- ORIGINAL ARTICLE -

Short Term Cloud Nowcasting for a Solar Power Plant based on Irradiance Historical Data

Predicción de Nubes a Corto Plazo para una Planta Solar a partir de Datos Históricos

Rafael Caballero[1], Luis F. Zarzalejo[2], Álvaro Otero[1], Luis Piñuel[1] and Stefan Wilbert[3]

[1] *University Complutense de Madrid, 28040 Madrid, Spain*
{rafacr, alvama06, lpinuel}@ucm.es
[2] *Renewable Energy Division. Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), 28040 Madrid, Spain.*
lf.zarzalejo@ciemat.es
[3] *Institute of Solar Research, German Aerospace Center (DLR), 04200 Tabernas, Spain*.
stefan.wilbert@dlr.de

Abstract

This work considers the problem of forecasting the normal solar irradiance with high spatial and temporal resolution (5 minutes). The forecasting is based on a dataset registered during one year from the high resolution radiometric network at a operational solar power plan at Almeria, Spain. In particular, we show a technique for forecasting the irradiance in the next few minutes from the irradiance values obtained on the previous hour. Our proposal employs a type of recurrent neural network known as LSTM, which can learn complex patterns and that has proven its usability for forecasting temporal series. The results show a reasonable improvement with respect to other prediction methods typically employed in the studies of temporal series.

Keywords: cloud nowcasting, GHI, LSTM, supervised machine learning.

Resumen

Es este trabajo se aborda el problema de la predicción de radiación global sobre superficie horizontal con alta resolución espacial y temporal (5 minutos) a partir de los datos registrados durante un año en la red radiométrica de alta resolución ubicada en la Plataforma Solar de Almería. En particular se muestra un método capaz de predecir el valor de radiación en los siguientes minutos a partir de los valores de los minutos anteriores. El método emplea el tipo de red neuronal recurrente conocido como LSTM, capaz de aprender patrones complejos y predecir el próximo elemento de una serie temporal. Los resultados muestran una mejora apreciable en con respecto a los métodos de predicción empleados habitualmente en el estudio de series temporales.

Palabras claves: aprendizaje automático supervisado, GHI, LSTM, previsión de nubes.

1. Introduction

In the last decades the generation of energy from renewable resources has become a pressing need due to the environmental problems associated with waste and emissions that conventional production systems cause. Moreover, the production of this energy is reaching a high level of competitivity with respect to the traditional energy generation sources. In particular, the solar energy has acquired a more relevant role among the renewable resources, with a noticeable increment on the number of operational solar power plants. This increment, which corresponds to an increase of the power supplied to the energy distribution grids, implies the necessity of forecasting how much energy will be supplied by the solar power plant. Additionally, the own plants also require this forecasting to participate in the energy markets and in order to plan the maintenance operations. Thus, the capacity of forecasting the available solar resources is critical for operating solar-based power stations.

This work deals with the problem of nowcasting, that is, short term forecasting of the global horizontal irradiance (GHI). In particular, we use as a example a dataset obtained from the high resolution radiometric network located at the *Plataforma Solar de Almería* (www.psa.es), although we think that similar results can be obtained in similar environments.

Citation: R. Caballero, L.F. Zarzalejo, A. Otero, Luis Piñuel and S. Wilbert. *Short term cloud nowcasting for a solar power plant based on irradiance historical data.* Journal of Computer Science & Technology, vol. 18, no. 3, pp. 186–192, 2018.

DOI: 10.24215/16666038.18.e21

Received: August 1, 2018 **Accepted:** October 31, 2018

The goal is to obtain a technique that:

a) Reduces the average prediction error with respect to traditional forecasting methods, such as using the last known value as predictor, or statistical techniques such as ARIMA.
b) Our medium-term goal is to integrate the forecasting model in a real photovoltaic solar power plant. Thus, the proposed method must define the models and predict GHI values on real time. In particular, considering for instance that the sensors emit new values every minute, it would be interesting to obtain the predictions in less than one minute, even if the forecasting horizon is of several minutes. Otherwise, the technique will not be using the last information received, with the corresponding loss of accuracy.

Summarizing, we look for a lightweight GHI nowcasting method that can improve the most basic statistical predictions which are, as we show in the experiments, very good predictors for very short time horizons.

The radiation forecasting models can be grouped in three different types. The first type are the *physical models*, based on mathematical equations that describe the atmosphere physics [1], for instance using vectors representing the movement of the clouds [2-3]. A second type is constituted by the *statistical models* [4-5], which establish statistical connections between past and future observations. Our proposal can be included in the third line, which also employ historical data but based on *machine learning models*, such as neuronal networks [6-8].

The main objective of this work is the very short-term prediction, between 1 and 10 minutes (nowcast). This is a significative difference compared to previous references mentioned, that focus on predictions that focus of horizons over 30 minutes [8]. It must be emphasized that any improvement in the forecast, even if it is small, has an economic impact: a more accurate forecast of the solar irradiation will result in a better estimation of the supply of electric power to the electricity network.

In the next section we present the dataset to which we have applied the proposed methodology. In section 3 we will look for a possible segmentation for differentiating between cloudy and cloudless periods. Section 4 shows the type of neural network proposed and the methods that we have chosen to compare with our proposal. The results obtained are explained in section 5. Finally, section 6 presents the final conclusions.

2. Initial Dataset

2.1. Radiometric stations

In 2014, within the context of the DNICast Project (Direct Normal Irradiance Nowcasting methods for optimized operation of concentrating solar technologies, http://www.dnicast-project.net), a radiometric network formed by 19 stations covering a surface area of approximately 0.5 km2 was installed in the Plataforma Solar de Almería (PSA-CIEMAT); Figure 1 shows the location of these stations into PSA area.

For this work, we will use global horizontal irradiance data (GHI) registered in 7 of these stations.

Solar radiation has been measured using high quality Kipp & Zonnen thermoelectric pyranometers. The name, frequency of sampling, location and altitude of each of these experimental stations is shown in Table 1.

Fig. 1 Localization of the PSA radiometric stations

The configuration of the radiometric network allows us the creation of daily files with the registered data of each station (the variables registered are generally: horizontal global, diffuse and direct normal irradiance, temperature, relative humidity, wind speed/direction), all these files are uploaded automatically to one central server. However, in this study we will use the data of global horizontal radiation corresponding to the full year 2015. The data has been prepared for analysis in the 4 steps sequence described in the next subsections.

Table 1: Stations data

Name	Freq.	Lat.	Long.	Alt.
BSRN	60 s	37.092	-2.363	490.6
ARFRISOL	60 s	37.094	-2.357	499.6
DISS	5 s	37.098	-2.359	504.4
TSA	1 s	37.093	-2.357	499.1
KONTAS	1 s	37.095	-2.355	505.8
CESA1	60 s	37.095	-2.361	503.4
PSA-HP	10 s	37.091	-2.358	500.0

2.2. Preprocessing and temporal adjustment

Given the difference sampling frequencies of each station, as shown in Table 1, we have chosen the greater of them, which corresponds to a frequency on one sampling per minute.

Then, we proceed to synchronize each stating clocks, after observing some unexpected delays. The delay could be corrected, except for the case of the station DISS, that finally had to be discarded.

The next task was the delimitation of a time interval common for all the de dataset, minimizing the effect of the number the different daylight hours in different seasons. We opted for keeping the data in the time interval between 9 and 16, including the 9h and excluding the 16h. This reduces our dataset to 7h x 60min vectors per day, each vector of 6 dimensions, thus corresponding to 153 300 vectors per year.

2.3. Missing data

The second step consists of dealing with missing data. Fortunately, missing data constitutes a very small percentage of (<0.01%) our dataset. However, even these few missing values can produce a bad behavior or event avoid obtaining results when applying some forecasting techniques. The usual alternatives to correct missing data are, either to remove the incomplete rows, or to complete the missing values from the rest of the data. In our case the missing data correspond to particular values of some stations in some minutes, and we have decided that the better approach is to complete the values with the average of the rest of the working stations.

2.4. Clear Sky Models

The third step consist in applying a suitable *clear sky model*, that allows normalizing the irradiance values by correcting the apparent movement of the sun. In In our case, we have followed the method described in [10]. Figure 2 shows the graphical obtained after applying the model to the 24th of January of 2015, which corresponds to cloudy weather during the first hours, and to clear sky after the 12:30h. The vertical axis represents the *clear sky index* (k_{cs}), obtained as the ratio between the GHI value obtained by the sensor and the global horizontal clear sky irradiance at the hour of the day according to [10]. To facilitate the visualization, in the graphic the vertical coordinate of each station has been slightly changed in order to avoid overlapping.

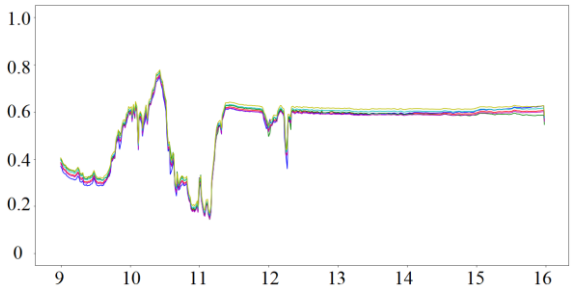

Fig. 2 k_{cs} for the 24th of January of 2015

2.5. Erroneous values

After the clear sky model has been applied, a fourth and last phase consisted of detecting anomalous values provided by the sensors in some particular cases, again replacing the erroneous values by the average value for this minute in the rest of the stations.

3. Cloud detection using clustering techniques

A first interesting analysis is to determine whether it is possible to user some unsupervised clustering technique to group similar data. We have found that in fact these techniques allow us to distinguish the presence the clouds in a very natural way.

3.1. Determining the number of clusters

The usual problem when employing unsupervised clustering techniques such as k-means is to choose the number k of clusters. One of the best known methods to determine this value is the index proposed by T. Caliński and J. Harabasz [11].

Initially, the dataset is represented as vectors in a n-dimensional Euclidean vector. In our case we have a space of n=6 dimensions, with each dimension corresponds to the value emitted by one station in a particular minute.

Then, the optimum number of clusters correspond to that data partition that minimizes the sum of the quadratic differences between the center of the cluster and their members (the so called *within-cluster distance*). The algorithm is implemented as part of libraries available for the most common data science languages Python (library *sklearn*), y R (library *vegan*). Both libraries yield as optimum the value k=2. For instance, using the language R, and assuming that the values of provided by the stations have been already loaded into the *dataframe* d:

```
fit ← cascadeKM(scale(d, center = TRUE,
                      scale = TRUE), 1, 10, iter = 1000)
calinski.best ← as.numeric(which.max(fit$results[2,]))
cat("K optimum:", calinski.best, "\n")
```

which shows the value 2.

3.2. Clustering process and interpretation of the results

The Caliński-Harabasz index provides the optimum number of clusters, but not the centers of each one. However, once determined that k=2 we can employ the k-means algorithm to determine the center of each cluster. In the language R we can write simply *kmeans(d,2)*, obtaining the results of Table 2.

Table 2: Cluster centers for each station

	Cluster 1	Cluster 2
ARFRISOL	0.259	0.682
CESA	0.264	0.679
PSA	0.269	0.685
TSA	0.268	0.690
KONTAS	0.277	0.703
BSRN	0.286	0.703

The first column corresponds to the name of each of the stations, while the other two columns indicate the center value for the associated dimension.

The natural interpretation is that the first cluster, with low irradiation values, corresponds to the irradiation obtained in cloudy conditions. Then, the second correspond to the standard values of clear sky conditions. Observing the frequencies histogram for each station confirms this impression. In particular, Figure 3 contains the histogram associated to the station ARFRISOL.

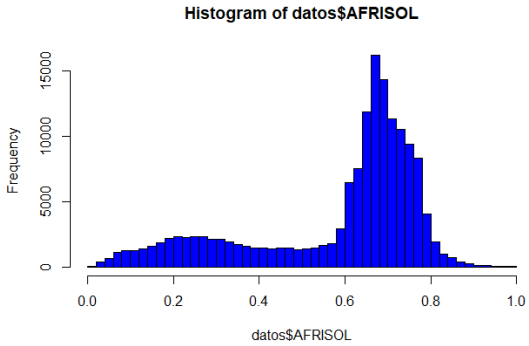

Fig. 3 Frequency histogram for the irradiations associated to the station ARFRISOL

The figure corresponds to the combination of two probabilistic distributions. The center for ARFRISOL in the first cluster (0.259) corresponds roughly to the mean of the distribution situated at the left, with lower irradiation values, that represents the presence of clouds. However, the second distribution, at the right, indicates clear sky, which is more frequent in the geographical situation considered (south of Spain).

4. GHI nowcasting

The technique described in the previous Section can be employed to detect automatically the presence of clouds and constitutes the first contribution of the paper. In the rest of the work we explain our proposal for the short term forecasting of the global horizontal irradiance.

4.1. Stationarity

A common preprocessing stage in the case of temporal series is to check if it is stationary, that is if its properties are independent of the particular moment of the series examined [12]. In order to check this we have employed the Dickey–Fuller (ADF) technique [13]. In our case, the technique indicates that indeed we are in the presence of a stationary time series, either if we consider the whole year, one month or even a single day.

Stationary temporal series are more difficult to predict, and thus it is convenient to try to transform the series to avoid stationarity if possible. The most common method to achieve this is to use the differences technique [14], which replace each value by its increment, positive or negative, with respect to the previous data. If we apply the technique on a daily basis we lose one datum per day (the first in the morning, whose difference with respect to the

last value of the previous day makes no sense), but this is a very small loss of data that can be admitted.

After this transformation, the Dickey-Fuller technique indicates that now the values are stationary, more clearly when studying each day separately and by a small margin when considering the whole dataset. Thus, we have decided to work with each day data separately in the experiments described in the rest of the section.

4.2. Direct methods

We start by employing to simple, albeit powerful methods, employed usually when forecasting temporal series.

- Naïve predictor: The prediction is just the last known value.
- Average: The prediction is the average of the last known values in some time interval (one hour in our experiments).

Often, especially with highly stationary values, these simple methods are the most effective for short-term forecasting. This is the case of our proposal, where we consider time horizons of a few minutes. In particular, we have found that the naïve prediction is very difficult to beat, as we will see in the experiments section.

4.3. ARIMA

The Auto Regressive Integrated Moving Average (ARIMA) models [13], use a combination of the last p previous values of the variable we wish to predict, corrected with the errors observed in the q previous predictions. Usually a third parameter d, indicating the number of differences to perform is added to the parameters p and q. This number of differences is devoted to convert non-stationary data into stationary and indicates the number of times that the differences method mentioned on subsection 4.1 must be applied. For computing the best values of (p,d,q) we have employed the library *forecast* of the R language, and, in particular, the method *auto.arima*, obtaining p=5, d=1 and q=0 has best values for predicting with ARIMA in our dataset.

4.4. LSTM neural networks

An artificial neural network (ANN) [15], is a computation mechanism based on *artificial neurons*. Each neuron receives a input signal, which is processed and resent to other interconnected neurons.

The number of neurons in the network, the topology of the connections graph, and the processing function attached to each neuron, are decided during the design of the network. However, both the functions and the connections have *weights* that will adjusted automatically during the learning process. The adjusted network is then considered a *model* that can then use to predict values.

The network employs a *training dataset* to adjust these weights, which are in our case the data collected by the stations in the minutes previous of to the forecasting moment. The tuning of the weights tries to minimize some cost or loss function that relates the outputs produced by the neural network and the expected output, which is our case the prediction for a certain future horizon in minutes.

In our case, we have programmed the network using the Keras Python library (https://keras.io/), which works over the library TensorFlow (https://github.com/tensorflow/tensorflow). Keras includes a large number of cost functions (parameter *loss*). Among all of them we have chosen the function '*mean_squared_error*', because we use this measurement (in particular the RMSE) to determine the error of each method in our experiments.

The neurons of an ANN are organized in *layers*. Each layer applies some transformation on its input and passes the output to the following layer. Thus, the data traverses the ANN from the *input* to the *output* layer, possible after traversing a certain number of *hidden* layers.

Our ANN starts with an input layer with a single input signal (we have a model for each station, which currently are treated separately), and just one hidden layer, of the type known as LSTM (Long Short-Term Memory networks) [16]. This type of recurrent (that is, with cycles) neural network is often employed for forecasting temporal series [17].

The number of neurons in the hidden layer depends on the temporal horizon considered. In our experiments we have found that a LSTM with four neurons provides good results.

Figure 4 shows an example of LSTM for a time horizon of 11 minutes, from 11:49 to 12 of the 24th of January of 2015. The neural network has been trained using the data between 9 and 11:48 hours of the same day.

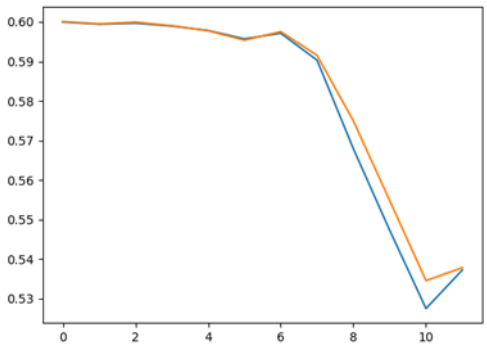

Fig. 4 LSTM prediction for the 11 minutes before the 12h of the 24th of January of 2015. The orange line represents the prediction and the blue line the real values

5. Experimental results

Table 3 shows the results of our experiments

Table 3: RMSE of the LSTM, together with % of increment of RMSE of the rest of the methods with respect to LSTM

f (min)	LSTM (RMSE)	AVG (% inc)	Naïve (% inc)	ARIMA (% inc)
1	0.054	6.59	7.08	7.10
2	0.070	7.46	2.03	1.17
3	0.093	2.31	1.14	1.27
4	0.11	3.72	1.05	1.15
5	0.12	3.02	1.03	1.16
10	0.15	2.47	0.96	1.17
20	0.21	1.37	0.93	1.02

The first column shows the different temporal horizons in minutes. The second column shows the RMSE of the LSTM neural network. The remainder columns show the % of increment of the rest of the methods when compared with the RSME of the LSTM. In particular, the third column includes the error increment of the average method (AVG), which varies between 7.46% (prediction for 2 minutes) and 2.47% (prediction at 10 minutes) with respect LSTM. The fourth column includes the increment of error for the Naïve method. This simple method is the best for time horizons over 10 minutes. However, in the first 10 minutes it is overcome by our proposal, the LSTM network, although just by 3% after only five minutes. The prediction based on ARIMA increments the error in the first 10 minutes analogously to the Naïve method and only improves the efficiency of LSTM after 20 minutes.

The data have been obtained as the average of applying the techniques to all the available days (365). For each day, the result considered is the average of three experiments:

- GHI prediction at minute 12h+f, taking a model created with data between 9 and 12 hours (with f the temporal horizon in minutes taking the values of Table 3).
- GHI prediction at minute 13h+f from the model build with the data obtained between 12 and 13 hours.
- GHI prediction at minute 15h+f from the data collected between 13 and 15 hours of the same day.

We have checked that the results do no experience any noticeable variation when considering training data of more than one hour. That is, there is no improvement after accumulating, for instance, three hours instead of one hour of historical data to train the LSTM, ARIMA or the AVG method. This is positive, because it implies that accurate predictions can be obtained from 10am every day. However, with training values under one hour the accuracy of the prediction decreases.

Summarizing, the table shows a total of 365 (days) x 3 (predictions per day) x 6 (stations) = 6570 tests.

Since the differences in the error are small in some cases, we applied a Wilcoxon [18] test. The results indicate that the differences in RSME shown in the table correspond to statistically significative differences with $p<0.001$.

To finish this section, and in relation to our initial goal of obtaining an efficient predictive method in terms of computation time, we must point out that the models, with one hour of training data, are generated each 23 seconds on average, using a computer DELL XPS 13 9350, with 4 CPUs at 2.20 GHz and 16 gigabytes of RAM, thus satisfying our requirement of generating the models in less than one minute.

6. Conclusions

In this paper we have considered the problem of forecasting the global horizontal irradiance in very short time using a particular dataset obtained from the *Plataforma Solar de Almería* (PSA-CIEMAT) during one year. We have checked that using LSTM networks is possible to improve, albeit by a narrow margin, the prediction based on repeating the last known value, a simple but very effective method for the first few minutes that improves the results of other usual techniques such as the use of the average or other more complex methods such as ARIMA.

Regarding future work, it must be pointed out that our neural network only considers each station individually, that is, the prediction is based on the data collected from the same station in the previous hour. We think that the development of multidimensional models that employ the data from all the stations to predict the GHI of any of them can result in more precise predictions. The intuition is that the irradiation in some station can be anticipated for the irradiation of nearby stations where, for instance, a cloud as already started to be noticed. The computation time required by these, more complex networks, will more likely require also some more complex hardware solutions in order to obtain the models before the next prediction arrives.

Acknowledgements

This work has been partially supported by the Spanish MINECO project TIN2015-66471, and by the Santander-UCM project PR26/16-21B-1.

Competing interest

The authors declare that no competing interests exist.

References

[1] D. Renne, *Semi-Annual Status. Task 36: Solar. Resource Knowledge Management*. Solar Resource Knowledge Management. 2009.

[2] E Lorenz, A Hammer, and D Heinemann. *Short term forecasting of solar radiation based on satellite data*. In EUROSUN 2004 (ISES Europe Solar Congress), pages 841- 848, 2004.

[3] Bosch, J. L. y Kleissl, J. Cloud motion vectors from a network of ground sensors in a solar power plant. Solar Energy 95(1), 13-20, 2013.

[4] G. Reikard. *Predicting solar radiation at high resolutions: A comparison of time series forecasts*. Solar Energy, 83(3):342-349, 2009.

[5] Martín, L., Zarzalejo, L. F., Polo, J., Navarro, A., Marchante, R. y Cony, M. Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning. Solar Energy 84(10), 1772-1781, 2010.

[6] C. Paoli, C, Voyant, M. Muselli, and M.L. Nivet. *Forecasting of preprocessed daily solar radiation time series using neural networks*. Solar Energy, 84(12), 2146-2160, 2010

[7] A. Mellit and A. M. Pavan. *A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy*. Solar Energy, 84(5), 807-821, 2010.

[8] M. Bou-Rabee, S. A. Sulaiman, M. Saleh, and S. Marafi. *Using artificial neural networks to estimate solar radiation in Kuwait*. Renewable and Sustainable Energy Reviews, 72, 434-438. 2017.

[9] M. Diagne, M. David, P. Lauret, J. Boland, and N. Schmutz. *Review of solar irradiance forecasting methods and a proposition for small-scale insular grids.* Renewable and Sustainable Energy Reviews, 27, 65-76. 2013.

[10] I. A. Walter, R. G. Allen, R. Elliott, M.E. Jensen, D. Itenfisu, B Mecham, ... and T Spofford. *ASCE's standardized reference evapotranspiration equation.* In Watershed Management and Operations Management 2000 (pp. 1-11). 2000.

[11] T. Caliński, and J. Harabasz *A dendrite method for cluster analysis*. Communications in Statistics-theory and Methods, 3(1), 1-27. 1974

[12] D. Kwiatkowski, P. C B Phillips, P. Schmidt, and Y. Shin. 1992. *Testing the Null Hypothesis of Stationarity Against the Alternative of a Unit Root: How Sure Are We That Economic Time Series Have a Unit Root?* Journal of Econometrics 54 (1-3): 159–78. 1992.

[13] R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts. 2018.

[14] S. Makridakis, S.C. Wheelwright, and R.J. *Hyndman. Forecasting: methods and applications*. John Wiley & Sons, 1998.

[15] S. Haykin. *Neural Networks: A Comprehensive Foundation Upper*. Saddle River. NJ, USA, pp. 1–842. 1998.

[16] S. Hochreiter and J. Schmidhuber. *Long Short-Term Memory.* Neural Computation. 9 (8): 1735–1780. 1997

[17] F. A. Gers , J. Schmidhuber and F. Cummins. *Learning to Forget: Continual Prediction with LSTM*. Neural Computation Volume 12-10. p.2451-2471. 2000

[18] F. Wilcoxon. *Individual Comparisons by Ranking Methods*. Biometrics 1, 80-83. 1945