Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 18/19

Luca Di Persio, Matthias Ehrhardt and Beatrice Verzé

Solar Energy Forecasting: from classical statistical approaches to innovative machine learning based models

December 2018

http://www.math.uni-wuppertal.de

Solar Energy Forecasting: from classical statistical approaches to innovative machine learning based models

Luca Di Persio[1], Matthias Ehrhardt[2], and Beatrice Verzé[3]

[1] *Chair of Probability and Mathematical Statistics, University of Verona, Strada le Grazie 15, 37134 Verona, Italy*
[2] *Lehrstuhl für Angewandte Mathematik und Numerische Analysis, Bergische Universität Wuppertal, Gaussstrasse 20, 42119 Wuppertal, Germany*
[3] *University of Verona, Strada le Grazie 15, 37134 Verona, Italy*

Abstract

The role played by electricity (production, distribution, consumption) is undoubtedly a key point within the current energy trading framework. The growing demand for energy and the effects of CO_2 emissions to the climate change is the main reason which led finally to the integration of renewable energies into electricity grids.

As the need for energy is constantly increasing, it is important to increase the production of energy from photovoltaic (PV) systems. As a result, forecasting information becomes important to ensure the efficient use of PV systems and accurate predictions of changes in solar radiation can improve service quality. This integration of solar energy and accurate forecasting can help to improve the energy planning and distribution.

In this paper we will mainly address the competition GEFCom2014, using the data referring to the forecast part of solar energy. In particular we will exploit the Artificial Neural Networks (ANN) approach and Seasonality AutoRegressive Integrate Moving Average (SARIMA) models.

1 Introduction

The last decades have witnessed an increasing interest for the development and exploitation of power sources, both from the industrial and the private point of view. In order to ensure an efficient production, centralized regulation of the electricity supply industry was deemed necessary.

To better understand such a challenge we have focused our attention on the solar energy production. In particular, we show how rigorous mathematical models developed exploiting classic statistics and stochastic tools as well as advanced neural networks (NNs) approaches are able to produce accurate forecasts.

Solar and wind energy have fundamental different production characteristics compared to energy produced from conventional sources, such as fossil fuels. While the electricity production of the latter can be adapted to the given electricity demand, the availability of solar and wind energy is largely determined by prevailing weather conditions by high variability because of, e.g., difference between day/night and summer/winter, wind direction, melting snow in summer, rainfall in winter, etc.

In recent years, a number of challenges have been posed to various financial companies all over the world, in order to try to design the best methods for forecasting energy price. The two most significant competitions were: GEFCom2012 [29], and GEFcom2014 [28].

Here we will mainly address the 2014 competition, using the data referring to the forecast part of solar energy which comes from an area, associated to one solar power plant located in a certain region of Australia.

The present work is structured as follows: in Section 2 and 3 we motivate the first model, namely the Artificial Neural Networks (ANN) methods and the second model, that is the Seasonality AutoRegressive Integrate Moving Average (SARIMA) model, and explain the way they are designed. The comparison between them is given in Section 4, with the consecutive considerations. In Section 5 we discuss the special case of the *deseasonalization* of ANNs.

2 Artificial Neural Networks Models

We have studied two special cases of Artificial Neural Networks: the *NAR Network* and the *NARX Network*. Artificial Neural Networks (ANNs) are models that benefit from an inspiration from the human brain. For interested readers we refer to the article [26].

We fixed the *structure* of the network choosing the delays and the number of hidden neurons. The optimal number of hidden neurons is based on a complex relationship between the number of input and output nodes, the amount of training data available, the complexity of the function that is trying to be learned and the training algorithm. To minimize the error and to have a trained network that generalizes well, we need to pick an optimal number of hidden neurons. Too few nodes will lead to high error for the system as the predictive factors might be too complex for a small number of nodes to capture. Too many nodes will overfit to the training data and not generalize well. To this end and according with our goals and related constraints, we have considered ANNs characterized by 2 and 4 delays, with 5, 10, 15 hidden neurons. Moreover we adopted, as activation function for each node, the Sigmoid Function.

Then we train the model trough an appropriate algorithm. The most common, and the one we have used, is the *Levenberg-Marquardt*, for sample application see e.g. [45] and [12], which is a type of beckpropagetion described in [56] and [1]. In Matlab it is called *'trainlm'*, which is often the fastest backpropagation algorithm in the toolbox, and is highly recommended as a first-choice supervised algorithm.

In what follows, we provide a description of the two models we have used in our analysis, namely the Nonlinear AutoRegressive Neural Networks (NAR) model and Nonlinear AutoRegressive Neural Networks with exogenous input (NARX) model. We refer the interested reader to [26] for further details and application examples.

We distinguish the NAR and the NARX between two cases: *Open loop* and *Closed loop*.
Open loop does not react on the feedback to obtain the output. This is why it is also called a non-feedback control system. There are no disturbances or variations in this system and works on fix conditions. The closed loop records the output instead of input and modify it according to the need. It generates preferred condition of the output as compared to the original one. It does not encounter any external or internal disturbances. Some differences between these two system are:

- The cost of open loop systems is low than in closed loop.
- The open loop can be easily implemented, while the closed loop is difficult to implemented.
- The open loop are more stable than the closed loop.

Hence open loop is simple and works on the input while closed loop is complex and works on the output and modifies it.
In the next sections we will see that the closed loop gives a higher error compared to the open loop.

2.1 NAR Networks

Nonlinear AutoRegressive Neural Networks and NAR Networks are defined by a functional difference.

$$y(t) = f\big(y(t-1), \dots, y(t-d)\big).$$

In this case the future values of the time series $y(t)$ are a forecast only from past values of the same time series. In fact, NAR networks can be trained to predict a time series from that series past values. This time series will be divided into three parts:

- *Training part:* we have chosen 70% of the time series, which represent the input y_t;
- *Validation part:* we have chosen 15% of the time series;
- *Testing part:* we have chosen 15% of the time series.

Once the time series has been divided we can train the network using the *Levenberg-Marquardt* algorithm. To evaluate the performance of the NAR Network we estimate the error, namely the Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}, \tag{1}$$

where y_t, $t = 1, \dots, N$ are the real values and $\hat{y}_t$, $t = 1, \dots, N$ are the forecast values, where N denotes the number of observations. We used this RMSE error because the data are of photovoltaic (PV) origin and are normalized, hence the quantities are all between 0 and 1, so the square of a number less than 1 reduces the error without having a good forecast.

2.1.1 Results using a NAR Network

The following three tables show the RMSE to see in which case we obtain the better results using a training period of: a whole year from 1 April 2012 to 1 April 2013, the spring-summer period form 1 October 2012 to 1 April 2013, this is due to the fact that the data comes from a place in Australia, and the autumn-winter period from 1 April 2012 to 1 October 2012; to forecast one week. Looking at Table 1 we note that the minimum error is obtained in the case where we have 4 delays and 10 hidden layers, that is 0.1049. While for the closed loop the best performance is obtained in the case where we have 2 delays and 10 hidden layers, that is 0.1945.

NAR Network RMSE 2012-2013			
Delays	**Hidden neurons**	**Open loop**	**Closed loop**
1:2	5	0.1115	0.2958
	10	0.1116	**0.1945**
	15	0.1110	0.2997
1:4	5	0.1058	0.2979
	10	**0.1049**	0.2707
	15	0.1051	0.2621

Table 1: RMSE of NAR Network Model of the whole year 2012-2013.

In Table 2 we can see that in the spring-summer period the best performance for open loop is in the case of 4 delays and 10 hidden layers, that is 0.1072. While for the closed loop is in the case with 2 delays and 15 hidden layers, that is 0.1596. In Table 3, in autumn-winter period, we note that the minimum error for open loop is obtained in the case where we have 4 delays and 15 hidden layers, that is 0.1027. While for closed loop the best performance is in the case of 2 delays and 10 hidden layers, that is 0.1965.

NAR Network RMSE 2012-2013 Spring-Summer			
Delays	**Hidden neurons**	**Open loop**	**Closed loop**
1:2	5	0.1135	0.2616
	10	0.1162	0.3295
	15	0.1141	**0.1596**
1:4	5	0.1105	0.1835
	10	**0.1072**	0.2485
	15	0.1088	0.2505

Table 2: RMSE of NAR Network Model for spring/summer of 2012-2013.

Looking at Tables 1, 2 and 3, we note that in every case the open loop structure of the NAR network give a better performance than the closed loop structure of the NAR network.

1. In the open loop case we obtained a better error whit four delays with respect the case with two delays.

NAR Network RMSE 2012-2013 Autumn-Winter			
Delays	**Hidden neurons**	**Open loop**	**Closed loop**
1:2	5	0.1090	0.2464
	10	0.1083	**0.1965**
	15	0.1076	0.1979
1:4	5	0.1046	0.2959
	10	0.1035	0.2391
	15	**0.1027**	0.2487

Table 3: RMSE of NAR Network Model for autumn/winter of 2012-2013.

2. While in the closed loop we obtained a better result with two delays with respect the case with four delays.

About the number of hidden neurons we can say that in most cases 5 hidden neurons give a worst result, hence is more appropriate to use an higher number of hidden neurons like 10 or 15. At the same time, using a very high number of hidden neurons is not necessary: the error between 10 and 15 hidden neurons is not so different and in some cases 10 hidden neurons gives a smaller error than 15 hidden neurons.

Now we pass to analyze the solar power for the whole year from 1 April 2013 to 1 April 2014, for the spring-summer period form 1 October 2013 to 1 April 2014 and for the autumn-winter period from 1 April 2013 to 1 October 2013. Let us recall that the data come from an area in Australia.

If we observe the Table 4 we note that the minimum error in open loop is obtained in the case where we have 4 delays and 5 hidden layers, that is 0.1095. While for closed loop the best performance is in the case of 2 delays and 15 hidden layers, that is 0.1741. In Table 5 we can see that in spring-summer period the best performance in open loop is in the case of 4 delays and 15 hidden layers, that is 0.1096. While for the closed loop the minimum error is in the case of 4 delays and 5 hidden layers, that is 0.1861. In Table 6, in autumn-winter period, we can note that the minimum error for open loop is obtained in the case where we have 4 delays and 15 hidden layers, that is 0.1083. While for the closed loop the minimum error is in the case of 4 delays and 10 hidden layers, that is 0.1546.

If we look at Tables 4, 5 and 6, we note that in every case the open loop structure of the NAR network yields a better performance than the closed loop structure of the NAR network. Furthermore, again, we can observe that

NAR Network RMSE 2013-2014			
Delays	**Hidden neurons**	**Open loop**	**Closed loop**
1:2	5	0.1162	0.2793
	10	0.1182	0.2770
	15	0.1167	**0.1741**
1:4	5	**0.1095**	0.2536
	10	0.1101	0.2500
	15	0.1103	0.6632

Table 4: RMSE of NAR Network Model of the whole year 2013-2014.

NAR Network RMSE 2012-2013 Spring-Summer			
Delays	**Hidden neurons**	**Open loop**	**Closed loop**
1:2	5	0.1191	0.2228
	10	0.1166	0.3175
	15	0.1150	0.1987
1:4	5	0.1125	0.1861
	10	0.1101	0.2591
	15	0.1096	0.2525

Table 5: RMSE of NAR Network Model for spring/summer of 2013-2014.

NAR Network RMSE 2012-2013 Autumn-Winter			
Delays	**Hidden neurons**	**Open loop**	**Closed loop**
1:2	5	0.1132	0.3241
	10	0.1131	0.3092
	15	0.1128	0.1648
1:4	5	0.1101	0.2054
	10	0.1101	0.1546
	15	0.1083	0.2410

Table 6: RMSE of NAR Network Model for autumn/winter of 2013-2014.

in the case of an open loop we obtain a better result with 4 delays. Also in this case we can say the same considerations apply as those made in the previous case.
Contrary, in the closed loop case is more difficult to find a scheme.

2.2 NARX Networks

The Nonlinear AutoRegressive Neural Network with eXternal input (NARX) was proposed by Lin, Horne, Tino and Giles [39]. Differently from the NAR Network, to predict the future values of a time series $y(t)$ we do not have only the past values of this time series but another time series $x(t)$ come into play. In fact, NARX can learn to predict one time series given past values of the same time series, the feedback input, and another time series, called the eXternal or eXogenous time series. Thus the NARX Network uses two time series of the same length as input. In our case we used as external time series the corresponding period of $y(t)$ in the year before. In particular a NARX model is characterized by

$$y(t) = f\big(y(t-1), \ldots, y(t-d), x(t-1), \ldots, x(t-d)\big).$$

As in the case before, we evaluate the performance trough the error, in particular trough the RMSE error (1).

2.2.1 Results using a NARX Network

The Table 7 illustrates the results for the whole year from 1 April 2013 to 1 April 2014, for which the best performance for an open loop is obtained in the case of 4 delays and 10 hidden layers, that is 0.1061. While for a closed loop the minimum error is given by 4 delays and 10 hidden layers, that is 0.1173. In Table 8 we can see that in spring-summer period the best performance for open loop in this period is the one with 2 delays and 15 hidden layers, that is 0.1039. While for a closed loop the minimum error is given by 4 delays and 15 hidden layers, that is 0.1192. While in autumn-winter period, the best performance in an open loop is obtained in the case of 2 delays and 10 hidden layers, that is 0.1021. While for a closed loop the minimum error is given by 2 delays and 5 hidden layers, that is 0.1127. Looking at Tables 7 and 8, we can see that in every case we obtained a better performance with an open loop structure than with a closed loop structure.

In the case in which the training period is one year the consideration made previously, with the NAR model, are found again, that is, the model seems to work better with 4 delays and ten hidden neurons. While in the case

RMSE of NARX			
Delays	**Hidden neurons**	**Open loop**	**Closed loop**
1:2	5	0.1096	0.1216
	10	0.1085	0.1230
	15	0.1091	0.1203
1:4	5	0.1085	0.1177
	10	**0.1061**	**0.1173**
	15	0.1070	0.1200

Table 7: RMSE for whole year 2013-2014.

NARX Network RMSE 2013-2014					
Delays	**Hidden neurons**	**Spring-Summer**		**Autumn-Winter**	
		Open loop	**Closed loop**	**Open loop**	**Closed loop**
1:2	5	0.1072	0.1266	0.1054	**0.1127**
	10	0.1057	0.1334	**0.1021**	0.1165
	15	**0.1039**	0.1257	0.1067	0.1170
1:4	5	0.1115	0.1269	0.1040	0.1177
	10	0.1043	0.1235	0.1251	0.1469
	15	0.1041	**0.1192**	0.1066	0.1281

Table 8: RMSE of NARX Network Model for spring/summer of 2013-2014.

in which the training period is of 6 months, we do not find a correspondence with the previous case, and the model seems to work better with 2 delays.

2.2.2 Comparison of different training periods of NAR and NARX

Since we consider a training period of 1 year from 1 April 2013 to 1 April 2014, the forecast period that we obtain corresponds to the forecast one that we have when we use a training period of six months from 1 October 2013 to 1 April 2014.
Hence we compare the RMSE obtained for the different training period for the NAR and NARX models.

We cannot say a lot about which is the better case, in fact in the two cases of the NAR model we can observe that the model seems to works better when we use 1 year as training period, but this is not the case when we use

RMSE NAR			
Delays	**Hidden Neurons**	**Whole 2012-2013**	**Spring-Summer**
1:4	10	**0.1049**	0.1072

RMSE NAR			
Delays	**Hidden Neurons**	**Whole 2013-2014**	**Spring-Summer**
1:4	5	**0.1095**	0.1125

RMSE NARX			
Delays	**Hidden Neurons**	**Whole 2013-2014**	**Spring-Summer**
1:4	10	0.1061	**0.1043**

the NARX method.

3 SARIMA Model

In this section we analyze the solar power forecast trough a statistical based approach, that is the Seasonal AutoRegressive Integrated Moving Average Model, SARIMA model, proposed by Contreras in [15], and widely used in a series of applications ranging from finance to biology, from physics to social media analysis, etc. see, e.g., [30, 13].

This model emerges from the more simple approach called ARMA model, which includes an AutoRegressive part (AR) and a Moving Average part (MA). A pth-order Autoregressive model, AR(p), is defined as:

$$L_t - \sum_{i=1}^{p} \phi_i L_{t-1} = \epsilon_t, \tag{2}$$

where ϵ_t is a random prediction error, which is assumed to be Gaussian with zero mean and σ^2 variance, $\phi_1, ..., \phi_p$ are the unknown AR coefficients.
The order of the model, p, tell us how many lagged past values are included.
The simplest AR model is the first-order AR(1).
The Moving Average model of order q, MA(q), is defined as:

$$L_t = \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-1}. \tag{3}$$

In this model the time series is regarded as a moving average of a random shock series ϵ_t.

If we combine the AR model with MA model we obtain the ARMA model, as said before. An ARMA model of order p and q, ARMA(p,q), is given by:

$$L_t - \sum_{i=1}^{p} \phi_i L_{t-1} = \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-1}. \tag{4}$$

The ARMA model assumes that the time series is stationary. If it is not the case, we need to transform the series into a stationary one. A way to compute this transformation using differencing was introduced in 1976 by Box and Jenkins [8]. Box and Jenkins included explicitly the differencing part in the formulation, hence the ARMA model becomes the ARIMA, or Box-Jenkins model characterized by three types of parameter:

- the autoregressive parameters $(\phi_1, \ldots, \phi_p)$;
- the number of differencing passes at lag 1, which transform the non stationary time series in a stationary one, d;
- the moving average parameters $(\theta_1, \ldots, \theta_q)$.

The equation of the ARIMA(p, d, q) model is given by:

$$\phi(B)\nabla^d L_t = \theta(B)\epsilon_t, \tag{5}$$

where $\nabla x_t \equiv (1 - B)x_t$ is the lag 1 differencing operator, a particular case of a more general lag-h differencing operator given by $\nabla_h x_t \equiv (1 - B^h)x_t \equiv x_t - x_{t-h}$. B denotes the backward shift operator, that is $B^h x_t \equiv x_{t-h}$. While $\phi(B)$ is a shorthand notation for

$$\phi_p(B) = 1 - \phi_1 B - \cdots - \phi_p B^p \tag{6}$$

and $\theta(B)$ is a shorthand notation for

$$\theta_q(B) = 1 - \theta_1 B - \cdots - \theta_q B^q. \tag{7}$$

A special case is the ARIMA$(p, 0, q)$ model, which is equal to ARMA(p, q) model.
The structure of ARMA and ARIMA models are illustrated in Figure 1.

In the case we have a seasonal component, such as in our case with solar energy, we needed to add another parameter. Then from the ARIMA model we arrive at the Seasonal ARIMA model (SARIMA). The SARIMA model with both seasonal and non-seasonal parameters is the ARIMA$(p, d, q) \times (P, D, Q)_s$. Where (p, d, q) gives the order of the non-seasonal part, while $(P, D, Q)_s$ denotes the seasonal part. The parameter s represents the number

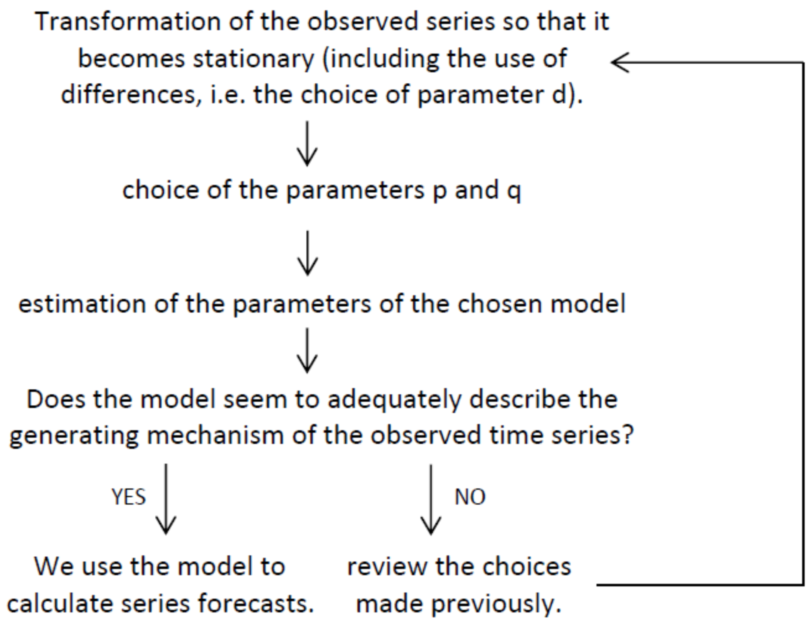

Figure 1: Structure of ARMA and ARIMA.

of observations in the seasonal pattern, in our case we consider hourly series with daily periodicity, hence the value of s is equal to 24. If we consider monthly data, s will be equal to 12. The SARIMA model is defined by

$$\phi(B)\Phi(B^s)\nabla^d\nabla_s^D L_t = \theta(B)\Theta(B^s)\epsilon_t. \tag{8}$$

Also in this case to evaluate the performance of the model we used the RMSE error.

3.1 Results using the SARIMA model

We tested the model using p and $q \in \{1, 2, 3, 4\}$, and $d = D = 1$.
We decided to report the best performance obtained, hence with the combination of $(2, 1, 1) \times (1, 1, 1)_{24}$. We will show the forecast performance through the RMSE for the years 2013 and 2014, both considering the whole year and a particular time window, that is the seasons, hence dividing the year in the spring-summer period and in the autumn-winter period.

3.1.1 Forecast of one week

The Table 9 shows the RMSE obtained by the implementation of the model using the whole year 2012-2013, the spring-summer period and the autumn-winter period as training period to forecast one week. In this case we note that with a longer training period we get a smaller error than in the two autumn-winter, spring-summer cases. In any case, this result could be affected by climatic conditions. In fact looking at Table 10 we observe that we obtain the same result in the following year, i.e. a smaller error in the case of one year of training than in sixth months. But we cannot say the same when we look at the error in the case of two years of training in Table 11. In fact, here the error is greater than both the whole year from 1 April 2012 to 1 April 2013 and the whole year from 1 April 2013 to 1 April 2014.

SARIMA RMSE of 2012-2013			
$(p,d,q)\times(P,D,Q)_s$	**RMSE**		
	whole 2013	**Spring/ Summer**	**Autumn/ Winter**
$(2,1,1)\times(1,1,1)_{24}$	0.1333	0.1585	0.1864

Table 9: *RMSE for the years 2012-2013.*

The Table 10 shows the RMSE obtained by the implementation of the model using the whole year 2013-2014, the spring-summer period and the autumn-winter period as training period to forecast one week. The Table 11

SARIMA RMSE of 2013-2014			
$(p,d,q)\times(P,D,Q)_s$	**RMSE**		
	whole 2014	**Spring/ Summer**	**Autumn/ Winter**
$(2,1,1)\times(1,1,1)_{24}$	0.1284	0.1710	0.1382

Table 10: *RMSE for the years 2013-2014.*

shows the RMSE obtained by the implementation of the model using the two whole years 2012-2013 and 2013-2014 as training period to forecast one week.

3.1.2 Forecast of one day, 24 hours.

The Table 12 shows the RMSE error obtained by the implementation of the SARIMA model using the two whole years 2012-2013, the spring-summer period and the autumn-winter period as training period to forecast one day. The

SARIMA RMSE of 2012-2014	
$(p,d,q)\times(P,D,Q)_s$	**RMSE** **whole 2012-2014**
$(2,1,1)\times(1,1,1)_{24}$	0.1405

Table 11: *RMSE for the years 2012-2014.*

SARIMA RMSE of 2012-2013			
$(p,d,q)\times(P,D,Q)_s$	**RMSE**		
	whole 2013	**Spring/ Summer**	**Autumn/ Winter**
$(2,1,1)\times(1,1,1)_{24}$	0.0842	0.1469	0.1846

Table 12: RMSE for the year 2012-2013.

Table 13 shows the RMSE obtained by the implementation of the SARIMA model using the two whole years 2013-2014, the spring-summer period and the autumn-winter period as training period to forecast one day. The last

SARIMA RMSE of 2013-2014			
$(p,d,q)\times(P,D,Q)_s$	**RMSE**		
	whole 2014	**Spring/ Summer**	**Autumn/ Winter**
$(2,1,1)\times(1,1,1)_{24}$	0.1692	0.2160	0.1389

Table 13: RMSE for the years 2013-2014.

Table 14 shows the RMSE obtained by the implementation of the SARIMA model using the two whole years 2012-2013 and 2013-2014 as training period to forecast one day.

In this case using a longer period as training period we obtain a better performance. In fact, if we look at Tables 9 and 10 we can see that we have a smaller RMSE in the case that the training period is one year than when it is only six months, and we still get a smaller RMSE error if the training period is two years, as we can see in Table 11.

<table>
<tr><th colspan="2">SARIMA RMSE
of 2012-2014</th></tr>
<tr><td rowspan="2">$(p,d,q) \times (P,D,Q)_s$</td><td>RMSE</td></tr>
<tr><td>whole
2012-2014</td></tr>
<tr><td>$(2,1,1) \times (1,1,1)_{24}$</td><td>0.1310</td></tr>
</table>

Table 14: RMSE for the years 2012-2014.

4 ANNs vs. SARIMA

In this section we compare the forecast performance obtained using both the NAR and NARX models against these reached through the SARIMA approach. As in the previous cases, we use the RMSE error to evaluate the performance of each model. We compare these three models through three different cases, which correspond to the different training periods:

1. for a period from 1 April 2013 to 1 April 2014;
2. for the autumn - winter period from 1 April 2013 to 1 October 2013;
3. for the spring - summer period from 1 October 2013 to 1 April 2014.

For every case we forecast 168 hours, that is a week. In each model we consider the best performance obtained: regarding the NAR Network method, we use in the first case 4 delays and 5 hidden layers, while in the second and in the third case we use 4 delays and 15 hidden layers. In the case of a NARX Network we use for the first and second case 4 delays and 10 hidden layers, while in the third case we use 4 delays and 10 hidden layers. In the last method, that is for the SARIMA model, we use the case in which the parameters are $p = 2$ and $P = Q = q = 1$.

The Results are illustrated through Tables 15-17 and Figures 2-4. In the Table 15 we illustrate the result for the whole year from 1 April 2013 to 1 April 2014, with the respective comparison between real and forecast values obtained by the implementation of the models which are shown in Figure 2.

Looking at the Figure 2 we note that the real values changes its behaviour during the week. In the first three days the power values do not exceed 0.23, probably because there was the presence of clouds, while in the last four days the power values increase until they arrive to the peak of 0.76 in the sixth day. This change of behaviour causes a difficulty for models to follow the

RMSE 01/04/2013 - 01/04/2014	
NAR	0.1084
NARX	0.1069
SARIMA	0.1473

Table 15: RMSE of NAR, NARX Network and SARIMA Models.

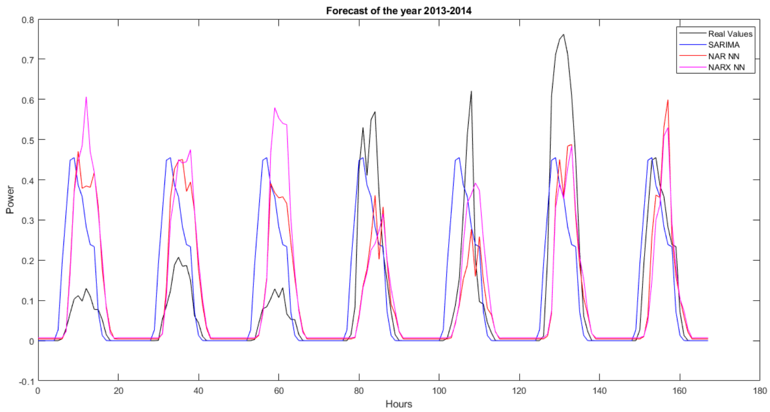

Figure 2: Comparison between ANN and SARIMA model for the whole year.

real values. In fact all of them follow a medium trend, not so low in the first three days nor so high when there is a spike as in the sixth day.

In Table 16 we present the result for the autumn-winter period from 1 April 2013 to 1 October 2013, with the respective comparison between real and forecast values obtained by the implementation of the models which are shown in Figure 3. Differently from the Figure 2, in the Figure 3

RMSE 01/04/2013 - 01/10/2013	
NAR	0.1064
NARX	0.1035
SARIMA	0.1248

Table 16: RMSE of NAR, NARX Network and SARIMA Models.

the behaviour of the real values follows a typical trend of a sunny day, and therefore the optimal condition to get more solar energy. In this case all the

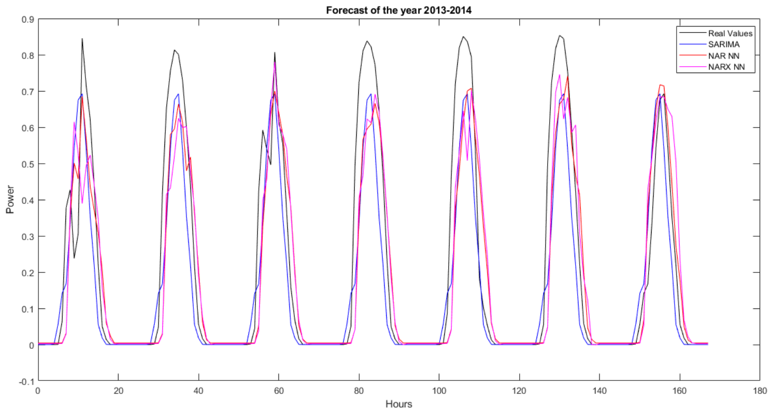

Figure 3: Comparison of ANN and SARIMA model for the autumn-winter period.

models give a better forecast than the previous one. This is also confirmed by Tables 15 and 16.

In the Table 17 we illustrate the result for the spring-summer period from 1 October 2013 to 1 April 2014, with the respective comparison between real and forecast values obtained by the implementation of the models which are shown in Figure 4.

RMSE 01/10/2013 - 01/04/2014	
NAR	0.1102
NARX	0.1091
SARIMA	0.1751

Table 17: RMSE of NAR, NARX Network and SARIMA Models.

If we look at the tables above, we can see that in any case the best performance is given by the NARX Network model, the second by the NAR Network model and the worst one by the SARIMA model.

Since we consider a training period of one year from 1 April 2013 to 1 April 2014, the forecast period that we obtain correspond to the forecast one that we have when we use a training period of six months from 1 October 2013 to 1 April 2014. Hence we compare these two cases to understand in which one we obtain a better performance, i.e., when we use a larger training period or a smaller one.

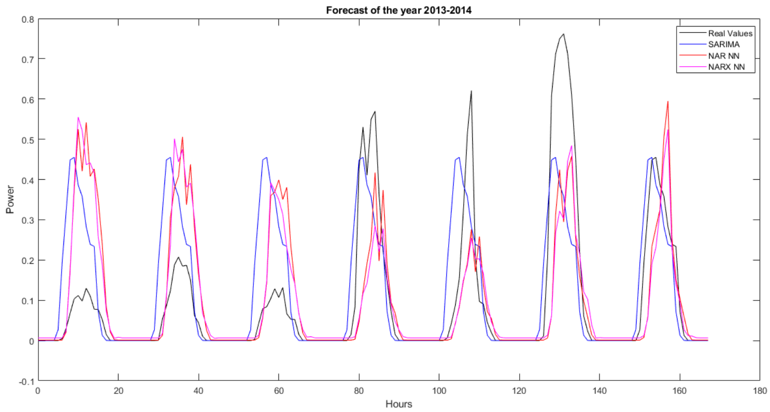

Figure 4: Comparison of ANN and SARIMA model for the spring-summer period 2013-2014.

Observing Figures 2 and 4 we observe that we obtain a better forecast in all the models using a training period of a year rather than of 6 months. This can be noted also in Tables 15 and 17.

5 Deseasonalization

In this section we will consider the *deseasonalization* approach. We will discuss the topic of seasonal adjustment, or *deseasonalization*, trying to remove the effects of seasonality through the use of the *Weron function* to verify if the NAR Network and the NARX Network give a better performance. In Matlab the Weron function is called *deseasonalize* function, you can find the details on how it works in [54, Chapter 2, Section 3 and 4].

There are some important considerations to take into account when using the Weron function. In fact, it contains some parameters, in particular the *long-term seasonal component (LTSC)* and the *short-term seasonal component (STSC)*, respectively called in the function as MET1 and MET2, which define the long-term and the short-term seasonal decomposition. The long term seasonal component plays an important role which does not always have the right recognition, as specified in the article [35]. Since we use a year and six months as a training period to make the forecasts, we have taken into account the long term seasonal component in changing data trends.

We applied the Weron function to the time series starting from 01/04/2013 until 01/04/2014 in the first case, starting from 01/04/2013 until 01/10/2013 in the second case and starting from 01/10/2013 until 01/04/2014 in the third case, and it returns the time series deseasonalized using the method of the *Wavelet decomposition* or *Sinusoid methods*, [35]. After this we compared the performance obtained from the NAR Network and the NARX Network with and without the seasonality. This comparison was always made according to the RMSE error metric.

The Table 18 below gives a comparison between the NAR and NARX Network with and without the seasonality.

01/04/2013 - 01/04/2014		
	RMSE with seasonality	**RMSE without seasonality**
NAR	0.1092	0.0996
NARX	0.1125	0.1035

Table 18: RMSE of NAR and NARX models with and without seasonality.

The Table 19 presents the comparison between the NAR and NARX Network with and without the seasonality.

01/04/2013 - 01/10/2013		
	RMSE with seasonality	**RMSE without seasonality**
NAR	0.1085	0.1083
NARX	0.1070	0.1005

Table 19: *The RMSE of NAR and NARX models with and without seasonality.*

The Table 20 states the comparison between the NAR and NARX Network with and without the seasonality.

01/10/2013 - 01/04/2014		
	RMSE with seasonality	**RMSE without seasonality**
NAR	0.1103	0.1003
NARX	0.1059	0.0989

Table 20: RMSE of NAR and NARX models with and without seasonality.

If we look at the tables above, we can see that in the case without the seasonality we get a better performance both in the NARX Network model and in the NAR Network model compared to the case with the seasonality. This is some evidence of what some researchers, like Nelson et al. (1999) [44], Zhang and Kline(2007) [56], have staited, namely that by removing the seasonal effects from the data, it is possible to obtain a better ANN forecasting.

However, both cases of ANN give a good performance of the solar energy even taking into account the seasonality. As we saw in the Section 4, where we compared the RMSE error between ANNs and SARIMA Model, the error in ANNs models is smaller than in the SARIMA model.

In fact, not all the researchers think that eliminating the seasonality from the time series yields a better performance, such as Franses and Draima [23] and Alon et al. [2], which found that without removing seasonal effects, ANNs are able to model seasonal and trend effects in the data structure.

6 Conclusion and Outlook

In this work we have shown that ANNs based models can be efficiently applied to solve the forecast problem related to solar energy behaviour. In particular we have considered both NAR and NARX architectures, finding their characterizing parameters according to peculiarities of analyzed time series. Moreover we have tested the obtained performances versus these derived by using one of the most powerful stochastic method for (energy) time series analysis, namely the SARIMA model.

We tried to forecast the solar energy trough two relevant models. But there exsist alternative research efforts that found a good prediction trough the use of different methods, as in the case treated in the article [4], where Peder, Henrik and Aalborg used the clear sky modeling with statistical smoothing techniques. Or in the case described in the article of Chaturvedi and Isha [14], where the forecasts are based on numerical weather prediction (NWP) models. Given the increase in the production of energy from PV plants, and given the importance of being able to predict with increasing precision the trend of energy production, helping in the planning and distribution of solar energy, it is important to continue research in this area and search for a model that yields a better solar energy forecasting.

References

[1] Aggarwal, S.K., Saini, L.M., & Kumar, A. (2009); *Electricity price forecasting in deregulated markets: A review and evaluation.* International Journal of Electrical Power and Energy Systems, 31, 13-22.

[2] Alon, I., Sadowski, R.J., & Qi, M. (2001). *Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods.* Journal of Retailing and Consumer Services 8(3), 147-156.

[3] Amjady, N. (2006). *Day-ahead price forecasting of electricity markets by a new fuzzy neural network.* IEEE Transactions on Power Systems, 21, 887-996.

[4] Bacher, P., Madsen, H., & Nielsen, N.A. (2009). *Online short-term solar power forecasting.* Solar Energy, 83(10), 1772-1783.

[5] Baldick, R., Grant, R., & Kahn, E. (2004). *Theory and application of linear supply function equilibrium in electricity markets.* Journal of Regulatory Economics, 25(2), 143-167.

[6] Barlow, M. (2002). *A diffusion model for electricity prices.* Mathematical Finance, 12, 287-298.

[7] Batlle, C. (2002). *A model for electricity generation risk analysis.* Ph.D. thesis, Madrid: Universidad Pontificia de Comillas.

[8] Box, G.E.P., & Jenkins, G.M. (1976). *Time series analysis: forecasting and control.* San Francisco: Holden-Day.

[9] Bollerslev, T. (1986). *Generalized autoregressive conditional heteroscedasticity.* Journal of Econometrics, 31, 307-327.

[10] Brockwell, P.J., & Davis, R.A. (1996). *Introduction to time series and forecasting.* (2nd ed.). New York: Springer-Verlag.

[11] Cartea, A., & Figueroa, M. (2005). *Pricing in electricity markets: a mean reverting jump diffusion model with seasonality.* Applied Mathematical Finance, 12(4), 313-335.

[12] Catalao, J.P.S., Mariano, S.J.P.S., Mendes, V.M.F., & Ferreira, L.A.F.M. (2007). *Short-term electricity prices forecasting in a competitive market: a neural network approach.* Electric Power Systems Research, 77, 1297-1304.

[13] Chae, J., Thom, D., & Bosch, H., (2012). *Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition.* IEEE Conference on Visual Analytics Science and Technology.

[14] Chaturvedi, D.K., & Isha (2016). *Solar Power Forecasting: A Review.* International Journal of Computer Applications (0975 - 8887) Volume 145 - No.6.

[15] Contreras, J., Espinola, R., Nogales, F.J., & Conejo, A.J. (2003). *ARIMA models to predict next-day electricity prices.* IEEE Transactions on Power Systems, 18(3), 1014-1020.

[16] Day, C.J., Hobbs, B.F., & Pang, J.S. (2002). *Oligopolistic competition in power networks: a conjectured supply function approach.* IEEE Transactions on Power Systems, 17(3), 597-607.

[17] Di Franco, R., Di Persio, L., & Ehrhardt, M., (2017). *Forecasting German Electricity Spot Prices using Neural Network, SARIMA Models and Markov Switching Method.* Manuscript, 2017.

[18] Dordonnat, V., Pichavant, A., & Pierrot, A., (2016). *GEFCom2014 probabilistic electric load forecasting using time series and semi-parametric regression models.* International Journal of Forecasting, 32, 1005-1011.

[19] Dudek, G., (2016). *Multilayer perceptron for GEFCom2014 probabilistic electricity price forecasting.* International Journal of Forecasting, 32, 1057-1060.

[20] Durbin, J., & Koopman, S.J., (2001). *Time series analysis by state space methods.* Oxford University Press.

[21] Engle, R.F. (1982). *Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation.* Econometrica, 50, 987-1007.

[22] Fan, S., Mao, C., & Chen, L. (2007). *Next-day electricity-price forecasting using a hybrid network.* IET Proceedings of Generation, Transmission and Distribution, 1(1), 176-182.

[23] Franses, P.H., & G. Draisma, (1997). *Recognizing changing seasonal patterns using artificial neural networks.* Journal of Econometrics 81(1): 273-280.

[24] Gareta, R., Romeo, L.M., & Gil, A. (2006); *Forecasting of electricity prices with neural networks*. Energy Conversation and Management, 47, 1770-1778.

[25] Habena, S., & Giasemidis, G. (2016). *A hybrid model of kernel density estimation and quantile regression for GEFCom2014 probabilistic load forecasting*. International Journal of Forecasting, 32, 1017-1022.

[26] Haykin, S. (2001). *Neural Networks A Comprehensive Foundation*. Second Edition, McMaster University Hamilton, Ontario, Canada.

[27] Hong, T., & Fan, S. (2016). *Probabilistic electric load forecasting: A tutorial review*. International Journal of Forecasting, 32, 914-938.

[28] Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., & Hyndman, R.J. (2016). *Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond*. International Journal of Forecasting, 32, 896-913.

[29] Hong, T., Pinson, P., & Fan, S. (2014). *Global energy forecasting competition 2012*. International Journal of Forecasting, 30(2), 357-363.

[30] Hu, W., Tong, S., Mengersen, K., & Connell, D. (2007). *Weather Variability and the Incidence of Cryptosporidiosis: Comparison of Time Series Poisson Regression and SARIMA Models*. Annals of Epidemiology Volume 17, Issue 9, Pages 679-688.

[31] Hyndman, R., & Athanasopoulos, G. (2013). *Forecasting: principles and practice*. OTexts.

[32] Iversen, E.B., Morales, J.M., Moller, J.K., & Madsen, H., (2016). *Short-term probabilistic forecasting of wind speed using stochastic differential equations*. International Journal of Forecasting, 32, 981-990.

[33] Jacobsson, H. (2005). *Rule extraction from recurrent neural networks: A taxonomy and review*. Neural Computation, 17(6), 1223-1263.

[34] Janczura, J., & Weron, R. (2010). *An empirical comparison of alternate regime-switching models for electricity spot prices*. Energy Economics, 32, 1059-1073.

[35] Janczura, J., Trück, S., Weron, R., & Wolff, R.C. (2013). *Identifying spikes and seasonal components in electricity spot price data: A guide to robust modeling*. Energy Economics, 38, 96-110.

[36] Juban, R., Ohlsson, H., Maasoumya, M., Poirier, L., & Kolter, J.Z. (2016). *A multiple quantile regression approach to the wind, solar, and price tracks of GEFCom2014*. International Journal of Forecasting, 32, 1094-1102.

[37] Konar, A. (2005). *Computational intelligence: principles, techniques and applications*. Springer.

[38] Landry, M., Erlinger, T.P., Patschke, D., & Varrichio, C., (2016). *Probabilistic gradient boosting machines for GEFCom2014 wind forecasting*. International Journal of Forecasting, 32, 1061-1066.

[39] Lin, T.N., Horne, B.G., Tino, P., & Giles, C.L. (1996). *Learning long-term dependencies in NARX recurrent neural networks*. IEEE Transactions on Neural Networks, 7(6), 1329-1337.

[40] Ljung, L. (1999). *System identification - theory for the user*. (2nd ed.). Prentice Hall: Upper Saddle River.

[41] Lorenz, E., Scheidsteger, T., Hurka, J., Heinemann, D., & Kurz, C. (2011). *Regional PV power prediction for improved grid integration*. Progress in Photovoltaics: Research and Applications, 19(7), 757-771.

[42] Maciejowskaa, K., & Nowotarski, J. (2016). *A hybrid model for GEFCom2014 probabilistic electricity price forecasting*. International Journal of Forecasting, 32, 1051-1056.

[43] Mangalova, E., & Shesterneva, O., (2016); *K-nearest neighbors for GEFCom2014 probabilistic wind power forecasting*. International Journal of Forecasting, 32, 1067-1073.

[44] Nelson, M., Hill, M., Remus, T., & O'Connor, W. (1999). *Time series forecasting using neural networks: should the data be deseasonalized first?*. Journal of Forecasting, 18(5), 359-367.

[45] Rodriguez, C.P., & Anders, G.J., (2004). *Energy price forecasting in the Ontario competitive power system market*. IEEE Transactions on Power Systems, 19(1), 366-374.

[46] Shahidehpour, M., Yamin, H., & Li, Z. (2002). *Market operations in electric power systems: forecasting, scheduling, and risk management*. Wiley.

[47] Shumway, R.H., & Stoffer, D. S. (2006). *Time series analysis and its applications*. (2nd ed.). Springer.

[48] Tong, H., & Lim, K.S. (1980). *Threshold autoregression, limit cycles and cyclical data.* Journal of the Royal Statistical Society, Series B, 42, 245-292.

[49] Uniejewski, B., Nowotarski, J., & Weron, R. (2016). *Automated Variable Selection and Shrinkage for Day-Ahead Electricity Price Forecasting.* Energies, 9, 621.

[50] Uniejewski, B., Marcjasz, G., & Weron, R. (2018). *On the importance of the long-term seasonal component in day-ahead electricity price forecasting. Part II, Probabilistic forecasting.* Energy Economics.

[51] Vahvilainen, I., & Pyykkonen, T. (2005). *Stochastic factor model for electricity spot price - the case of the Nordic market.* Energy Economics, 27(2), 351-367.

[52] Vapnik, V. (1995). *The nature of statistical learning theory.* Springer.

[53] Weron, R. (2014). *Electricity price forecasting: A review of the state-of-the-art with a look into the future.* International Journal of Forecasting, 30(4), 1030-1081.

[54] Weron, R. (2006). *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach.* Chichester: Wiley.

[55] Xie, J., & Hong, T. (2016). *GEFCom2014 probabilistic electric load forecasting: An integrated solution with forecast combination and residual simulation.* International Journal of Forecasting, 32, 1012-1016.

[56] Zhang, G.P., & Kline, D.M., (2007). *Quarterly Time Series Forecasting with Neural Networks.* IEEE Transactions on Neural Networks, 18(6), 501-514.

[57] Zhang, G., Patuwo, B.E., & Hu, M.Y. (1998). *Forecasting with artificial neural networks: the state of the art.* International Journal of Forecasting, 14, 35-62.

[58] Ziel, F., & Liu, B., (2016). *Lasso estimation for GEFCom2014 probabilistic electric load forecasting.* International Journal of Forecasting, 32, 1029-1037.