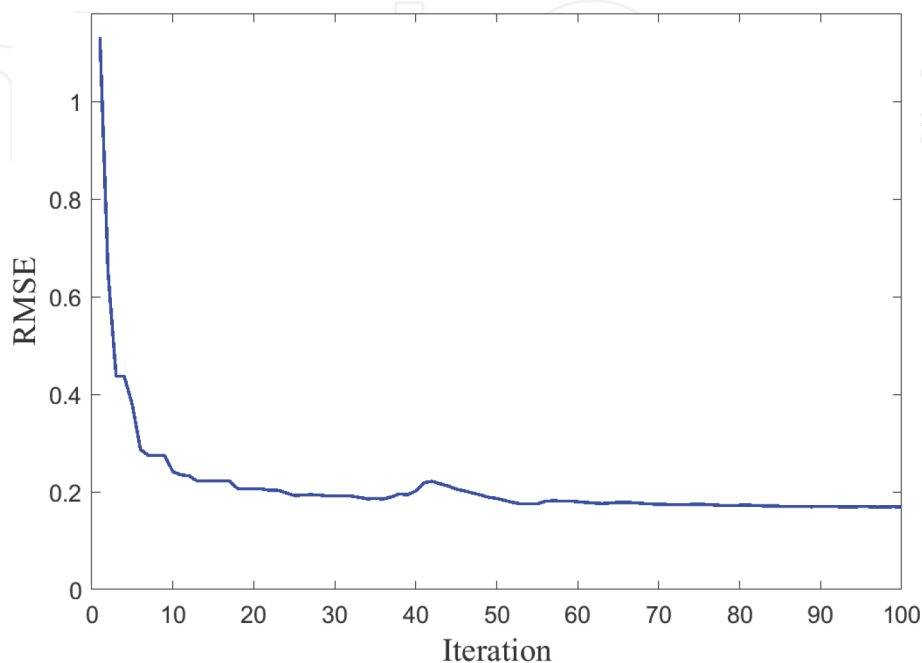


Figure 11.
Evolution RMSE of LSTM model during training stage.

the capability of the above LSTM model to forecast future values of the PV power generation. **Figure 12** shows the forecasting results of the PV power generation compared with the real data over a time horizon. **Figure 13** shows the scatter plot of the measured and forecasted power production via the LSTM model. It can be seen from **Figures 12** and **13** that the computed LSTM model has the ability of short-term forecasting of PV power generation. In addition, the forecasting result in **Figure 12** illustrates the efficiency of the LSTM model to forecast PV power production even under a cloudy day (i.e., the second day in **Figure 12**) where the power data is very dynamic.

In summary, the LSTM model showed good forecasting capacity with the coefficient of determination $R^2 = 0.98$ close to 1 and relatively small mean absolute

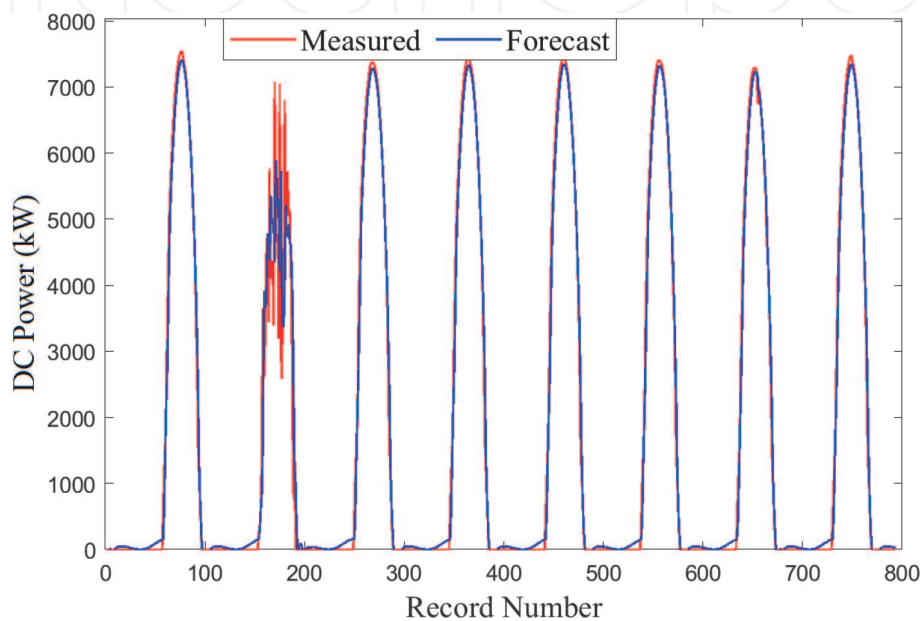


Figure 12.
Plot of collected solar power and forecasted one using LSTM model.

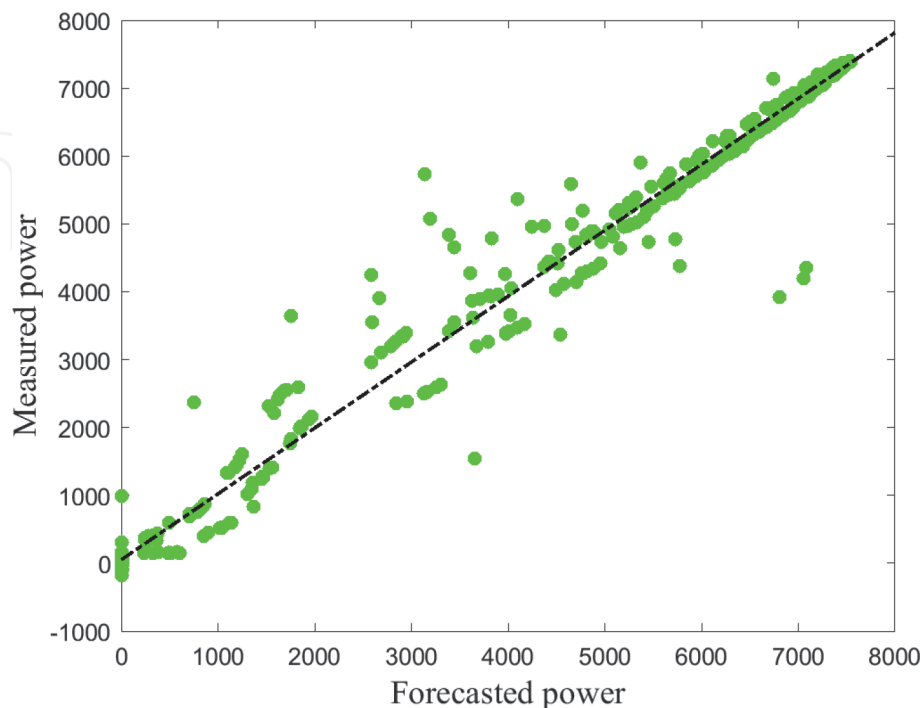


Figure 13.
Scatter graph of measured and LSTM forecast solar power output.

percentage error (MAPE), MAPE = 8.93. It should be pointed out that the forecasting accuracy in cloudy days could be improved by including meteorological variables, such as solar irradiance, ambient temperature, and wind velocity, as input variables.

4. Conclusion

The major challenge in solar energy generation is the volatility intermittent of photovoltaic system power generation due mainly to weather conditions. Thus, accurate forecasting of photovoltaic power generation is becoming indispensable for reducing the effect of uncertainty and energy costs and enable suitable integration of photovoltaic systems in a smart grid. This chapter employed a Long Short-Term Memory (LSTM) model to accurately forecast short-term photovoltaic solar power. This approach exploits the desirable properties of LSTM, which is a powerful tool for modeling dependency in data. The forecasting quality of this approach has been verified using data from January 2018 to December 2018 collected from a 9 MWp grid-connected plant. Promising results have been achieved by the proposed LSTM-based approach to short-term forecasting of photovoltaic solar power production. As future work, to further enhance the forecasting quality we plan to implement and test the performance of other RNN models like Gated recurrent unit (GRU) model and to incorporate other information such as meteorological data. Also, as most data from real plants are multiscale in nature and noisy, we plan in future work to merge the desirable LSTM model with the wavelet-based multiscale presentation [29]. This permits to get a multiscale LSTM model able to capture feature in both time and frequency and possess good ability to handle noisy data.

Acknowledgements

This publication is based upon work supported by King Abdullah University of Science and Technology (KAUST), Office of Sponsored Research (OSR) under Award No: OSR-2019-CRG7-3800.

IntechOpen

Author details


Fouzi Harrou^{1*}, Farid Kadri^{2*} and Ying Sun¹

1 Electrical and Mathematical Sciences and Engineering (CEMSE) Division, King Abdullah University of Science and Technology (KAUST) Computer, Saudi Arabia

2 Aeroline Customer Services, Sopra Steria Group, France

*Address all correspondence to: fouzi.harrou@kaust.edu.sa and farid.kadri@soprasteria.com

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. Distributed under the terms of the Creative Commons Attribution - NonCommercial 4.0 License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited. 

References

- [1] Sobri S, Koohi-Kamali S, Rahim NA. Solar photovoltaic generation forecasting methods: A review. *Energy Conversion and Management*. 2018;**156**: 459-497
- [2] Harrou F, Taghezouit B, Sun Y. Improved kNN-based monitoring schemes for detecting faults in PV systems. *IEEE Journal of Photovoltaics*. 2019;**9**(3):811-821
- [3] Antonanzas J, Osorio N, Escobar R, Urraca R, Martinez-de Pison F, Antonanzas-Torres F. Review of photovoltaic power forecasting. *Solar Energy*. 2016;**136**:78-111
- [4] Eseye AT, Zhang J, Zheng D. Short-term photovoltaic solar power forecasting using a hybrid wavelet-PSO-SVM model based on SCADA and meteorological information. *Renewable Energy*. 2018;**118**:357-367
- [5] Wang H, Yi H, Peng J, Wang G, Liu Y, Jiang H, et al. Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network. *Energy Conversion and Management*. 2017;**153**:409-422
- [6] Wang K, Qi X, Liu H. A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network. *Applied Energy*. 2019;**251**:113315
- [7] Harrou F, Dairi A, Taghezouit B, Sun Y. An unsupervised monitoring procedure for detecting anomalies in photovoltaic systems using a one-class support vector machine. *Solar Energy*. 2019;**179**:48-58
- [8] Harrou F, Taghezouit B, Sun Y. Robust and flexible strategy for fault detection in grid-connected photovoltaic systems. *Energy Conversion and Management*. 2019;**180**: 1153-1166
- [9] Behera MK, Majumder I, Nayak N. Solar photovoltaic power forecasting using optimized modified extreme learning machine technique. *Engineering Science and Technology, an International Journal*. 2018;**21**(3): 428-438
- [10] Qing X, Niu Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy*. 2018;**148**:461-468
- [11] Srivastava S, Lessmann S. A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data. *Solar Energy*. 2018;**162**: 232-247
- [12] Kushwaha V, Pindoriya NM. A SARIMA-RVFL hybrid model assisted by wavelet decomposition for very short-term solar PV power generation forecast. *Renewable Energy*. 2019;**140**: 124-139
- [13] Persson C, Bacher P, Shiga T, Madsen H. Multi-site solar power forecasting using gradient boosted regression trees. *Solar Energy*. 2017;**150**: 423-436
- [14] Lin K-P, Pai P-F. Solar power output forecasting using evolutionary seasonal decomposition least-square support vector regression. *Journal of Cleaner Production*. 2016; **134**:456-462
- [15] Raza MQ, Mithulananthan N, Summerfield A. Solar output power forecast using an ensemble framework with neural predictors and Bayesian adaptive combination. *Solar Energy*. 2018;**166**:226-241
- [16] Chen Z, Chen Y, Wu L, Cheng S, Lin P, You L. Accurate modeling of photovoltaic modules using a 1-d deep residual network based on iv

characteristics. *Energy Conversion and Management*. 2019;**186**:168-187

[17] Harrou F, Dairi A, Sun Y, Kadri F. Detecting abnormal ozone measurements with a deep learning-based strategy. *IEEE Sensors Journal*. 2018;**18**(17):7222-7232

[18] Harrou F, Dairi A, Sun Y, Senouci M. Statistical monitoring of a wastewater treatment plant: A case study. *Journal of Environmental Management*. 2018;**223**:807-814

[19] Dairi A, Harrou F, Sun Y, Senouci M. Obstacle detection for intelligent transportation systems using deep stacked autoencoder and k-nearest neighbor scheme. *IEEE Sensors Journal*. 2018;**18**(12):5122-5132

[20] Ugurlu U, Oksuz I, Tas O. Electricity price forecasting using recurrent neural networks. *Energies*. 2018;**11**(5):1255

[21] Fu R, Zhang Z, Li L. Using LSTM and GRU neural network methods for traffic flow prediction. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC). Wuhan, China: IEEE; 2016. pp. 324-328

[22] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*. 1997;**9**(8):1735-1780

[23] Dorffner G. Neural networks for time series processing. In: *Neural Network World*. Citeseer^x; 1996

[24] Oksuz I, Cruz G, Clough J, Bustin A, Fuin N, Botnar RM, et al. Magnetic resonance fingerprinting using recurrent neural networks. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). Venice, Italy: IEEE; 2019. pp. 1537-1540

[25] Mukkamala MC, Hein M. Variants of RMSProp and Adagrad with

logarithmic regret bounds. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. Sydney, Australia: JMLR.org; 2017. pp. 2545-2553

[26] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*. 2014;**15**(1): 1929-1958

[27] Watt N, du Plessis MC. Dropout for recurrent neural networks. In: Oneto L, Navarin N, Sperduti A, Anguita D, editors. *Recent Advances in Big Data and Deep Learning*. Genova, Italy: Springer International Publishing; 2020. pp. 38-47

[28] Gal Y, Ghahramani Z. A theoretically grounded application of dropout in recurrent neural networks. In: *Advances in Neural Information Processing Systems*. Barcelona, Spain; 2016. pp. 1019-1027

[29] Harrou F, Sun Y, Madakyaru M. An improved wavelet-based multivariable fault detection scheme. In: *Uncertainty Quantification and Model Calibration*. IntechOpen; 2017. p. 175