Conference Proceedings

EuroSun 2016

Palma de Mallorca (Spain), 11 – 14 October 2016

Short-term solar irradiance prediction using Time series analysis and Neural Networks for Green Energy Park Photovoltaic Plant.

Abdelkrim Bouabbou[2]*, Abdellatif Ghennioui[1], Sébastien Vaudreuil[2] and Zakaria Naimi[1]

[1] The Research Institute for Solar Energy and New Energies IRESEN – Rabat, Morocco

[2] The Euro-Mediterranean University of Fez - Morocco

Abstract

Short-term solar irradiance forecasting is of great importance for optimal operation and grid integration of photovoltaic (PV) plants. Morocco's Green Energy Park (GEP) of Benguerir is investigating the possibility of providing accurate solar forecasts for its own PV plant. However, to overcome the variability of solar resources under changing weather conditions, a reliable forecasting model is needed. In this context, this work provides a detailed research in short-term solar irradiance forecasting, an implementation and a technical analysis were performed in this study for a suitable and a reliable Global horizontal irradiance (GHI) forecasting model. Comparing the performance of different models such as ARIMA, Multiple linear regression, curve fitting and artificial neural networks, gave valuable information for further development of the final model. This final model is based on NARX ANN. In order to achieve meaningful results when comparing these different approaches as well as for NARX ANN model assessment, a standardized methodology for evaluation is used. NARX model outperformed all the approaches carried out in this study and it gave very satisfying results of 4,57 % MAPE error for Clear-sky and 14,67% MAPE error for Cloudy weather conditions.

Keywords: GHI, Forecasting, *ARIMA, MLR, Curve Fitting, ANN, NARX*

1. Introduction

Solar power generation is gaining substantial prominence worldwide, especially Photovoltaic (PV) power technologies, with an average yearly growth rate of 50% over the last ten years (Jäger-Waldau, 2015), making PV one of the fastest growing industries at the present. However, while PV output is dependent on solar irradiance, natural intermittency and variability of solar resources poses limitations for a more stable grid integration. Furthermore, this variability of solar technologies output at higher grid penetration levels induces problems economically, regarding dispatchability, generation, and grid reliability in general (Rich H. Inman, 2013). While energy storage in large amounts on-site is very problematic, this limitation can be tackled by accurate solar irradiance forecasting, which is crucial for efficient integration of solar energy into the electric power grid and for improving its proportion in total energy consumption.

Solar irradiance at the ground level are highly variable mostly due to weather related variables like clouds coverage variability and their motion, and to a lesser extent aerosols and gases in the atmosphere. However, thanks to solar meteorology for renewable generation, a number of promising approaches for prediction have been developed in the past few years, such as Remote sensing based Numerical weather prediction (NWP) methods (Diagne, 2013) (Lima, 2016), Regression methods, Machine learning and artificial intelligence (AI) forecasting using data acquired by local sensing instrumentation such as total sky imager (Chi Wai Chowa, 2011) , pyranometers and local weather instrumentation (Rich H. Inman, 2013).

Regressive methods are not new to forecasting, beginning with pre-stochastic algorithms like Multiple Linear Regression (MLR), and Curve Fitting (CF) to newly developed nonlinear stochastic algorithms. Hard computing techniques like MLR and Curve fitting predict different types of data, the authors of (Jain, 2012) have carried out these techniques on variables like temperature, humidity and "day type" parameters to recognize load patterns in short-term load forecasting. Also, authors in (Sahin, 2012) have compared MLR based model with Artificial Neural networks (ANN) for estimating solar radiation, and they are generally very simplistic while maintaining the ability to model non-linear patterns. Nonetheless, a lot of models were developed based on Box-Jenkins model ARMA, such as ARIMA (Zhang, 2003) taking into consideration external information to the time-series under analysis. Also we can find many variations of such model like ARMAX and ARIMAX with the same philosophy.

doi:10.18086/eurosun.2016.09.02 Available at http://proceedings.ises.org

For solar radiation, many works have incorporated these methods like (Aguiar RJ, 1989) used clearness index K_t to model daily sequences of hourly irradiance with ARMA techniques, where K_t was obtained by multiplying a clear sky index values by a non-stationary fluctuation, with probability depending on the hour of the day.

Since the late 1990s, ANNs have seen an increase in solar forecasting applications. Several works proposed in literature modeled solar irradiance using various ANN's internal topologies, different inputs and for several time scales (Lopez, 2005) (Koca, 2011) .Al-Alawi and Al-Hinai used weather variables as inputs to an ANN to predict monthly values of Global Horizontal Irradiance (GHI) over a year (Al-Alawi S, 1998). Authors in (A. Di Piazza, 2013) have acquired very satisfactory forecasts using a feedforward Time delay Neural Networks called NARX. In addition, In the work of (Vaz, 2015) PV output forecast have been generated using NARX by employing meteorological variables as exogenous inputs and assed the influence of multiple PV neighboring systems.

In this context, a thorough methodology was undertaken for short-term global horizontal irradiance forecasting using local meteorological measurements. This paper is organized as follows. In section 2, the study overview and experimental data collection were presented. The methodology is illustrated in Section 3. Results and discussion in section 4 and finally this paper ends with a conclusion.

2. Study overview and Experimental setup

2.1 Overview and Objective.

The Research Institute for Solar Energy and New Energies (IRESEN) was established in 2011 to lead and promote Moroccan R&D in the field of renewable energy through applied R&D projects. To expand this leadership further, a research platform called "Green Energy Park" (GEP) was set-up in Benguerir by IRESEN and the OCP Group in 2014. The goal of this technical platform is to provide the means to characterize various PV technologies in local weather conditions and investigate their long-term behavior. The GEP's PV plant is facing the same limitations in regards to the export of electricity to the grid. Thus, the aim of this work is to provide a precise model for a short-term solar irradiance forecasting for optimal operation and a prospect grid integration in Morocco.

2.1 Experimental solar irradiance data

In the present study, we have selected some weather related variables that are directly or indirectly influencing global horizontal irradiance (GHI). GEP PV Plant is equipped with a Meteo-Station measuring accuratly various weather related variables like wind speed, temprature, pressure, ...etc. Also, a very accurate SOLYS2 solar monitoring station and it is composed of two pyranometers, one directly exposed to the sun and measures Global Horizontal Irradiance, one with a shading ball to help measure DHI the diffuse Diffuse Horizontal Irradiance and finally, one pyrheliometer measuring Direct Normal irradiance (DNI). When solar data are logged, they are sent via a Modbus data logger to GEP Servers. These data is then treated and organized in a SQL database. Data quality is checked and calibrated according to World Meteorological Organization (WMO).

For this study GHI data were aggregated with time, temprature and relative humidity. As an example Fig.1 present data of three days, 21th, 22th, 23th of april 2016 with 10 minutes lag.

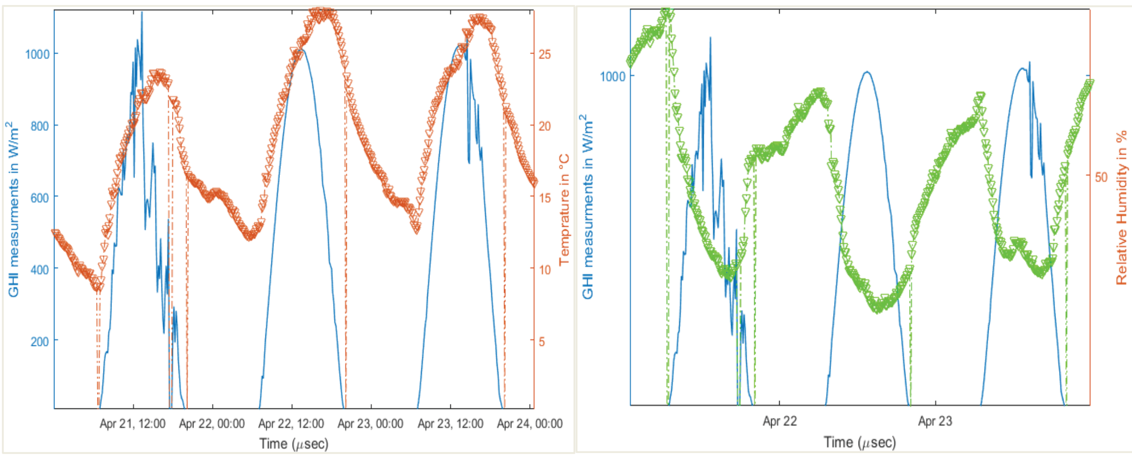

Fig. 1: Variability comparison of Temperature (Left) in red and relative humidity (Right) in green with GHI in blue.

Several datasets were created in order to investigate the influence of lags on the computation time, forecast error and finaly the choice of forecasting method.

3. Approach and implementation

3.1 Forecasting approach

The methodology undertaken was to choose two of the extreme scenarios (Clear-Sky and cloudy) of data and test them using ARIMA and time series analysis with different lags (1 min, 10 min, 1 hour) to assess the quality of forecast and for different forecast horizons. The developed ARIMA model was tuned up to see its capabilities for 48h forecast horizon.

The next step was to determine a suitable model from the different methods that deal with local sensing data, in order to do so, a MATLAB program was developed to benchmark 2 stochastic methods and one of the soft computing based on neural networks (Fig. 2).

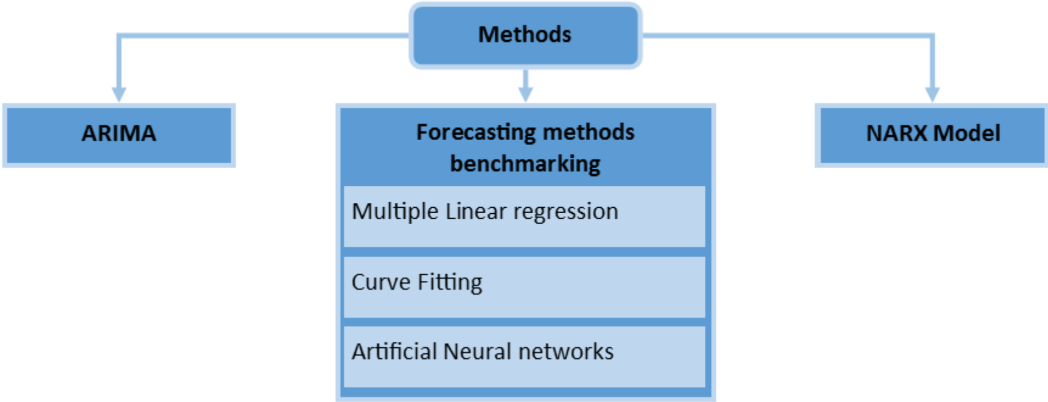

Fig. 2: Forecasting methods involved in this study.

3.2 ARIMA

Unlike the stationary model ARMA, ARIMA is a non-stationary regressive model. Non-linear methods would enable powerful structures with the ability to accurately describe complex non-linear behavior problems (Suykens JAK, 1996). This model has 3 orders, and it is known as ARIMA (p,I,q), AutoRegressive (AR), Moving average (MA) and I for the integrative part and it means that the model take into account the difference between response variable data, which is suitable for non-stationary values. This makes the model look like the following:

$$Y_t = (1 - L)^d X_t \qquad \text{(eq. 1)}$$

With Y_t being forecasted values, L the lag, d seasonality exponent and X_t actual values. In a wide sense it can be written as such:

$$\left(1 - \sum_{i=1}^{p} \emptyset_i L^i\right) Y_t = \left(1 + \sum_{i=1}^{q} \beta_i L^i\right) \varepsilon_t \qquad \text{(eq. 2)}$$

Where β_j are Moving average ***MA*** coefficients and $\emptyset_i$ are AutoRegression ***AR*** coefficients. This equation creates a multitude of forecasting models, determining the orders of ARIMA was carried out by computing ACF and PACF functions provided by *autocorr* and *parcorr* in MATLAB, also this model was tuned up using an optimization for parameters like, ***SAR*** (Seasonal AutoRegression Coefficents), ***SMA*** (Seasonal MovingAverage coefficents) and Variance. Using *optimset* function in Matlab help estimate these coefficients by incorporating a nonlinear iterative optimization algorithm called Sequential quadratic programming (SQP).

3.3 Forecasting models benchmarking

For the sake of comparison a program had been developed to select the most suitable approach for our data‘s behavior. The program compares two well-known statistical methods against one based on Artificial Neural Network.

3.3.1. Multiple Linear regression (MLR) Model

In order to be able to predict accurately, and to fit data in the best way possible, regression models are often constructed based certain conditions that must be verified. The basic linear model is a simple linear regression between dependent and independent variables. Linear regression model fits the data to a model of the following form:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \qquad \text{(eq. 3)}$$

Where i is the number of measurements, (β_0, β_1) are coefficients and ε is a random error.

The term ε is a catch-all for differences between predicted and observed values of y. These differences are due to process fluctuations (changes in β), measurement errors (changes in x) and model misspecifications (for example, nonlinear relationships between x and y). It is usually assumed that ε is generated by an unobservable innovations process with stationary covariance $\Omega_t = Cov(\varepsilon_1, \dots \varepsilon_t)$.

If the model has more than one independent variable, it will become a multiple linear regression model and will be in this form:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_k x_i^k + \varepsilon_i \qquad \text{(eq. 4)}$$

Where β_i ($i = 1,2,3, \dots, k$) are the coefficients of the model, this method is used to analyze the effect of more than one independent variable ($x_1, x_2, x_{3,} \dots, x_k$) on the dependent variable (Solar Radiation). For a given dataset ($y, x_1, x_2, x_{3,} \dots, x_k$) the multiple linear regression fits the dataset to the model according to (eq.4).

Assuming that ε are independent and identically distributed as normal random variables with $\overline{\varepsilon_i} = 0$ and the unknown variance of these random errors $\sigma^2 = Var(\varepsilon_i)$. In Order to minimize the $\|\varepsilon_i\|$ with respect to β_i, we solve the function:

$$\frac{\delta\, \varepsilon_i^T \varepsilon_i}{\delta \beta_i} = 0 \qquad \text{(eq. 5)}$$

Where ε_i^T is the transposed errors vector.

Which leads to the optimal coefficients $\hat{\beta}$ that are being estimated from the vector β by least squares method:

$$\begin{bmatrix} \widehat{\beta_0} \\ \vdots \\ \widehat{\beta_k} \end{bmatrix} = (X^T X)^{-1} X^T \qquad \text{(eq. 6)}$$

Matlab provides script driven tools to carry out multiple linear regression, the regression used returns a p-by-2 matrix of 95% confidence intervals for the coefficient estimates. The inputs were designed to bring about seasonality features in solar irradiance and to make sure that our forecasts revolve around the average of each hour.

3.3.2. Curve Fitting Model

Curve fitting finds an appropriate mathematical model that expresses the relationship between dependent and the independent variables and the parameters are estimated using nonlinear regression.

Using the historical data to find coefficients of an equation is the basic principle of Curve fitting; the equations could be multiple sin waves, Fourier or polynomial equations, etc. Matlab Curve fitting GUI is a good way to find the potential of this Model. In our case, our equation is fourier8 of the form (eq. 7):

$$Fit_{model}(x) = a_0 + a_1\cos(x*w) + b_1\sin(x*w) + a_2\cos(2*x*w) + b_2\sin(2*x*w) + a_3\cos(3*x*w) + b_3\sin(3*x*w) + a_4\cos(4*x*w) + b_4\sin(4*x*w) + a_5\cos(5*x*w) + b_5\sin(5*x*w) + a_6\cos(6*x*w) + b_6\sin(6*x*w) + a_7\cos(7*x*w) + b_7\sin(7*x*w) + a_8\cos(8*x*w) + b_8\sin(8*x*w) \qquad \text{(eq. 7)}$$

This model uses the same inputs in MLR model and all coefficients a_n, b_n and c_n will be found using “fit” function in Matlab.

3.3.3. Artificial neural network (ANN)

Artificial intelligence techniques use a symbolic approach to intelligent systems, it is a processing architecture based on the human brain focusing on information representation by its ability to learn and adapt. ANNs are constituted by a mathematical model of biological neuron called perceptron arranged in nodes and connected by weigh vectors or simply called weights (Fig.3).

ANNs can model any actual data variations by constantly changing the weights between the nodes based on information flow through the network during the learning phase.

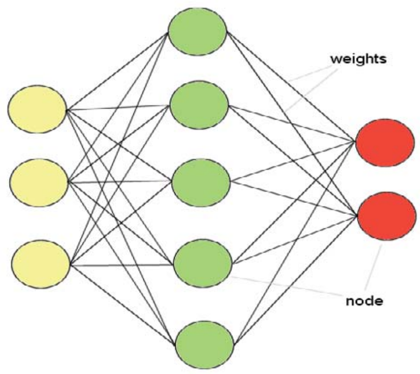

Fig. 3 : ANNs incorporate the two components of biological neural nets: Neurones (nodes) and Synapses (weights), inputs layer is in yellow, hidden layer in green and output nodes are in red.

ANN is well suited for modeling complex relationships between inputs and outputs with an ability to construct a map without explicit analytical equation, which is at the same time a very powerful tool to model nonlinear statistical data. The basic mathematical model of ANNs is shown below.

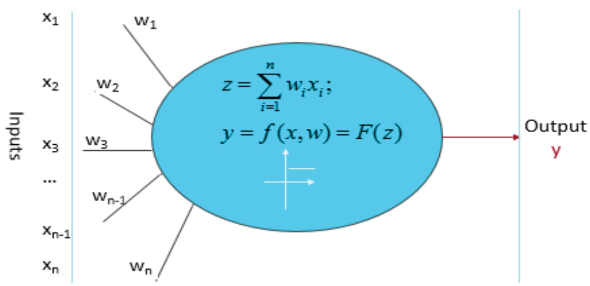

Fig. 4 : Neurons work by processing information as they receive and provide it in form of spikes.

(Fig.4) present the McCullogh-Pitts model, where x_i is an input vector, w_i are weights values, $f(x, w)$ an activation function and y is the output. The architecture used in this study is a feedforward neural network and is designed in a way to acquire a forecasts error less or equal to the error imposed by the user via a constraint imposed in a loop, so the program can reiterates until it finds the forecast with the minimum error. The training algorithm used is Levenberg-Marquardt, Training, Test and Validation dataset are divide randomly from the original data, the best results were acquired for 5 hidden neurons with a logistic sigmoid activation function and one output neurons with an adaptive linear function.

3.4 NARX model implementation

The Nonlinear AutoRegressive with eXogenoues inputs model or NARX in short, is a specific type of recurrent neural architecture (RNN), it was derived from derived from Autoregressive exogenous (ARX) model commonly used in time-series modeling. It is a recurrent dynamic network, with feedback connections enclosing several layers of the network. It has limited feedback architectures that come only from the output neuron instead of from hidden neurons.

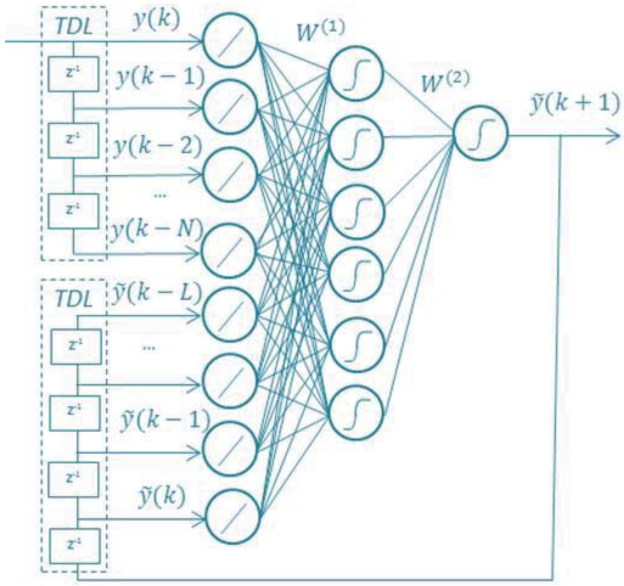

Fig. 5: NARX NEURAL NETWORK (Mladenov, 2013)

The NARX neural network structure is shown in Fig. 5 is equipped with both a tapped delay line (TDL) at the input y and global recurrent feedback connections $\tilde{y}$. It represents a Dynamic Multilayer Perceptron. The input vector can be written as such:

$$x(k) = [y(k) \dots y(k-N) \quad \tilde{y}(k) \dots \quad \tilde{y}(k-L)]^T \qquad \text{(eq. 8)}$$

Where N is the order of the input tapped delay and L is the order of the feedback tapped delay line. Long-term dependencies that solar radiation has with other exogenous input was the reason for choosing an ANN that uses TDL.

The output is then generalized by:

$$\tilde{y}(k+1) = g(\textstyle\sum_j w_j{}^{(2)}(f(\sum_i w_i{}^{(1)} x_i))) \qquad \text{(eq. 9)}$$

Where $w^{(1)}$ and $w^{(2)}$ are weights of the hidden and output layers, $f(.)$ *and* $g(.)$ are activation functions respectively of the hidden and the output layers.

NARX use taps to set-up the delays across the inputs. Also, it incorporates the past values of the output. Recurrent networks have loops within intermediate layers and incorporate memory via these loops.

The inputs for the NARX ANN are GHI time serie and as exogenous inputs we have chosen what might reflect the changes in GHI, variables like *relative humidity* and *ambient temperature* were often mentioned in literature as exegenous inputs for NARX ANN and as the fig.2 shows they follow to a certain extent solar irradiance‘s seasonality.

Also, we have divided our dataset to inputs and targets, target series are what we compare the response of the neural network to. So the first group is used to train the Neural Network while the second group of variables are Validation series, this new data is used for simulation. For this model we have used a 10 min resolution time-series from 1st of January 2016 to 1st of February 2016.

For learning process, ANNs training uses mathematical procedures to adjust the network's weights and biases. The training function necessitates a global algorithm that affects all the weights and biases of our network. The algorithm used is *Levenberg-Marquardt algorithm* found in NN toolbox in Matlab. And then we have divided our dataset into 3 parts, Training, validation and test datasets by setting up ratios respectively 60%, 20%, 20% of input vectors. The training performance function is Mean squared error (MSE), minimizing this error while our network learns the dataset behavior. Also, we assigned *Tansig* activation function for hidden layers and *Purelin* for the output layer. Number of the hidden layers is determined by trial and error until the program converges at an error less or equal to the desired error entred by the user. The evaluation of each trial is carried out by Mean Absolute Percent Error (MAPE).

4. Results and discussion

4.1 Forecasting evaluation metrics:

A standardized performance evaluation was adopted to asset the accuracy of each model involved in this study, the metrics used are RMSE and MSE because they give more weight to the largest error and MAE was used to give the average of the magnitude of forecast error. Also we have used MAPE as a constraint over forecast iterations in our programs.

We define the forecast error: $e_i = y_{i,forecasted} - y_{i,observed}$ (eq. 10)

where $y_{i,forecasted}$ is the ith forecasted value, $y_{i,observed}$ the ith actual value and N is the length of the test dataset.

$$RMSE = (MSE)^{\frac{1}{2}} = \left(\frac{1}{N}\sum_{i=1}^{N} e_i^2\right)^{\frac{1}{2}} \quad \text{(eq. 11)}$$

$$MAE = \left(\frac{1}{N}\sum_{i=1}^{N}|e_i|\right) \quad \text{(eq. 12)}$$

$$MAPE = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{|e_i|}{y_{i,observed}}\right) * 100 \quad \text{(eq. 13)}$$

4.2 ARIMA

The first set of experiments was carried out for intra-day forecasts with the horizon of 30 minutes. The two datasets used for clear-sky and cloudy day have a 1 min resolution. Running ARIMA gave us an error less than 5% MAPE for clear-sky and 10% cloudy day after the beginnig of each day. The dataset used was only one day of GHI. However, it seems very problematic if the program is supplied with more than two days of data. Taking into account the computing power of the processor (4 CPU 1.6 GHz, 8G RAM), running time average in this case around 15 to 30 seconds. If the dataset is too large, computation takes a lot more time to produce results. So we have concluded that it is best to reason with the length of the dataset, not with its duration.

For a Days-ahead forecast, we have begun the procedure by estimating the model through ACF and PACF, calculating residuals, testing residuals, running simulation and finally comparing results for the three data sets chosen. A constraint in the program has been integrated in order to stop it from running if the calculation was going to take a lot of time, this is the reason behind the poor results acquired for larger datasets.

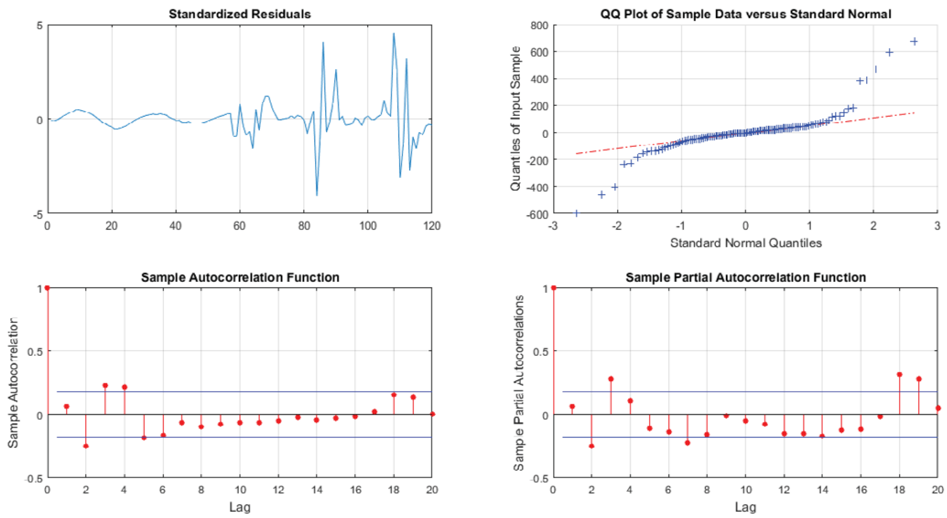

Fig. 6: Tuned up ARIMA Standarized Residuals and QQplot (1 Week 1 Hour)

After optimizing the process of getting ARIMA coefficient, we have noticed a reduction in standardized residuals amplitude and the evenly distributed quantiles on the references line (Fig. 6 Standarized residuals plot). However, there are fractions that indicate that our forecasted values is still greatly different from real measurements (Fig.6 QQ plot).

ARIMA results were evaluated and are presented in (Tab. 1):

Tab. 1: Tuned-up ARIMA Model results for the 3 chosen datasets

	1 Week 1 Hour	1 Week 10 minutes	1 Month 1 Hour
MAE W/m²	98,78	159,67	335,769
MSE W/m²	2,47E+04	7,85E+04	2,05E+05
RMSE W/m²	157,19	280,14	453,64

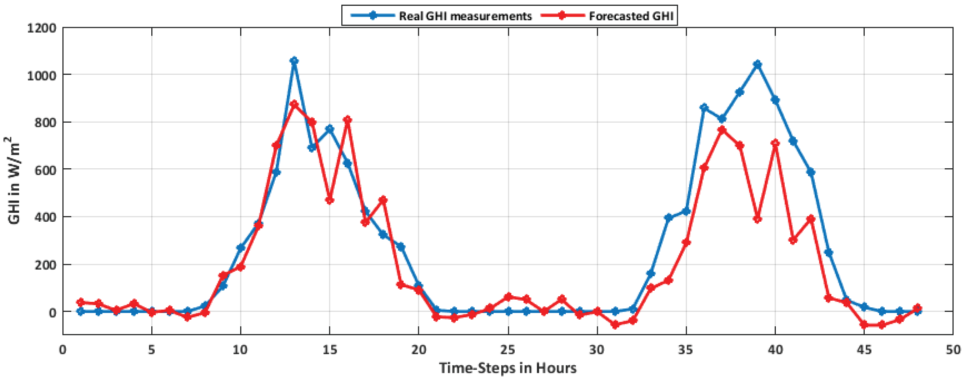

Fig. 7: Forecasted GHI results using 1 Week of data with 1 Hour resolution

The best results were recorded for one week of solar data with one hour resolution presented in (Fig. 7), the larger the dataset is the more time the computer takes to produce forecasts. A constraint over computational time was however put in place to have some rapidity while maintaining the same reliability. This constraint had largely increased the error on one Month dataset with the resolution of one hour. But it still gives relatively acceptable results if compared to the other datasets. In either ways, ARIMA seems incapable of producing reliable forecast for long periods. One limitation of ARIMA models is that they often overlook the physical behavior of time series objects, in our case sunrise and sunset. A study by (Moreno-Munoz, 2008) showed that ARIMA can accurately forecast hourly irradiance levels using data from a few previous hours. However, due to the discontinuity of solar irradiance, forecasts immediately adjacent to sunrise and sunset are problematic and increase forecast errors.

4.3 Benchmarking Program Results and Analysis

For this task we have used a year of historical GHI data with the resolution of 1 hour, the results are compared for clear-sky and cloudy conditions as presented in Fig.8:

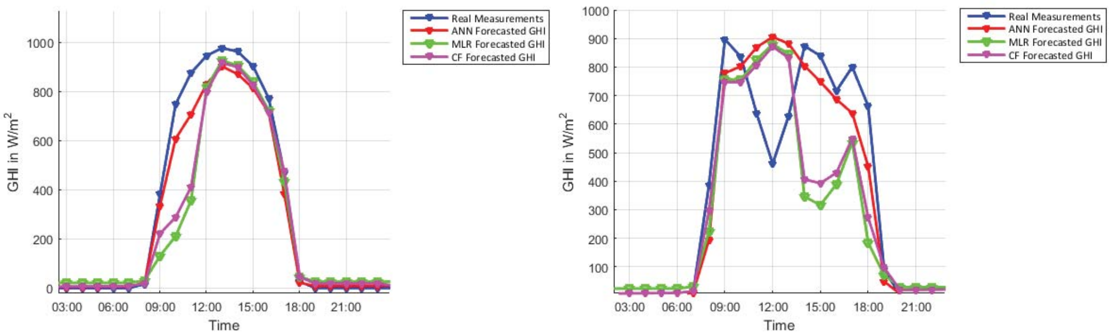

Fig. 8: Forecasting Benchmark results of a clear-sky day and cloudy day

Using the daily and hourly averages of solar irradiance of the previous day imposes that our MLR model to mimic yesterday's behavior, the choice of this procedure was taken because generally adjacent days follow relatively the same behavior. MLR model gave 56.352% MAPE for Validation dataset, this Error was reduce to 22.733% Using Curve Fitting model. ANN model is seeeminly independent of adjecent days but gives relatively satisfying results of 16, 264% MAPE.

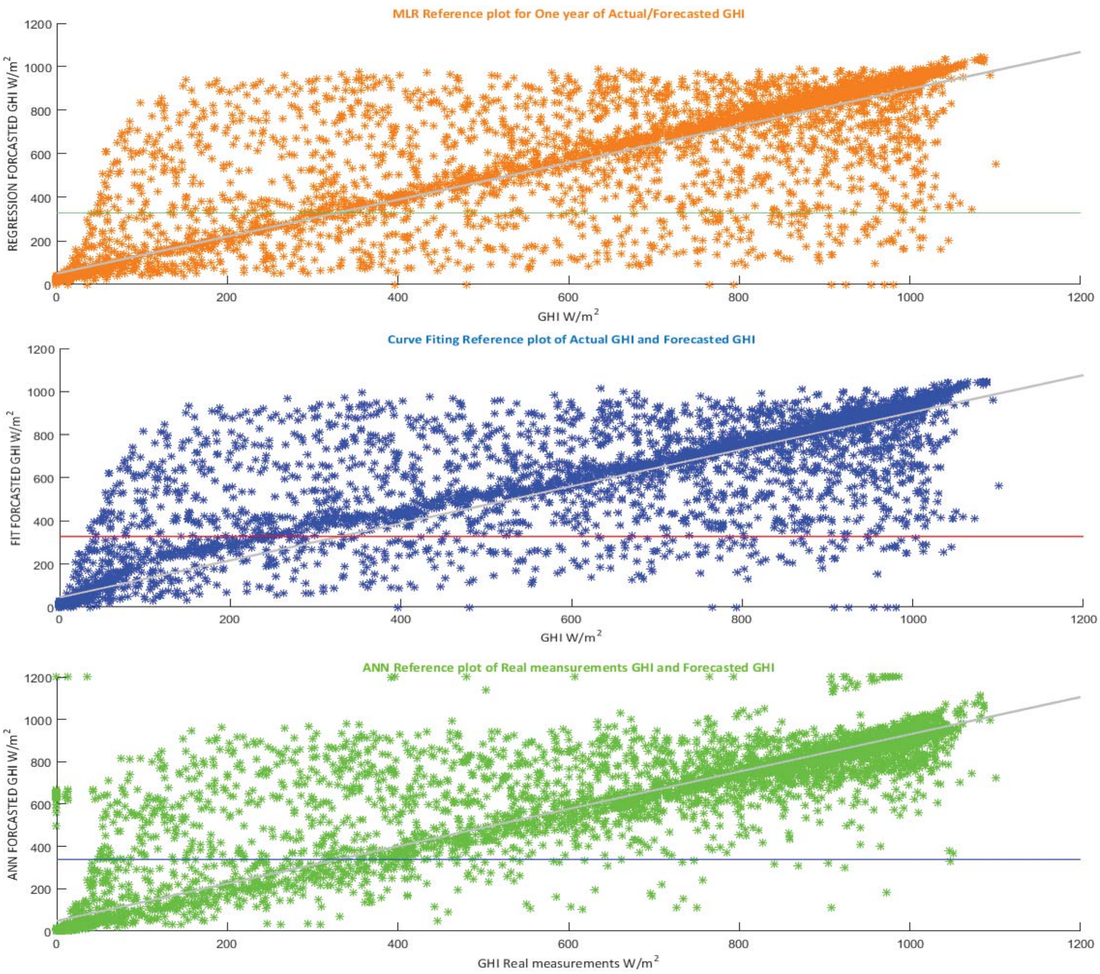

Fig. 9: Reference plot of the three benchmarked methods using 1 year of GHI with 1 hour resolution

For visual comparison we present in (Fig. 9) reference plot for MLR model in orange, Curve Fitting in blue and ANN model in green, horizontal line represent mean value, and the inclined line represent least-squares line.

If the diffenrence between forecasted and measured values was smaller the dots tend to be close to the least-square line, so we can notice that ANN response fits GHI historical data better than MLR and CF models, which is verified by (Tab.2) of Metrics presented in section 4.1:

Tab. 2: Benchmarking forecasting models evaluation using 1 year of GHI with 1 hour resolution

	Curve Fitting		Multiple Linear Regression		ANN	
	Validation Error	Historical Error	Validation Error	Historical Error	Validation Error	Historical Error
MAE W/m^2	23,35	81,1673	29,94	75,57	20,77	66,87
MSE W/m^2	1,05E+03	2,42E+04	1,48E+03	2,21E+04	0,77E+03	3,38E+04
RMSE W/m^2	32,58	155,57	38,3	148,513	22,1	183,729
MAPE	22.733%	44.644%	56.352%	123.045%	16, 264%	331,11%

While regression barely gave admissible results, improving it with fitting curve has considerably enhanced it. But between the three combined models, the best is Artificial Neural Networks model as the Validation MAPE error indicates, even with the high error of historical GHI due the learning phase and night time values. Ergo tuning up Neural Networks model will surely produce a robust forecasting engine.

4.4 NARX ANN model

Mean squared Error MSE gives a huge proportion to the difference between predicted and real values, using it in as performance function in learning phase helped reduce the error drastically. However evaluating regression models is often done by MAE or MAPE as they only evaluate the difference between two data values. The simulation were carried out and the Neural Network stopped training with ten layers at a delay of 2, in (Fig. 10) the NARX ANN is presented.

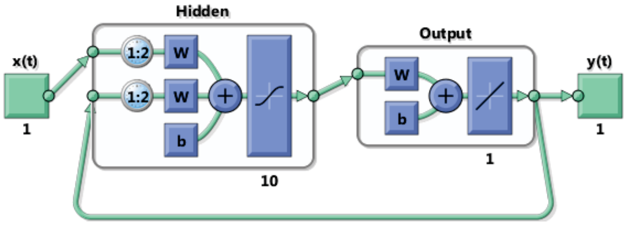

Fig. 10: NARX ANN Model architecture.

x(t) is a matrix containing historical data lagged by 10 minutes from 15th of April to 15th of May of GHI, Relative humidity and ambient Temperature and as an output y(t) is forecasted GHI. The choice of the input matrix was taken for examining various meteorological variables and how they change with GHI (Fig. 2).

We have tried this model for three Forecast Horizons, one day, two days and three days ahead results are shown in (Fig. 11) below:

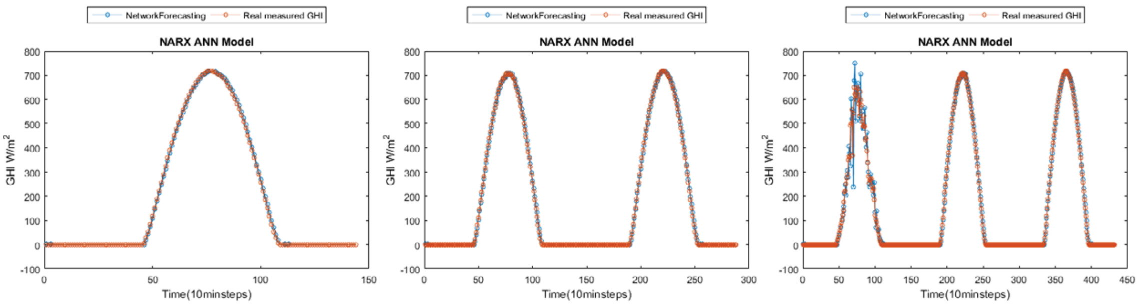

Fig. 11: Plot of forecasted values and real measurements of NARX model for the three Forecast Horizons

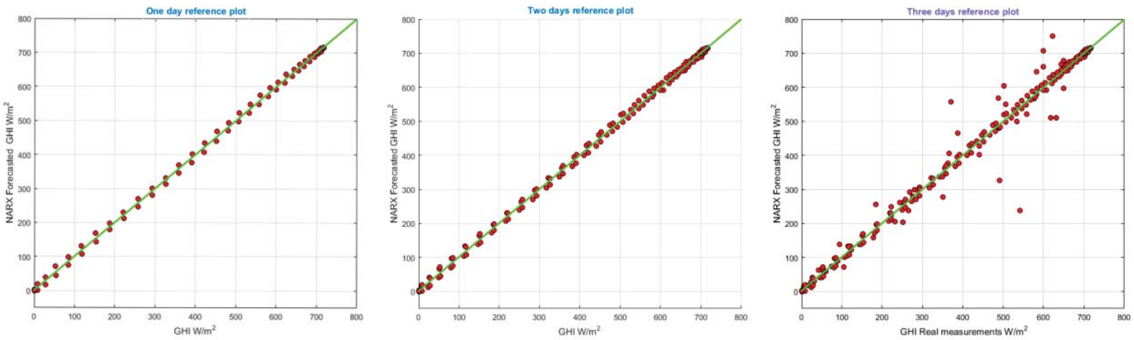

Fig. 12: Reference Plot of NARX model for the three Forecast Horizons

For a clear sky conditions, the results are near perfect as shown in (Fig.12) for one and two days forecast horizon, with less than 4,57 % MAPE error, however for cloudy day the minimum error acquired was 14,67% which by far less than the other models previously developed in this study. Also, as presented in (Tab. 3) we noticed that when the Forecast Horizons gets bigger the accuracy decreases, it was the same for all models, however for NARX model this lost of accuracy is really subtle and relative to cloudy days presence, but for clear sky the forecast is very good which demonstrates the robustness for NARX model.

Tab. 3: Evaluation metrics results for NARX model

	1 Days ahead	**2 Days ahead**	**3 Days ahead**
MAE W/m^2	8,82	9,3126	4,21E+01
MSE W/m^2	1,47E+02	1,64E+02	1,71E+04
RMSE W/m^2	12,10	12,81	130,95

The results in Tab. 3 shows that NARX ANN model outperforms all the other models, and the best way to make a neural network learn seasonality and the trend is to make a construct of inputs that are related to each other to a certain extent, temperature and relative humidity gave better results if compared with clearness index.

For Clear-sky day the forecasted values seem to be close to real measurements as they follow the least squared error line, as for cloudy days the majority of values are near real measurements, the other minority have bigger differences from real measurements.

5. Conclusion

In this paper, a methodology has been undertaken for short-term globale horizontal irradiance forecasting for the case of Green Energy Park of benguerir. An ARIMA model was developed and showed its efficiency in Intra-hour forecasts and its inability to produce reliable forecasts for longer horizons. Other stochastic methods were also developed for the sake of comparison. Multiple linear regression (MLR) and curve fitting were compared with artificial neural networks. We also have demonstrated that MLR barely has the ability to determine the general trend of used solar data but it cannot model data non-linearities, MLR was tuned with curve fitting by trying to find coefficients for the Fourier polynomial equation, the improvement seen were barely noticeable. However, for ANNs, the results were promising and it was a better choice for further developments. For this reason, a new forecasting model was developed based on NARX ANN, the results were very satisfying as the forecast error (MAPE) was decreased to 4,57 % for Clear-sky day and 14,67% cloudy day.

To make a good use of enhancements made in GEP's forecasting model, using SANDIA PV library in Matlab, we have succeeded in modeling 12 photovoltaic systems with different photovoltaic technologies from different PV modules manufacturers. This precise model will be used as conversion process for solar irradiance forecasts to acquire DC power output predictions of GEP's PV power Plant.

6. Future work

In order to make the model developed in this study more reliable, a hybrid model based on satellite images and NWP is under development. As for the GEP PV power plant model, some enhancement are underway to make the model more responsive to real weather conditions.

7. References

A. Di Piazza, M. D. (2013). Solar radiation estimate and forecasting by neural networks-based approach. *XIII Spanish Portuguese Conference on Electrical Engineering*.

Aguiar RJ, C.-P. M. (1989). The modeling of daily sequences of hourly radiation through autoregressive methods. *Proceedings of the congress of the international solar energy society*, vol. 3. p. 2107e11.

Al-Alawi S, A.-H. H. (1998). An ANN-based approach for predicting global radiation in locations with no direct measurement instrumentation. *Renewable Energy* .

Chi Wai Chowa, B. U. (2011). Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. *Solar Energy - Elsevier*, 2881–2893.

Diagne, M. (2013). Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews Volume 27,* , 65–76.

Jäger-Waldau, A. (2015). *PV Status Report.* Publications Office of the European Union, European Commission. Joint Research Centre.

Jain, M. B. (2012). Curve fitting and Regression line method based seasonal short term load forecasting. *IEEE*.

Koca, A. O. (2011). Estimation of solar radiation using artificial neural networks with different input parameters for Mediterranean region of Anatolia in Turkey. *Expert Systems with Applications -Volume 38, Issue 7*, 8756–8762.

Lima, F. J. (2016). Forecast for surface solar irradiance at the Brazilian Northeastern. *Renewable Energy 87*, 807-818.

Lopez, G. B.-P. (2005). Selection of input parameters to model direct solar irradiance by using artificial neural networks. *Energy 30*, 1675–168.

Mladenov, V. (2013). Artificial Neural Networks and Machine Learning-. *Springer proceedings of the 23rd international conference of AI and machine learning*.

Moreno-Munoz. (2008). Short term forecasting of solar radiation. *IEEE International Symposium on Industrial Electronics*, pp. 1537–1541.

Rich H. Inman, H. T. (2013). Solar forecasting methods for renewable energy integration. *Progress in Energy and Combustion Science*, 535-576.

Sahin, M. (2012). Comparison of ANN and MLR models for estimating solar radiation in Turkey using NOAA/AVHRR data . *Elsevier - Advances in space research turkey*.

Suykens JAK, V. J. (1996). Artificial neural networks for modeling and control of non-linear systems. *Kluwer Academic*.

Vaz, A. (2015). An artificial neural network to assess the impact of neighbouring photovoltaic systems in power forecasting in Utrecht, the Netherlands. *Renewable energy 85 Elsevier*, 631-641.

Zhang, G. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing 50* , 159 - 175.