Hourly solar radiation forecasting on SAURAN network datasets using deep learning method: La Reunion and Durban cases study

Mathieu Delsaut, Claire Quatrehomme, Patrick Jeanty, Miloud Bessafi, Jean-Pierre Chabriat

Hourly solar radiation forecasting on SAURAN network datasets using deep learning method: La Reunion and Durban cases study

Mathieu Delsaut[1], Claire Quatrehomme[2], Patrick Jeanty[1], Miloud Bessafi[1], Jean-Pierre Chabriat[1]

[1] LE²P laboratory, University of La Reunion; firstname.lastname@univ-reunion.fr

[2] INSA Lyon; firstname.lastname@insa-lyon.fr

Abstract

The purpose of this article is to describe the data pre-treatment (smoothing, normalizing) and present hourly forecasting method using XGBoost deep learning tool on the global horizontal irradiance (GHI). This method will be applied on two sites with different typical meteorological profiles. An estimation of prediction skills will be given and discussed against classical persistence model.

Keywords: solar forecasting; deep learning; XGBoost; pretreatment; SAURAN; solar field.

1. Introduction

For over a period of 8 years now, the LE²P lab (University of La Reunion) has been deploying a ground base solar stations network mainly on Reunion Island but also on neighboring islands (Mauritius, Rodrigues). Some of those stations became part of the Southern African Universities Radiometric Network (SAURAN) within the context of EU funded programs based on cooperation between University of KwaZulu-Natal and University of La Reunion. These datasets mainly coming from SPN1 pyranometers, include global and diffuse irradiances that are collected every minute together with meteorological data.

Solar forecasting is important to enhance the part of renewable energies in the energetic mix. Main power photovoltaic producers are required to give an estimation of their projected output. Such approach is therefore a solution to compensate for the intermittency of this source of energy.

2. Irradiance forecasting

Solar forecasting prediction methods are the subject of many articles in scientific literature worldwide. It helps to determine which method to focus on to build a new prediction process, more accurate than linear regression. Going through several papers which compare different forecasting methods [6] [7] show there are 3 main types: satellite derived models, parametric models and machine learning models.

Parametric models are based upon meteorological measures, for instance temperatures and precipitation [6], in order to forecast solar radiation. Satellite-based models use irradiance measured by satellites to forecast GHI [7] (GHI: global horizontal irradiance).

Machine learning is by far the most used method today. It is mostly due to the exponential growth of computer sciences. Therefore, this research work focuses on machine learning methods.

Many specialized books [8] [9] explain machine learning science. It displays generalized information about methods such as artificial neural network (ANN) and naive models. It is interesting to understand the general meaning of models. However, some scientific articles display the use of these methods in the specific case of solar radiation and are therefore more useful.

2.1 Solar forecasting in insular context

Lauret et al.'s article [10] is particularly interesting because it compares different machine learning methods used to forecast solar radiation in an insular context. Precisely, a set of machine learning methods have been tested to forecast GHI in three different places: Corsica, Reunion and Guadeloupe. This experiment used a year of training data, registered by pyranometers, and another year of data to test the prediction methods. The tested methods to forecast solar radiation were:

- Linear regression,
- 2 naive models, one based on the GHI persistence and the other on the clear sky index persistence,
- Machine learning methods: neural network, Gaussian processes, support vector machines.

The conclusion was that machine learning methods slightly improve prediction, compared to linear regression and naive methods. This improvement is more important when the weather is often unstable, as is the case in the study spot of Guadeloupe.

The limit of this paper is that the chosen spots of pyranometers on the three islands are really spatially limited. For example, the spot on Reunion is in the city of Saint-Pierre, in the south of the island. Saint-Pierre has a really steady weather, the sky remains clear all day, so machine learning methods in that case do not improve solar forecasting much.

However, Reunion Island has many micro climates, depending on orientation and altitude.

Mainly the east coast is rainy and the west coast has a more steady and fine weather. So the conclusion found in Saint-Pierre (southwest) cannot be generalized to the whole island. Therefore, machine learning methods are worth being applied on LE^2P's solar stations.

2.2 Importance of data

Another article, by Voyant et al. [11], also compares several machine learning methods applied to solar forecasting. Crossing it with the Lauret et al.'s article [10] highlights an important point: data must be carefully chosen and prepared. Indeed machine learning methods are based on the learning of data, so their preparation is as important as the chosen method to get an accurate prediction.

First coded model was a linear regression and used mean GHI values as inputs. This hourly-average is too crude and has to be refined. Several data smoothing methods exist: moving average, exponential smoothing, normalization per clear-sky value are some of them.

The choice of a smoothing method has to be made in regard to raw data. It has to be adapted to it. Testing several smoothing methods and their impact on prediction accuracy enables to choose one of them.

2.3 Forecast by supervised learning models

Among machine learning methods, this paper found interest on XGBoost (extreme gradient boosting) model. It is used for supervised learning and has recently been dominating the machine learning field.

Basically, XGBoost learns sets of data as examples and builds a model, which is a black box. This model is then used to predict data, from new data sets input.

XGBoost is an algorithm which has been developed from gradient boosting tree model developed by Friedman et al [12]. It is widely used in the Kaggle community [13]. Kaggle is an exchange platform for people who work on data science and machine learning. It also organizes machine learning competition, where XGBoost has been used in many winning solutions.

Methods used to forecast solar radiation displayed in articles are really redundant, and there are few examples of the use of XGBoost for this specific application.

Urraca et al.'s article [6] proposes an example of using the XGBoost model to forecast solar radiation. Although the aim of their research work was to forecast solar radiation spatially, and not temporally, it also used data registered by pyranometers.

In this article, XGBoost was compared to other

models, such as parametric models and satellite-base models. The article concludes that XGBoost is an effective model but its weakness is that it produces point predictions, i.e. limited to a small spatial area. It is recommended to use XGBoost with on-ground measurement.

As we wants to produce temporal forecasts of the GHI received by each pyranometer, point predictions are in its case not a problem. Moreover, pyranometers produce on-ground measurement. Then it seems pertinent to try an XGBoost model on GHI data in order to yield solar forecasting.

Eventually, the chosen model was supervised learning, and more precisely using XGBoost algorithm. This algorithm uses gradient boosted decision tree to build prediction models from sets of data [12]. However, the XGBoost model has not been applied yet to temporal solar forecasting.

3. Data pretreatment

Raw GHI data induce too much noise on the signal, and therefore cannot be used as input to prediction models, mainly because of over fitting. Figures 1 and 2 display samples of a noisy GHI signal (grey signal on both plots).

It is then necessary to smooth the data, but not grossly as their general perturbations must be kept. There are mainly two ways to smooth data: averaging and normalization.

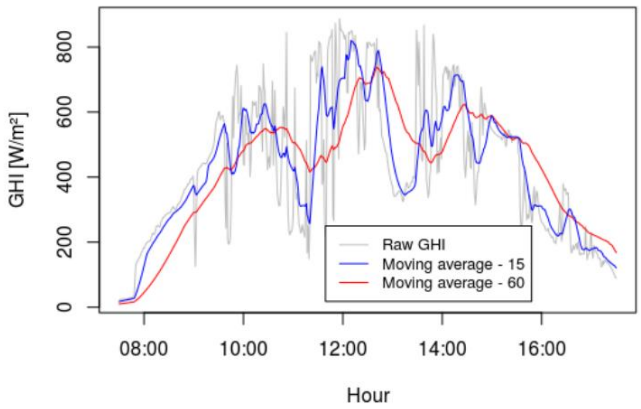

Fig 1. Two different moving average sets on raw GHI data

3.1 Moving average

GHI value for each minute is calculated by averaging GHI raw values from (t-n) minutes to t time. For instance, 10-minutes-moving average at 8 a.m. is calculated by averaging raw values per minute from 7:51 a.m. to 8:00 a.m.. An example of moving average taking different amounts of values applied on raw GHI values is displayed by figure 2.

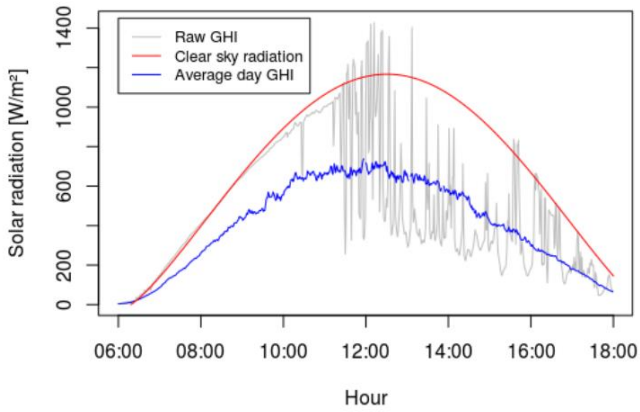

Fig 2. Perturbed day, clear sky index and average day

Moving average is quite adjustable, because it is possible to choose how many values the average is calculated from, and adjust how precise the smoothing is going to be. However, it introduces a delay in the signal, because the average is done with previous values. This delay can be observed on figure 1. It is possible to calculate a moving average as a centered average or a forward average but it does not really make sense when applied to a prediction. As a matter of fact, the next values are not supposed to be known.

3.2 Normalisation

Raw data are divided by other values. These other values can be average day GHI values of the month or clear sky values.

3.2.1 Using average day values

An average day is calculated for a location and a month. It is a day of GHI values, calculated by averaging for each minute all GHI values in the

dataset. An example of a typical day is displayed in figure 2. For each minute, GHI value is the average of raw GHI value for this minute from 2013-02-01 to 2017-02-28.

If raw data like those displayed in figure 2 are normalized by a typical day, such as the one on figure 2, the normalized data now looks like figure 3. It displays the difference between the average values and the raw GHI values of the day.

3.2.2 Using clear sky model values

A clear sky model can be calculated for each location by a model, coded in R. This model depends on GPS location, i.e. longitude and latitude, and time zone. It provides solar radiation values during a day without any disturbance (clouds). Clear sky radiation is the total radiation before it parts in direct radiation and diffuse radiation. It is therefore pure direct radiation. Figure 2 displays an example of a clear sky day (red plot).

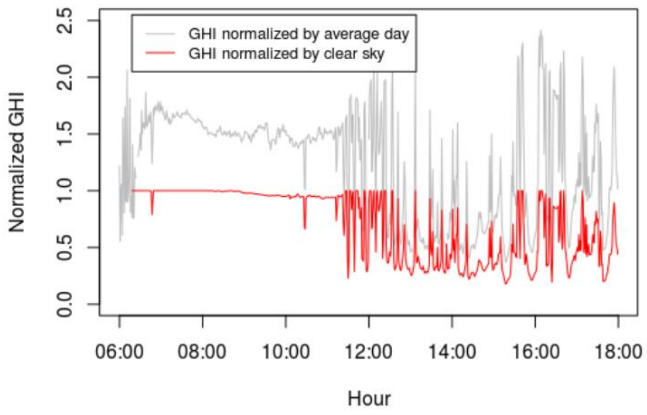

Fig. 3 Normalized GHI

It is interesting to notice that some points show raw GHI above clear sky radiation. Clear sky model displays an ideal solar radiation, without any disturbances. This ideal radiation can meet obstacles and then divide in normal radiation and diffuse radiation. Combination of both makes GHI.

Therefore, theoretically, GHI cannot read above clear sky radiation. When it is the case, it is due to a rapid change in the sky. The sensor is suddenly flooded by direct solar radiation and this brutal change leads to an over evaluated GHI measure.

To address this problem, in normalizing raw data by clear sky radiation, all values above 1 are set to 1. Such example is displayed by figure 3 (red signal).

Normalization presents the advantage not to introduce delay, unlike moving average. Moreover, it frees GHI data from temporality. This means that GHI data from 7:00 a.m. to 10:00 a.m. and GHI data from 2:00 p.m. to 5:00 p.m. are differentiated only by the change of disturbances in the sky, and by their difference from an average day or clear sky model. The fact that GHI tends to increase in the morning and decrease in the afternoon is erased.

This is an advantage because it reduces the complexity of prediction. With normalized data as input, XGBoost library has only to learn divergences from the normal, and not the increase or decrease due to the sun's path through the day.

However, clear-sky model presents the inconvenient not to give solar radiation values until the sun is up. For instance, on Reunion Island, clear sky model begins to give value from 7:00 a.m., although in practice the sun begins to rise at 6:30 a.m. It is not possible to normalize raw GHI values as long as the clear sky model does not give solar radiation values. Therefore, the desired GHI prediction hour cannot be too early in the morning, or too late in the afternoon. Obviously, the output is also a smoothed value, and therefore it has to be treated, by multiplying it per its day-type or clear-sky value.

4. Prediction models

4.1 Deep learning XGBoost

XGBoost is a recent machine learning model of supervised learning [14]. It learns from sets of data to build models. These models are then used to yield predictions. It is therefore purely empirical.

Technically, XGBoost model divide a set of data in several branches, as it uses decision trees.

Mathematical methods are used to boost these

decision trees, in order to accelerate the resolution. XGBoost also corrects the error it commits when dividing the data set, and therefore has the ability to correct its own errors.

Supervised learning is about learning how to build a link between inputs and desired outputs from examples. In the end, the model has to be able to generalize from training examples to unseen data, in order to provide predictions.

4.1.1 Providing data

The aim is to predict one GHI value from three hours of GHI values, one per minute. So XGBoost is given a table of data of 180 (3 × 60) values. These GHI values can either be raw or smoothed by data processing, in order to improve prediction.

The inputs have to be data that must be learnt to build the model. Inputs are all similar data from the same location, the same month and the same time frame. They do not include GHI data of the day of the desired prediction.

The data set is therefore a matrix of 180 columns, with nearly a hundred rows. These data are called training data and form a set of training examples.

Once the model is built, the data used as input are the three hours GHI values preceding the beginning of the prediction time. For example, if the prediction begins at 10 a.m., input data are GHI values from 07:01 a.m. to 10 a.m.. Therefore the input data are simply a row of 180 data, which constitute a test set.

4.1.2 Model building

The XGBoost model is built by learning training data. By learning these examples, it builds a function, also called a model. This function has to be able to generalize from learnt examples in order to predict values from unknown inputs.

Some parameters can be adjusted manually, for example the number of iterations in the learning process. It is interesting because iterations take time, and therefore it is useless to keep learning from training examples if it is no longer necessary.

To determine the optimal iteration, from which the learning can stop, XGBoost displays a cross validation device. It performs a cross validation on the training set, more precisely a leave-one-out (LOO) validation, for each iteration, and calculates the associated root mean square error (RMSE). During the first iterations, the RMSE decreases. The more XGBoost learns, the more accurate is the model. But after a certain iteration, the RMSE begins to increase.

This phenomena is called overfitting. It is due to the fact that if the algorithm learns too many examples, and so it gets tougher for it to generalize from them. It learns the noise of the signal rather than the general outshape of the signal. Therefore it is better not to learn too many examples, and to stop iterations when the RMSE is beginning to increase. The cross validation device of XGBoost gives the number of iterations from which the RMSE increases, which is the optimal iteration number.

This optimal iteration number is then used as a parameter in the construction of the model.

The XGBoost algorithm learns from the training set until it reaches the optimal iteration number, then stops. The model is then ready to be used.

4.1.3 Example

The desired GHI prediction is in Durban, May 5th, 2017, from 10 a.m. to a 30-minute time limit, .i.e. 10:30 a.m.

First, XGBoost is given a data table, with one GHI value per minute from 07:01 a.m. to 10:00 a.m., for each day of May in Durban stored on the SAURAN website. It is also given, for each day, the GHI value at 10:30 a.m.. To avoid testing prediction method on already known data, data of the year 2017 are not included in this training set. XGBoost learns these examples and builds a model.

Then, the desired prediction is a GHI value at 10:30 a.m. from 10:00 a.m. on May 5th, 2017 in Durban. A data set with GHI values from 07:01 a.m.

to 10:00 a.m. for this day is gathered. The XGBoost model is applied on this test set. It yields a GHI value for Durban, at 10:30 a.m., on May 5th, 2017.

4.1.4 Constraints

It appears that if there is only one general method, applicable to all datasets at any time, there are nevertheless many models. Actually, there is a model per location, per month, per starting time of prediction and per time-frame limit. It represents a huge amount of models.

A model can take long calculation time, however once calculated, the use of it to generate a prediction is quasi immediate. The calculated XGBoost models then have to be stored, in order to avoid a long time calculation each time a prediction is asked. There is no point in giving GHI prediction for the next minute if the model takes 3 minutes to be calculated.

Another constraint is the size of the input data set of the method. If the model has been built on three hours of data, that is 180 values, then the input has to be 180 values long too. It is due to the fact that the supervised learning from training examples is quite strict, if the model has learnt to predict 1 value from 180 values, it cannot operate any differently.

Overfitting is another constraint inherent to supervised learning model. XGBoost partly avoids it by using cross validation to determine where to stop the learning process. However, a noisy signal tends to create overfitting. Therefore, it can be useful to smooth data, in order to avoid noise and overfitting.

4.2 Comparison with linear regression model

First, XGBoost prediction has been compared to linear regression prediction, in order to evaluate its performance. Nevertheless, it has been tested on smoothed GHI data, to avoid overfitting and improve the prediction for both methods.

The smoothing method used to apply these tests is moving average (GHI value for each minute is an average of a certain number of raw GHI values). Smoothing methods choice has to be carefully tested.

4.2.1 Methodology

Empirical tests were driven to compare prediction given by the two methods. The same input data sets are given to the methods in order to build either a linear regression or an XGBoost model. Then, the same test set is given to each method to yield a prediction. These two predictions are eventually compared to the actual GHI value. If the inputs are average data, the output are compared to an average GHI value.

This comparison is made by calculating RMSE and R^2. These criteria assess the accuracy of the prediction model.

These tests have been applied to Durban (South Africa) and Le Port (Reunion Island) locations, for different months in the year, for different time frames and for different smoothing. Indeed a moving average can take into account different amounts of values in order to calculate the average GHI value.

4.2.2 Results

Some results of tests have been selected and are displayed in tables 4.1, 4.2 and 4.3. It displays the results for Durban and Le Port. For each prediction, the RMSE has been calculated in a time frame between 9:00 a.m. to 4:00 p.m. for each day of a month, and then the average RMSE has been calculated.

Table 4.1 displays results of prediction made from raw GHI data. Table 4.2 and 4.3 results are calculated from predictions made from smoothed data. The smoothing method was a moving average using 10 previous values (table 4.2) or 60 previous values (table 4.3) to calculate one average GHI value per minute.

It means, for example, that input GHI value at 10:15 a.m. is the average of 10 raw values, one per minute, from 10:06 a.m. to 10:15 a.m.

Location Month	Time horizon (min)	Linear regression	XGBoost
Durban December	+30	1 092	212
	+60	1 108	227
Le Port May	+30	901	158
	+60	1 153	166

Table 4.1: Mean RMSE (Watt/m²) for prediction tests on raw GHI data

Location Month	Time horizon (min)	Linear regression	XGBoost
Durban December	+30	238	175
	+60	248	192
Le Port May	+30	463	137
	+60	465	144

Table 4.2: Mean RMSE (Watt/m²) for prediction tests on smoothed GHI data – 10-minute moving average

Location Month	Time horizon (min)	Linear regression	XGBoost
Durban December	+30	110	102
	+60	112	143
Le Port May	+30	22	80
	+60	22	118

Table 4.3: Mean RMSE (Watt/m²) for prediction tests on smoothed GHI data – 60-minute moving average

5. Conclusion

We observe a significant improvement when using deep learning method compared to persistence model on both sites when smoothing is not too strong. It turns out that machine learning surpasses persistence and allows to yield a prediction with an error (RMSE) of about 200 W/m².

We show the importance of pretreatment time series using XGBoost model. We highlight the flexibility of this method on two tropical and subtropical sites with different weather and cloud covering.

The next step will be to extend pre-treatment and study the impact of including meteorological data as predictors. A future work would be also to implement this method on other time forecast horizons.

Acknowledgements

The different project conducted by LE²P lab from the University of La Reunion Island are mainly supported by European funding that has become essential for the development of renewable energies in La Reunion. Such ERDF programmes typically involve EU, Reunion Regional Council and French Government respectively up to 60%, 20% and 20% approximately.

LE²P lab particularly appreciate collaboration with University of KwaZulu-Natal and thank SAURAN team for providing data.

References

[1] Urraca et al. Estimation methods for global solar radiation: Case study evaluation of five different approaches in central Spain. *Renewable and Sustainable Energy Reviews*, 77:1098–1113, September 2017. doi:10.1016/j.rser.2016.11.222.

[2] Martín, L., L. Zarzalejo, J. Polo, A. Navarro, R. Marchante, et M. Cony (2010). 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.

[3] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5):1189–1232, 2001. doi:10.1214/aos/1013203451.

[4] Wikipedia. Root-mean-square deviation, June 2017. Page Version ID: 783262258. URL:https://en.wikipedia.org/w/index.php?title=Root-mean-square_deviation&oldid=783262258.

[5] Wikipedia. Coefficient of determination, July 2017. Page Version ID: 790301456. URL: https://en.wikipedia.org/w/index.php?title=Coefficient_of_determination&oldid=790301456.

[6] Urraca et al. Estimation methods for global solar radiation: Case study evaluation of five different approaches in central Spain. Renewable and

Sustainable Energy Reviews, 77:1098–1113, September 2017. doi:10.1016/j.rser.2016.11.222.

[7] Inman et al. Solar forecasting methods for renewable energy integration. Progress in Energy and Combustion Science, 39(6):535–576, December 2013. doi:10.1016/j.pecs.2013.06.002.

[8] Eric Biernat and Michel Lutz. Data science : fondamentaux et études de cas. Machine learning avec Python et R. Eyrolles, 1 edition, October 2015.

[9] Stéphane Tufféry. Data mining et statistique décisionnelle. L'intelligence des données. Technip, 4 edition, August 2012.BIBLIOGRAPHY 47

[10] Lauret et al. A benchmarking of machine learning techniques for solar radiation forecasting in an insular context. Solar Energy, 112:446–457, February 2015. doi: 10.1016/j.solener.2014.12.014.

[11] Voyant et al. Machine learning methods for solar radiation forecasting: A review. Renewable Energy, 105:569–582, May 2017. doi:10.1016/j.renene.2016.12.095.

[12] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 29(5):1189–1232, 2001. doi:10.1214/aos/1013203451.

[13] Kaggle: Your Home for Data Science. URL: https://www.kaggle.com/.

[14] XGBoost Documents. URL: https://xgboost.readthedocs.io/en/latest/.