Solar Irradiance Forecast Using Artificial Intelligence Techniques

José Diogo Marques do Rego*, Rui Manuel Gameiro de Castro*
* Instituto Superior Técnico
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
jose.rego@tecnico.ulisboa.pt, rcastro@tecnico.ulisboa.pt

***Abstract*—The importance of renewable energies has been growing at a fast pace, both because of the need to solve problems related to environmental issues and as a way of helping the increasingly difficult management of electricity grids. The techniques of Artificial Intelligence have already shown their effectiveness in tasks of high complexity, namely, Regression, Classification, Forecasting. Also in the field of Renewable energies these tools can be extremely useful, in particular in the prediction of Solar Irradiance. In this work, we developed two algorithms in Matlab^{circledR} of prediction of Solar Irradiance based on two methods of Artificial Intelligence, which are the Artificial Neural Networks and the K-Nearest Neighbors Method. In the forecasting process, the models are trained with subsets of the one-year Solar Irradiance register in the city of Lisbon and then the next hour's forecast is carried out. In order to understand the best method to perform predictions of solar irradiance among those studied, a comparative study between the models was carried out, taking into consideration the prediction errors and the simulation times of both models in the simulations made in different situations.**

***Index Terms*—ANN, KNN, solar power, forecasting, renewable energy, machine learning.**

I. Introduction

Renewable Energy is becoming a technology and an increasingly viable alternative to the conventional Non-Renewable Sources of Energy. The constant threat of the Climate Changes thus requires the mankind to look for better and more efficient ways of producing electricity. Specially when all our basic needs are based on this type of energy. The total amount of Solar Energy delivered by the Sun to the Earth is unimaginable. For example, the San Francisco earthquake of 1906 reached a 7.8 Richter's magnitude equal to $10^{17}J$ of released energy, which is the same amount of energy delivered by the Sun per second [10]. For these reasons is natural that Solar Power, due to is great potential, starts to showing some signs of increasing its use and it is one in which it has been invested most in order to improve its efficiency. Unfortunately, his dependence on external meteorologic conditions turns this technology somewhat volatile and sometimes unattractive both at the energy and at the economic level.

Currently the use of Artificial Intelligence (AI) Tools, like the Artificial Neural Networks(ANN) or the K-Nearest Neighbors(KNN), is becoming more and more common due to their capacity of solving highly complex problems. The improvement in the computers and in the algorithms performance also helped at solving problems, not only in engineering, but also in many areas like medicine, finances and literature. In the case of Renewable Energy Sources these artificial intelligence tools are used to perform forecasts of the energy produced in a Power Station or even forecasts of the behavior of weather conditions. An accurate forecast of this variable can promote a better planning and operation of Power Distribution at economic level or energy production level, reduce the impact of PV output uncertainty on the grid, improve the system reliability, maintain the Power Quality and increase the penetration level of PV Systems.

Several methods of Solar Irradiance Forecasting have been developed over the years, whether applying existing methods [1] [2] or combining more than one method [3]. The ANN [4] [5] [6] and the KNN [7] [8] [9] are two of the most well-known methods of Forecasting and for that reason a comparative method is necessary to understand which is the best to forecast Solar Irradiance. This paper presents two Solar Irradiance Forecast models using Matlab^{circledR} Language and based on the ANN and KNN methods as well as the comparative study of the performance of both models.

II. Forecasting Models

A. Persistence Model

The Persistence Model is the simplest forecasting model and is usually used as baseline or comparison for other models. During the study of new forecasting models, the performance of these models will be acceptable if they perform better than the Persistence Model. This Model calculates future time series values with the premise that all influential condition does not change between time t and time $t + \Delta t$. For this work, this premise is applied to the evolution of irradiance and the model considers that the irradiance for time $t + \Delta t$, or the next irradiance value, is equal to irradiance for time t, or the actual irradiance value (example in Figure 1). Mathematically this can be represented by the equation:

$$I(t + \Delta t) = I(t) \tag{1}$$

where $I(t+\Delta t)$ is the irradiance for future time (where Δt can be any time interval) and $I(t)$ the current value of irradiance.

B. Artificial Neural Network (ANN)

The Artificial Neural Networks (ANN) Model is an AI method based on the human capacity of learning and adapt

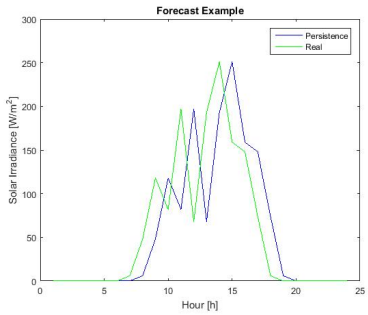

Fig. 1. Solar Forecast example to explain the Persistence Model Forecast.

his way of thinking through his obtained life time experience [11]. This method is capable of compute nonlinear modeling without knowing in the beginning the relation between input and output variables thanks to his nonlinear data-driven structures. The Artificial Neurons are composed by a Sum Block connected to a Transfer Function. The inputs of the Sum Block are the Input Vector multiplied by the correspondent Weight Vector and the Bias. After the sum the result, called Net Input, goes into a Transfer Function, and the result produced by it is called the Neuron Output (Figure 2):

$$a^{k+1}(i) = f^{k+1}(n^{k+1}(i)), \quad k = 0, 1, 2, ..., M-1 \tag{2}$$

where f^{k+1} is the Transfer Function and $n^{k+1}(i)$ is the Net Input for Neuron i. The Transfer Function depends on the specification of the problem that the neuron is trying to resolve and can be Linear or Non-Linear. Like a brain or a biological

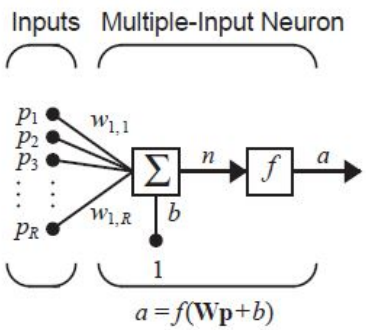

Fig. 2. Artificial Neuron.

neural network, an ANN is composed by a lot of Neurons and the way that these Neurons are connected between them defines the type of architecture we are using. There are different network architectures and Learning Processes that we can use. In this work the choices fell on a Feedforward Network for the architecture and for the Learning Process the Levenberg-Marquardt Backpropagation Algorithm.

The Feedforward Network is a Multilayer Neural Network composed by two layers, one Hidden Layer and one Output Layer. The transfer functions used in this network are Tan-sigmoid function in the Hidden Layer and the linear function in the Output Layer.

The training algorithm used in this ANN is the Levenberg-Marquardt Algorithm. This algorithm is an variation on Backpropagation Learning Algorithm which is an Supervised Learning Algorithm [12]. The first step of the Levenberg-Marquardt algorithm is propagate the input result to the end of the network and compute the sum of squares function:

$$V(\underline{x}) = \sum_{i=1}^{N} e_i^2(\underline{x}) = e^T(\underline{x})e(\underline{x}) \tag{3}$$

where $e_{\underline{x}}$ is the error associated to the vector:

$$\begin{aligned} \underline{x} = [w^1(1,1) \quad & w^1(1,2)...w^1(S_1,R) \quad b^1(1)... \\ & b^1(S_1) \quad w^2(1,1)...b^M(S_M)]^T \end{aligned} \tag{4}$$

$$e_q = t_q - a_q^M \tag{5}$$

and t_q and a_q are respectively the vector of the desired targets and the vector of the output resulting from Equation 2. After this step, the algorithm computes the Jacobian Matrix, which is composed by derivatives of the errors:

$$J(\underline{x}) = \begin{bmatrix} \frac{\partial e_1(\underline{x})}{\partial x_1} & \frac{\partial e_1(\underline{x})}{\partial x_2} & \cdots & \frac{\partial e_1(\underline{x})}{\partial x_n} \\ \frac{\partial e_2(\underline{x})}{\partial x_1} & \frac{\partial e_2(\underline{x})}{\partial x_2} & \cdots & \frac{\partial e_2(\underline{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(\underline{x})}{\partial x_1} & \frac{\partial e_N(\underline{x})}{\partial x_2} & \cdots & \frac{\partial e_N(\underline{x})}{\partial x_n} \end{bmatrix} \tag{6}$$

The last step is the update of the $\underline{x}$ vector through the equation:

$$\Delta\underline{x} = [J^T(\underline{x})J(\underline{x}) + \mu I]^{-1} J^T(\underline{x})e(\underline{x}) \tag{7}$$

where μ and β are factors used to control the learning rate. These values are equal to 0.001 for μ and 10 for β. In the end of these steps is time to compare the sum of square errors $x + \Delta\underline{x}$ with $V(\underline{x})$. If the new sum is bigger than the sum computed in the beginning of the iteration, then μ is multiplied by β and we repeat the computation of $\Delta\underline{x}$. If the new sum is smaller, then μ is divided by β and the process returns to the beginning of the algorithm.

This Learning Process is repeated until the gradient $\nabla V(\underline{x})$ is smaller than some predetermined value, the sum of square errors reduced to a predetermined value, the algorithm reaches the number maximum of iterations or the algorithm reaches the maximum of validation checks obtained by cross validation of the Early Stopping Method. The gradient can be calculated by:

$$\nabla V(\underline{x}) = J^T(\underline{x})\underline{e}(\underline{x}) \tag{8}$$

C. K-Nearest Neighbors (KNN)

The K-Nearest Neighbor (KNN) Model is, like the ANN method, an Artificial Intelligence method, being also considered a System of Pattern Recognition [13] [14]. This type of systems are able to identify a random object knowing previously the nature of one set of similar objects.

This method started to be used in Classification problems. The main objective of these problems is to assign a classification to a chosen point based on the class of the training set

points. This process can be compared to the human sensory capacity like recognizing a face, identify our accessories in our bag by feeling their shape, understanding different words in different languages by listening or decide if an apple is ripe only by its smell.

The KNN method is an application of the Maximum a Posteriori Probability (MAP) Classification with the Probability Density Functions obtained by the Parzen Method with Adaptive Window. In short, the KNN Classification Algorithm can be explained in three steps. First, we need to calculate the distances between the pattern $x \in \mathbb{R}^n$ (the point that we want to classify) and the other training patterns (Figure). This calculation is obtain through a Distance Function. The most used for continuous variables are:

$$\text{Euclidean} \quad \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{9}$$

$$\text{Manhattan} \quad \sum_{i=1}^{k}|x_i - y_i| \tag{10}$$

$$\text{Minkowski} \quad (\sum_{i=1}^{k}(|x_i - y_i|)^q)^{1/q} \tag{11}$$

After that, we select the k training patterns nearest to x. Finally, we determine the most present class among all k-Nearest Neighbors. The classification for x is the same class previously determined.

The KNN Method can also be used to solve Regression Problems. In this case, instead of classify, the main objective is to determine the numerical value of a variable of a unknown case, which means, each training pattern is not associated to a class but to a numerical value of a specific variable. To solve a problem like this, the algorithm of KNN Regression is in all similar to the one of KNN Classification except for the last steps. It starts the same way calculating the distance between all the training patterns and the point to be studied, x. After this, and after discovering the k-nearest patterns, the variable numerical value of x is equal to the average of all the variable numerical values of the k-Nearest Neighbors.

III. Case Study

A. Data

As the main objective of this thesis is to predict with the best performance the Solar Irradiance of one day, the data used for this purpose is the Solar Irradiance registered in Lisbon for 365 days, or one year. In this Case Study, beside knowing that the Solar Irradiance is, directly and indirectly, influenced by other meteorological parameters (Date and Time, Precipitation, Cloud Cover, Air Pollution, etc.), these parameters were not considered and were not used in any test or simulation during this work.

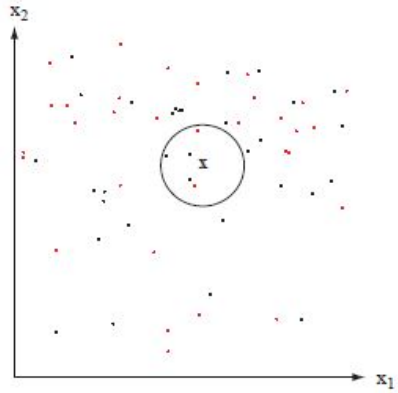

Fig. 3. Example of KNN Classification Algorithm. In this case the training patterns are arranged according to two parameters and the algorithm search for the 5 Nearest Neighbors to x.

B. ANN

To apply the ANN model to this case study we will need 3 matrices based on the Data Matrix, a Training Matrix, a Target Vector and an Input Vector. These matrices are created and filled for each hour forecast, which means that different predictions have different matrices.

The Training Matrix is the Matrix that stores all the information of the past hours Irradiance. Each column of the matrix contains the past information of a certain hour. The number of rows depends on the information we want to store about the same hour.

The Target Vector is simply the vector that stores the real value of Irradiance for one specific hour. This vector has only one row, due to the fact that it saves only one kind of information, but it has exactly the same number of columns of the Training Matrix.

Finally, the Input Vector is the vector used to predict the next value of Solar Irradiance. The dimension of this vector has to be equal to the number of rows of the Training Matrix and it includes the records right previous to the hour supposed to predict.

To get the best performance of the model it is necessary to test the main parameters (e.g. hidden layer size, number of epochs, type of input information, number of days to train the network). After these tests, the final simulations with this model are made with:

- a Storage Method that includes past records of Solar Irradiance and the hour to predict;
- 20 Hidden Layers;
- default maximum number of Training Epochs (1000 Epochs);
- 2 hour delays in the storage method;
- 1 month and a half of Solar Irradiance data to train the network.

C. KNN

Like the ANN Model, the KNN needs 3 matrices named Train Matrix, Target Vector and Input Vector. The purpose of

these matrices is the same as the purpose of the matrices in the ANN Model but the way that the information is stored is different. The Train Matrix stores the records of Solar Irradiance of different days in each row (by default was used records of 24 hours length), the Target Vector stores the Solar Irradiance record registered after the records of the Train Matrix, and the Input Vector stores the record of Irradiance of the last hours before the hour to be predicted.

The Distance Function chosen for this work was the Euclidean Distance Function. Since the version of Matlab^{circledR} used in this work does not have the tools for KNN Regression, but only for KNN Classification, the results using those tools were not quite correct. To solve this problem we created an algorithm using simpler tools to calculate the distance between points and the average of the target records.

As in ANN case, before testing the real performance of the KNN Model, the test of the main parameters of this model was required. the results of these tests revealed that the best parameters to forecast were:

- the chosen algorithm to train the model and calculate the K-Nearest Neighbors are an Ensembling Learning Approach with an Weighted Distance Function;
- the records have a length of 24 hours;
- the training matrix has 35 records.

The Ensembling Learning Approach is one solution developed in to overcome the problem of choosing the best K Parameter for each KNN Classification [15]. In this work, we made an adaptation of this solution to our forecast problem. The program predicts the value of Irradiance for a certain hour using different numbers of K Neighbors (1,2,...,$\sqrt{N}$, where N is equal to the number of training patterns). After that the predicted value is equal to the weighted average of all the predictions. The Weighting Function is an Inverted Logarithmic Function, similar to the one proposed in the article

$$w(k) = \frac{1}{\log_2(1+k)} \tag{12}$$

in which k is the number of the nearest neighbors.

D. Performance Evaluation Tools

To make the analysis as rigorous as possible the tests must be carried out in the same conditions whenever possible. Each test was repeated 10 times due to the random nature of the initial weights in the ANN model. Different weights and bias will converge to different output values. In the KNN case the performance error does not change because the points selected for the forecast remain the same in different simulations. Therefore, to obtain an accurate result we made an average between the 10 results of each test repetition.

In order to obtain the performance error of each forecast and in order to analyze every parameter and case in a better way it was necessary to use some mathematical tools, which are very common in this kind of studies:

- Root Mean Square Error: $RMSE = \sqrt{\frac{\sum_{t=1}^{N}(z_t - SI_{real})^2}{N}}$
- Mean Absolute Error: $MAE = \frac{\sum_{t=1}^{N}|z_t - SI_{real}|}{N}$
- Mean Absolute Percentage Error: $MAPE = \frac{1}{N}\sum_{t=1}^{N}|\frac{z_t - SI_{real}}{SI_{real}}|$

where z_t is the value of Solar Irradiance predicted in the model, SI_{real} is the real value and N is the total number of predictions. For the night period as we know *a priori* that the forecast for this period is equal to zero then the calculation of the performance error for this period is not considered in every tests. We will considerate the error when the actual Solar Irradiance is superior to 10 W/m^2.

The principal evaluation criteria used in this work is the MAPE. Nonetheless we have used other criteria for possible indecisions between different hypothesis. In some cases these 3 types of errors were not enough to decide which was the best case. Then, we calculated the relation between the error of the forecast and the Persistence Model as you can see in the next equation.

$$MAPE_{model} = \frac{MAPE_{pred}}{MAPE_{pers}} \tag{13}$$

IV. Results

After having chosen the best model parameters to perform the best predictions possible, the next step will be to make normal day predictions to compare the performance of both methods. We will carry out two different tests to make this comparison.

For the first test we have chosen to predict four specific days of the four different seasons. For training each model, we have chosen the central months of each season (January for Winter, April for Spring, July for Summer and October for Autumn), since these months have the best characteristics of the season. The days chosen to predict were the first days of the last months of the seasons (1st of February for Winter, 1st of May for Spring, 1st of August for Summer and 1st of November for Autumn). In the particular case of the Winter, since both methods need more than 31 days to train the respective models it was not possible to forecast the Solar Irradiance for the 1st of February as we said before. Because of that, instead of forecast the intended day it we predicted the 14th of February. The results of these comparative tests can be observed in table I and in the Figures 4 to 7. The first observation we can

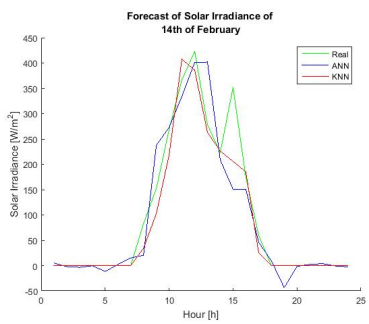

Fig. 4. Prediction of the 14th of February comparing both methods.

TABLE I
PERFORMANCE OF MODELS:SEASONS; DAILY AVERAGE ERRORS

Method	RMSE $[W/m^2]$	MAE $[W/m^2]$	MAPE	MAPE$_{model}$
Winter (14-Feb)				
ANN	74.79	55.28	0.3043	0.4791
KNN	57.72	43.52	0.2409	0.3794
Persistence	111.05	104.40	0.6351	1.0000
Spring (01-May)				
ANN	26.32	19.44	0.0804	0.1057
KNN	100.58	84.73	0.2314	0.3042
Persistence	145.35	130.36	0.7605	1.0000
Summer (01-Aug)				
ANN	33.66	25.69	0.0658	0.1427
KNN	31.88	22.99	0.0415	0.0900
Persistence	136.04	121.79	0.4610	1.0000
Autumn (01-Nov)				
ANN	91.02	74.33	0.4612	0.6426
KNN	60.39	45.43	0.2852	0.3974
Persistence	107.58	87.80	0.7177	1.0000

TABLE II
PERFORMANCE OF MODELS:SEASONS; DAILY AVERAGE SIMULATION TIME

	Winter 14-Feb	Spring 1-May	Summer 1-Aug	Autumn 1-Nov
ANN	38.60	42.70	35.45	38.70
KNN	1.11	1.20	1.09	1.10

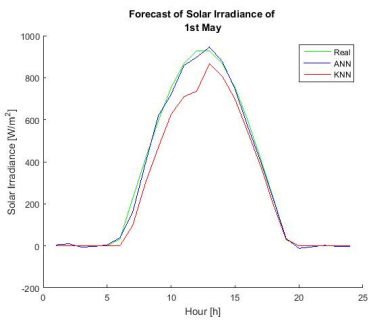

Fig. 5. Prediction of the 1st of May comparing both methods.

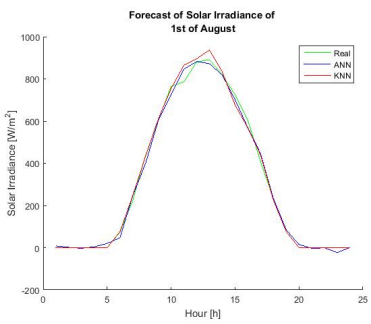

Fig. 6. Prediction of the 1st of August comparing both methods.

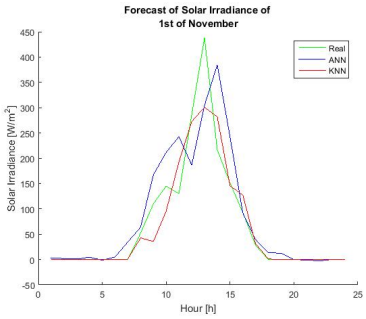

Fig. 7. Prediction of the 1st of November comparing both methods.

make after the comparative test is that both methods perform a better forecast than the Persistence Method. In some cases it reaches the range of less than 10% of MAPE. The ANN method presents results under the range of 30% of MAPE, excepting the forecast of 1st of November which is the worst prediction with a MAPE of 46.12%. The KNN case shows error values lower than ANN method, except for the prediction of 1st of May in which the KNN presents a MAPE of 23.14% while the ANN method presents a MAPE of 8.4%. In general, the period of Spring and Summer presents better results due to the regularity of the Solar Irradiance. Besides this, the forecast of the Solar Irradiance in the seasons of Autumn and Winter is hampered by the irregularity of the Solar Irradiance behavior.

Taking a more detailed analysis of the 14th of February forecast we can see that none of the models predict the second solar peak that occurs between the 14th and the 16th hour. Comparing this case with the Autumn Forecast, that have two peaks as well, we can conclude that maybe this happens due to the quick variation of short duration. The second peak of 14th of February only lasts two hours and the models are not well prepared (or trained) for this kind of variations. If we take more attention to the behavior of each model in this period we can observe that, besides they don't reach the real value of Irradiation, both models react to this variation (in the ANN case de forecast for hour 15 and 16 are equal).

In what concerns to the Simulation Time, the KNN method is clearly quicker when compared with the ANN case. The KNN method, like in the parameters tests, performed the predictions in about 1 second while the ANN method needed between 35 and 42 seconds to perform the same predictions.

In the second test fourteen days were randomly chosen, seven Clear Sky Days and seven Cloudy Days. The criteria used to choose these fourteen days was the behavior of the Solar Irradiance. If the Solar Irradiance profile for a certain day was too irregular we classify it as a Cloudy Day. If the profile had a normal behavior increasing the Solar Irradiance from the Sunrise until the Noon and then decreasing until the Sunset without big irregularities then we classified it as a Clear Sky Day. The results of this test can be observed in Figures 8 and 9 and in Table III. The Figures 10 and 11 presents the best cases of prediction for both models in each situation. The first conclusion to be drawn from the results

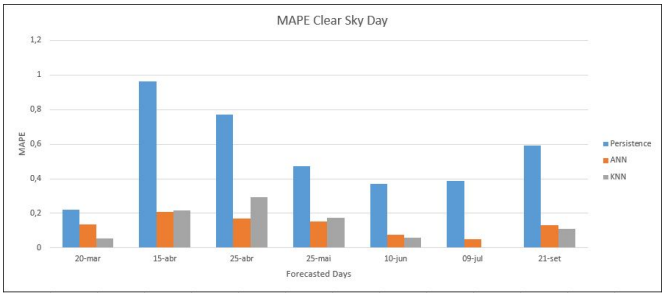

Fig. 8. MAPE error for the Clear Sky Day Test comparing the studied methods.

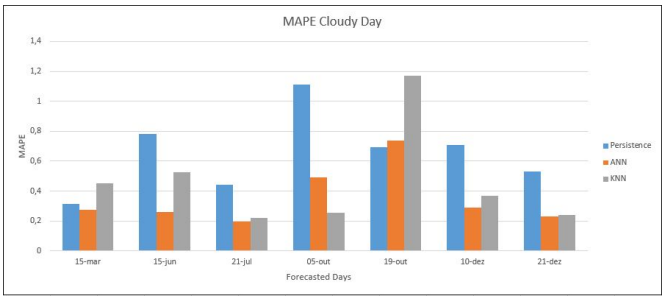

Fig. 9. MAPE error for the Cloudy Day Test comparing the studied methods.

TABLE III
CLEAR AND CLOUD DAY TEST 7 DAYS AVERAGE RESULTS

	Clear Sky Day			
	RMSE $[W/m^2]$	**MAE** $[W/m^2]$	**MAPE**	**Sim. Time** $[s]$
Persistence	134.60	112.31	0.5406	0.00
ANN	51.42	37.93	0.1319	44.79
KNN	48.77	36.60	0.1305	1.06

TABLE IV
CLEAR AND CLOUD DAY TEST 7 DAYS AVERAGE RESULTS

	Cloudy Day			
	RMSE $[W/m^2]$	**MAE** $[W/m^2]$	**MAPE**	**Sim. Time** $[s]$
Persistence	101.79	86.70	0.6537	0.00
ANN	84.67	64.63	0.3549	39.17
KNN	123.67	90.33	0.4616	1.08

is that, as the previous test showed us, both methods are in average better than the Persistence Model. In fact, the MAPE for both models was always below 50%. The only exception is the RMSE and the MAE of KNN to the Cloudy Day Tests which is the worst of the three models. In the Clear Sky Day test, as in the test of the Seasons, the KNN model presented in average better results with MAPE of 13.05% (in four of the seven forecasts the KNN presented the best result). In the Cloudy Days case, in contrast to what we saw in the previous tests, the best method in average is the ANN with a MAPE of 35.49% (in five of the seven forecasts the ANN model made a more accurate prediction than the others).

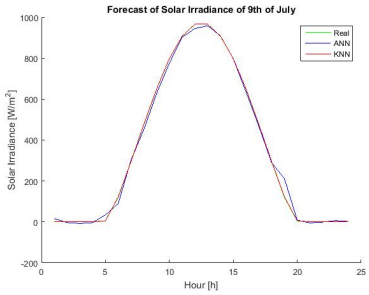

Fig. 10. Clear Sky Day Forecast: Prediction of the 9th of July comparing both methods. ANN's and KNN's best Clear Sky Forecast.

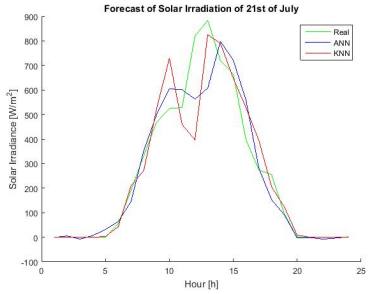

Fig. 11. Cloudy Day Forecast: Prediction of the 21st of July comparing both methods. ANN's and KNN's best Cloudy Day Forecast.

In this test it was very clear the dependency of the KNN model on the targets associated to past records. As we can see in Figure 9 the forecast made by the KNN model reached a MAPE error higher than 100% in the forecast of 19th of October, which is rare for the KNN model. This can be justified with the fact that the training set was not suitable to this case. In fact, if all the last 30 days reached an average Solar Irradiance of $400W/m^2$ and the day we want to forecast will reach a maximum of $200W/m^2$ the model can't adjust itself to this variation from the training set. If we analyze the performance of ANN, besides the MAPE of around 70% for the same day, the model could adjust itself more easily than KNN and at least present a result close to the Persistence model. This ANN characteristic was the main responsible for the better result in comparison with the KNN for the Cloudy Day Forecast test.

Regarding the simulation time, the results obtained in these tests were similar to the previous ones. The ANN model presented an average of $44.79s$ for the Clear Sky Test and $39.17s$ for the Cloudy Day Test, while the KNN model presented for the same tests an average of $1.06s$ and $1.08s$ respectively. This difference is due to the greater need of computational memory and perform many iterations to predict the next irradiance value. While the ANN perform many iterations and an unknown number of them the KNN perform always the same number of distance calculations in each forecast.

V. Conclusions

Throughout this work it was possible to learn more about the Artificial Intelligence Techniques, in particular about the ANN and the KNN models and how these models could be applied to the Solar Irradiance Forecast. In this work we can take some conclusions about the performance of both models.

In first place, the results on the performance tests showed us that both methods are better alternatives to the simple Persistence Model. In terms of results, both methods obtained average forecast errors below 30% of MAPE. The only exception was for the Cloudy Day tests where they got errors between 50% and 30%. The KNN showed the best results by surpassing ANN performance in every tests except for the Cloudy Day Test where he obtained an average of 46.16% and the ANN obtained an average of 35.49%.

Relatively to the Simulation Time the KNN were the quickest of the studied models by performing his forecasts in an average time of one second. The ANN performed in average his forecasts between 45 and 35 seconds. This happens due to the different complexity level of the each algorithm and the different memory needs of each model.

This work was important also to understand the vulnerabilities of both methods in performing Solar Forecast. In the first test was possible to see that these methods have some difficulties to predict small and quick variations in Solar Irradiance during the day. The faster this variation is, the worse it will be for the models to correct. In the second test we could see another handicap for these models, especially for the KNN. These models are dependent on the targets used to train them. If these targets belong to a range of values and the day we want to predict does not even reach irradiances half of these values, then the prediction made for this day will present high errors. This can be well explicit in the KNN forecast for the 19th of October where it reach errors of 100% MAPE. This disadvantage is not so explicit in the ANN due to is capability of adapting and learning from the information given.

To conclude, if we want a fast model that uses few computational resources regardless of its handicap on depending on the targets used to train, the KNN is the recommended method. If time and resources are not a problem and we want to favor a more adaptive method that presents good results in the accuracy of forecasts, the ANN is the best method.

References

[1] Shi, J.; Lee W. J.; Liu, Y.; Yang, Y.; Wang, P.; , "Forecasting Power Output of Photovoltaic Systems Based on Weather Classification and Support Vector Machines," *IEEE Transactions on Industry Applications* , vol.48, no.3, pp. 1064-1069, May 2012

[2] Monteiro, C.; Santos, T.; Fernandez-Jimenez, L. A.; Ramirez-Rosado, I. J.; Terreros-Olarte, M. S.; , "Short-Term Power Forecasting Model for Photovoltaic Plants Based on Historical Similarity," *Energies* , vol.6, no.5, pp.2624-2643, May 2013

[3] Bigdeli, N.; Zandieh, A. H.; , "Solar Irradiance Prediction by a New Forecast Engine Composed Wavelet Packet Transform and Adaptive Neuro-Fuzzy Inference System," *IJECCE*, vol.4, no.6, pp.1598-1606, Nov. 2013

[4] Yona, A.; Senjyu, T.; Saber, A. Y.; Funabashi, T.; Sekine, H.; Kim, C. H.; , "Application of neural network to 24-hour-ahead generating power forecasting for PV system," *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pp.1-6, Jul. 2008

[5] Paoli, C.; Voyant, C.; Muselli, M.; Nivet, M.; , "Forecasting of preprocessed daily solar radiation time series using neural networks," *Solar Energy* , vol.84, no.12, pp.2146-2160, 2010

[6] Mandal, P.; Madhira, S. T. S.; Ul haque, A.shraf; Meng, J.; Pineda, R. L.;, "Forecasting Power Output of Solar Photovoltaic System Using Wavelet Transform and Artificial Intelligence Techniques," in *Procedia Computer Science*, vol.12, 2012, pp. 332 – 337

[7] Lora, A. T.; Santos, J. R.; Santos, J. R.; Ramos, J. L. M.; Exposito, A. G.; , "Electricity Market Price Forecasting: Neural Networks versus Weighted-Distance k Nearest Neighbours," *Database and Expert Systems Applications: 13th International Conference, DEXA 2002 Aix-en-Provence, France, September 2–6, 2002 Proceedings*, p.321-330, 2002

[8] Zhang, Y.; Wang, J.; , "K-nearest neighbors and a kernel density estimator for GEFCom2014 probabilistic wind power forecasting," *International Journal of Forecasting* , vol.32, no.3, pp.1074-1080, 2016

[9] Pedro, H. T. C; Coimbra, C. F. M.; , "Nearest-neighbor methodology for prediction of intra-hour global horizontal and direct normal irradiances," *Renewable Energy*, vol. 80 , pp.770-782, 2015

[10] Castro, R; "Uma Introdução às Energias Renováveis: Eólica, Fotovoltaica e Mini-Hídrica, 1st Edition", IST Press, 2011.

[11] Hagan, M. T.; Demuth, H. B.; Beale, M. H.; De Jesús, O.; , "Neural Network Design, 2nd Edition", Hagan M., 2014.

[12] Hagan, M. T.; Menhaj, M. B.; , "Training Feedforward Networks with the Marquardt Algorithm," *IEEE Transactions on Neural Networks* , vol. 5, no. 6, pp.989-993, 1994

[13] Marques, J. S.; , "Reconhecimento de Padrões: métodos estatísticos e neuronais,", IST Press, vol. 8, 1999.

[14] Duda, R. O.; Hart, P. E.; Stork, D. G.; , "Pattern Classification, 2nd Edition" Wiley-Interscience, John Wiley & Sons Inc, 2001.

[15] Hassanat, A. B.; Abbadi, M. A.; Altarawneh, G. A.; Alhasanat, A. A.; , "Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach,", *(IJCSIS) International Journal of Computer Science and Information Security*, vol.12, no. 8, Aug.2014.