Solar Irradiation Forecasting for PV Systems by Fully Tuned Minimal RBF Neural Networks

Lucio Ciabattoni[1], Gianluca Ippoliti[1], Sauro Longhi[1],
Matteo Pirro[1], and Matteo Cavalletti[2]

[1] Universitá Politecnica delle Marche, Dipartimento di Ingegneria dell'Informazione,
Via Brecce Bianche 12, 60131 Ancona, Italy
{l.ciabattoni,g.ippoliti,s.longhi,m.pirro}@univpm.it
[2] Energy Resources S.p.A.,
Via Ignazio Silone, 60035 Jesi, Italy
m.cavalletti@energyresources.it

Abstract. An on-line prediction algorithm able to estimate, over a determined time horizon, the solar irradiation of a specific site is considered. The learning algorithm is based on Radial Basis Function (RBF) networks and combines the growing criterion and the pruning strategy of the minimal resource allocating network technique. An adaptive extended Kalman filter is used to update all the parameters of the Neural Network (NN). The on-line learning mechanism avoids the initial training of the NN with a large data set. The proposed solution has been experimentally tested on a 14 kWp PhotoVoltaic (PV) plant and results are compared to a classical RBF neural network.

Keywords: irradiation forecasting, minimal resource allocating networks, adaptive filtering, self learning algorithm, neural networks.

1 Introduction

Recently, energy saving and energy security have become major issues, especially in some countries where energy deficiency not only impacts economics, society and development of the country, but also results in the global warming. For these reasons, interest in renewable energy is growing around the world and electric system operators are addressing the challenge of how to integrate significant amounts of wind, solar and other forms of variable generation into electricity grids while ensuring system reliability. This calls for transmission additions and reinforcements, enhanced forecasting and planning techniques for variable generation, and access to flexible grid resources including customer participation in demand management programs, plug-in hybrid electric vehicles and large scale electricity storage to help reliably integrate variable resources to electricity systems. In particular, there are two tasks to integrate variable generation and Distributed Energy Resources (DER), both locally and globally: integrating them into the electricity network and into the energy market. One solution to decrease the problems caused by the variable output of some distributed generation is to add energy

B. Apolloni et al. (Eds.): *Neural Nets and Surroundings*, SIST 19, pp. 289–300.
DOI: 10.1007/978-3-642-35467-0_29

storages into the systems (centralised or distributed energy storages). Forecast information on the expected DER power production plays a primary role for the optimization and management of those storages. Another solution is to use flexibility in electricity consumption [17]. In fact, today time of use tariffs, for domestic use of energy, penalize some periods of time with a higher price. Prosumers (customers and producers of energy at the same time), knowing their forecasted energy production profile can (re)arrange their processes to minimize costs, having great economic benefits. Since both solutions need power production forecasts it's a crucial task the finding of a valid and reliable forecasting scheme. This problem has been deeply investigated in literature, in particular for what regards the forecasting and power scheduling from wind plant [15]. Also the prediction of solar yields is becoming more and more important, especially for countries where legislation encourages the deployment of solar power plants [16]. The irradiation forecasted can be used as input for a PhotoVoltaic (PV) model to obtain the PV production, or used in building energy simulation program that model energy and water use in buildings. Depending on the application and the corresponding time scale different approaches for modeling and forecasting solar irradiation may be appropriate and there have been a lot of researchers engaged in the modeling of solar irradiance. It is possible to use satellite-based cloud motion vectors for cloud fields and, therefore, surface irradiance short-term predictions with a forecast horizon of up to approximately 4 h or irradiance models (clear-day solar radiation, half-sine, Seasonal Auto Regressive Integrated Moving Average (SARIMA), WD-SARIMA, Support Vector Regression, see [9] and reference therein). However, those methods have not been efficient in all cases (most of them are efficient only to forecast up to $5-10$ minutes [14,3]) and contrarily they may yield to noised results. A possible solution is given by Neural Networks (NNs) which provide a nonlinear representation to implement mappings. In particular in this paper Radial Basis Function Networks (RBFNs) have been considered for this prediction and the system dynamics related to the irradiation have been taken into account through the RBFN input pattern that must be composed of a proper set of system input and output samples acquired in a finite set of past time instants [11]. These NNs have been widely used for nonlinear system identification [4,6,8,2,5] because they have the ability both to approximate complex nonlinear mappings directly from input-output data with a simple topological structure that avoid lengthy calculations [6] and to reveal how learning proceeds in an explicit manner. The considered on-line learning algorithm is based on the Minimal Resource Allocating Network (MRAN) technique [22], that adds hidden neurons to the network based on the innovation of each new RBFN input pattern which arrives sequentially. As stated in [22], to obtain a more parsimonious network topology a pruning strategy is introduced. This strategy detects and removes as learning progresses those hidden neurons which make little contribution to the network output. Pruning is necessary for the prediction of the irradiation changing dynamics because inactive hidden neurons could be present as the dynamics which caused their creation becomes nonexistent. If an observation has no novelty then the existing parameters of the network are adjusted by an Extended Kalman Filter (EKF) [22]. In this paper the performance of the filter is improved by an on-line adjustment of the noise statistics obtained by a suitably defined estimation algorithm; the proposed Adaptive Extended Kalman Filter (AEKF) is able to adaptively estimating the unknown statistical parameters [13]. The main advantage of the proposed MRAN algorithm is that a large data set of irradiation measurements, weather forecast, temperature for a specific location is no longer required for the training of the NN, drastically reducing the setup time. Another important advantage is that, due to the adaptive algorithm, some singular seasonal weather situation can be rapidly identified and corrected. A comparison of the performance obtained by the MRAN AEKF RBF Neural Network with respect to the standard RBF Neural Network is presented, considering data taken from a PV plant located in Jesi, Italy. The paper is organized as follows. The on-line prediction algorithm is described in Section 2 and the performance of the considered NNs are discussed in Section 3. The paper ends with comments on the performance of the proposed solution.

2 Prediction Algorithm

The approach to implement a Minimal Resource Allocating Network (MRAN) is based on a sequential learning algorithm and an Extended Kalman Filter (EKF) [22,10]. In particular the sequential learning algorithm adds or removes neurons on-line to the network according to a given criterion [22] and an EKF is used to update the net parameters. In this paper the MRAN algorithm is improved by an Adaptive Extended Kalman Filter (AEKF) in order to take into account the time-varying noise statistics [13], as shown in the following.

2.1 Radial Basis Function Neural Network

A RBFN with input pattern $\boldsymbol{x} \in \mathbb{R}^m$ and a scalar output $\hat{y} \in \mathbb{R}$ implements a mapping $f : \mathbb{R}^m \to \mathbb{R}$ according to

$$\hat{y} = f(\boldsymbol{x}) = \lambda_0 + \sum_{i=1}^{K} \lambda_i \phi\left(\|\boldsymbol{x} - \boldsymbol{c}_i\|\right) \tag{1}$$

where $\phi(\cdot)$ is a given function from $\mathbb{R}^+$ to $\mathbb{R}$, $\|\cdot\|$ denotes the Euclidean norm, λ_i, $i = 0, 1, \cdots, K$ are the weight parameters, $\boldsymbol{c}_i \in \mathbb{R}^m$, $i = 1, 2, \cdots, K$, are the radial basis function centers (called also units or neurons) and K is the number of centers [6]. The terms:

$$o_i = \lambda_i \phi\left(\|\boldsymbol{x} - \boldsymbol{c}_i\|\right), i = 1, \cdots, K \tag{2}$$

are called the hidden unit outputs.

In this paper the RBFN is used for the prediction of the output of a dynamical system and the system dynamics can be taken into account through the network input pattern $\boldsymbol{x}$, that must be composed of a proper set of system input and output samples acquired in a finite set of past time instants [11]. In particular in this paper, the system output is the solar irradiation (see Section 3) and the inputs are the weather forecast, the number of day of the year, the hour of the day (see Section 3).

Theoretical investigation and practical results show that the choice of the non-linearity $\phi(\cdot)$, a function of the distance d_i between the current input $\boldsymbol{x}$ and the centre $\boldsymbol{c}_i$, does not significantly influence the performance of the RBFN [6]. Therefore, the following gaussian function is considered:

$$\phi(d_i) = \exp\left(-d_i^2/\beta_i^2\right), \qquad i = 1,2,\cdots,K \tag{3}$$

where $d_i = \|\boldsymbol{x} - \boldsymbol{c}_i\|$ and the real constant β_i is a scaling or "width" parameter [6].

2.2 Minimal Resource Allocating Network Algorithm

The learning process of MRAN involves allocation of new hidden units and a pruning strategy as well as adaptation of network parameters [22]. The network starts with no hidden units and as input-output data $(\boldsymbol{x}(\cdot), y(\cdot))$ are received, some of them are used to generate new hidden units based on a suitably defined growth criteria. In particular at each time instant n the following three conditions are evaluated to decide if the input $\boldsymbol{x}(n)$ should give rise to a new hidden unit:

$$\|e(n)\| = \|y(n) - f(\boldsymbol{x}(n))\| > E_1 \tag{4}$$

$$e_{rms}(n) = \sqrt{\sum_{j=n-(M-1)}^{n} \frac{e(j)^2}{M}} > E_2 \tag{5}$$

$$d(n) = \|\boldsymbol{x}(n) - \boldsymbol{c}_r(n)\| > E_3 \tag{6}$$

where $\boldsymbol{c}_r(n)$ is the centre of the hidden unit that is nearest to $\boldsymbol{x}(n)$ and M represents the number of past network outputs to calculate the output error $e_{rms}(n)$. The terms E_1, E_2 and E_3 are thresholds to be suitably selected. As stated in [22], these three conditions evaluate the novelty in the data. If all the criteria of (4)–(6) are satisfied, a new hidden unit is added and the following parameters are associated with it:

$$\lambda_{K+1} = e(n) \tag{7}$$

$$\boldsymbol{c}_{K+1} = \boldsymbol{x}(n) \tag{8}$$

$$\beta_{K+1} = \alpha \|\boldsymbol{x}(n) - \boldsymbol{c}_r(n)\| \tag{9}$$

where α determines the overlap of the response of a hidden unit in the input space as specified in [22]. If the observation $(\boldsymbol{x}(n), y(n))$ does not satisfy the criteria of (4)–(6), an EKF is used to update the following parameters of the network:

$$\boldsymbol{w} = \left[\lambda_0, \lambda_1, \boldsymbol{c}_1^T, \beta_1, \cdots, \lambda_N, \boldsymbol{c}_N^T, \beta_N\right]^T. \tag{10}$$

The update equation is given by:

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \boldsymbol{k}(n)e(n) \tag{11}$$

where the gain vector $\boldsymbol{k}(n)$ is expressed by:

$$\boldsymbol{k}(n) = \boldsymbol{P}(n-1)\boldsymbol{a}(n)\left[\sigma_v^2(n) + \boldsymbol{a}^T(n)\boldsymbol{P}(n-1)\boldsymbol{a}(n)\right]^{-1} \tag{12}$$

with $\boldsymbol{a}(n)$ the gradient vector of the function $f(\boldsymbol{x}(n))$ (see Eq. 1) with respect to the parameter vector $\boldsymbol{w}(n-1)$ [22], $\sigma_v^2(n)$ is the variance of the measurement noise and $\boldsymbol{P}(n-1)$ is the error covariance matrix. This matrix is updated by:

$$\boldsymbol{P}(n) = \left[I_{z\times z} - \boldsymbol{k}(n)\boldsymbol{a}^T(n)\right]\boldsymbol{P}(n-1) + \sigma_\eta^2(n-1)\boldsymbol{I}_{z\times z} \tag{13}$$

where $\boldsymbol{I}$ is the identity matrix and $\sigma_\eta^2(n-1)$ is introduced to avoid that the rapid convergence of the EKF algorithm prevents the model from adapting to future data [22]. The $z \times z$ matrix $\boldsymbol{P}(n)$ is positive definite symmetric and z is the number of parameters to be adjusted. When a new hidden neuron is allocated, the dimension of $\boldsymbol{P}(n)$ increases to:

$$\boldsymbol{P}(n) = \begin{pmatrix} \boldsymbol{P}(n-1) & 0 \\ 0 & p_0 \boldsymbol{I}_{z_1\times z_1} \end{pmatrix}. \tag{14}$$

In (14), p_0 is an estimate of the uncertainty in the initial values assigned to the parameters and the dimension z_1 of the identity matrix $\boldsymbol{I}$ is the number of new parameters introduced by adding the new hidden neuron. As stated in [22], to keep the RBF network in a minimal size a pruning strategy removes those hidden units that contribute little to the overall network output over a number of consecutive observations. To carry out this pruning strategy, for every observation $(\boldsymbol{x}(n), y(n))$ the hidden unit outputs are computed (see Eq. (2)) and normalized with respect to the highest output:

$$\overline{o}_i(n) = \frac{o_i(n)}{\max\{o_i(n)\}}, i = 1, \cdots, K. \tag{15}$$

The hidden units for which the normalized output (15) is less than a threshold δ for ξ consecutive observations are removed and the dimensionality of all the related matrices are adjusted to suit the reduced network [22].

The weakness of the above algorithm is that all the parameters of the network, including all the centers of the hidden neurons, widths and weights, have to be update at every step; the size of the matrices to be update becomes large as the number of hidden neurons increases. Therefore, for the real-time implementation of the considered algorithm, it could be necessary to reduce the online computation effort and to this purpose a "winner neuron" strategy can be incorporate in the learning algorithm as proposed in [22]. The "winner neuron" is defined as the neuron in the network that is closest (in some norm sense) to the current input data. The criteria for adding and pruning the hidden neurons are all the same as in the above algorithm; the difference, in the "winner neuron" strategy, is that if the observation $(\boldsymbol{x}(\cdot), y(\cdot))$ does not meet the criteria to add a new hidden neuron (see Eqs. (4)–(6)), only the network parameters related to the selected "winner neuron" are updated by the EKF algorithm [22].

The EKF can be implemented once estimates of $\sigma_\eta^2(n)$ and $\sigma_v^2(n)$ are available. In general, a complete and reliable information about these estimates is not available; on the other hand it is well known how poor knowledge of noise statistics may seriously degrade the Kalman filter performance. This problem is here dealt with introducing an adaptive adjustment mechanism of $\sigma_\eta^2(n)$ and $\sigma_v^2(n)$ values in the EKF equations.

2.3 Adaptive Estimation of $\sigma_\eta^2(n)$ and $\sigma_v^2(n)$

A considerable amount of research has been performed on the adaptive Kalman filtering [13,23], but in practice it is often necessary to redesign the adaptive filtering scheme according to the particular characteristics of the faced problem. In the adaptive procedure here proposed, under proper assumptions given in [13], it is possible to define a simple and efficient estimation algorithm based on the condition of consistency, at each step, between the residual $e(n)$ and its predicted statistic $E\{e^2(n)\}$. Imposing such a condition, one-stage estimates $\hat{\sigma}_\eta^2(n-1)$ and $\hat{\sigma}_v^2(n)$, of $\sigma_\eta^2(n-1)$ and $\sigma_v^2(n)$, respectively, are obtained at each step. To increase their significance, the one-stage estimates $\hat{\sigma}_\eta^2(n-1)$ and $\hat{\sigma}_v^2(n)$ are average obtaining the relative smoothed version $\bar{\hat{\sigma}}_\eta^2(n-1)$ and $\bar{\hat{\sigma}}_v^2(n)$. After proper calculations [13], the following recursive form of estimates $\bar{\hat{\sigma}}_\eta^2(n-1)$ and $\bar{\hat{\sigma}}_v^2(n)$ is found:

$$\bar{\hat{\sigma}}_\eta^2(n-1) = \bar{\hat{\sigma}}_\eta^2(n-2) + \frac{1}{l_\eta + 1}[\hat{\sigma}_\eta^2(n-2) - \hat{\sigma}_\eta^2(n-(l_\eta+1))] \tag{16}$$

$$\bar{\hat{\sigma}}_v^2(n) = \bar{\hat{\sigma}}_v^2(n-1) + \frac{1}{l_v + 1}(\hat{\sigma}_v^2(n) - \hat{\sigma}_v^2(n-l_v)) \tag{17}$$

where:

- $\hat{\sigma}_\eta^2(n-1) = \max\{(\boldsymbol{a}(n)\boldsymbol{a}^T(n))^{-1}[e(n)^2 - \boldsymbol{a}(n)\boldsymbol{P}(n-1)\boldsymbol{a}^T(n) - \bar{\hat{\sigma}}_v^2(n)], 0\}$
- $\hat{\sigma}_v^2(n) = \max\{e^2(n) - [\boldsymbol{a}(n)\boldsymbol{P}(n-1)\boldsymbol{a}^T(n) + \boldsymbol{a}(n)\bar{\hat{\sigma}}_\eta^2(n-1)\boldsymbol{I}\boldsymbol{a}^T(n)], 0\}$
- l_η and l_v are the number of one-stage estimates $\bar{\hat{\sigma}}_\eta^2(n-1)$ and $\bar{\hat{\sigma}}_v^2(n)$ respectively, yielding the smoothed estimates.

Parameters l_η and l_v of estimators (16) and (17) are chosen on the basis of two antagonist considerations: low values would produce noise estimators which are not statistically significant, large values would produce estimators which are scarcely sensitive to possible rapid fluctuations of the true $\sigma_\eta^2(n-1)$ and $\sigma_v^2(n)$ [13]. In other words, the one-stage estimates are made by averaging past samples in order to increase the statistical significance of estimators; if the samples are too far the filter has a low reactivity, while if the samples are too near estimators have a low statistical significance [13]. During filter initialization, the starting values $\hat{\sigma}_\eta^2(0)$ and $\hat{\sigma}_v^2(0)$ in Eqs. (16) and (17) respectively, must be chosen on the basis of the *a priori* available information. In case of lack of such information, a large value of $P(0,0)$ is useful to prevent divergence.

The MRAN prediction algorithm [22] enhanced by the AEKF, called MRANAEKF algorithm, is summarized as follow:

1. For each observation $(\boldsymbol{x}(n), y(n))$ do: compute the overall network output: $\hat{y}(n) = f(\boldsymbol{x}(n)) = \lambda_0 + \sum_{i=1}^{K} \lambda_i \phi\left(\|\boldsymbol{x}(n) - \boldsymbol{c}_i\|\right)$ where K is the number of hidden units;
2. Calculate the parameters required by the growth criterion:
   - $\|e(n)\| = \|y(n) - f(\boldsymbol{x}(n))\|$
   - $e_{rms}(n) = \sqrt{\sum_{j=n-(M-1)}^{n} \frac{e(j)^2}{M}}$

```
      - $d(n) = \|\boldsymbol{x}(n) - \boldsymbol{c}_r(n)\|$
3. Apply the criterion for adding a new hidden unit:
   if
   $\|e(n)\| > E_1$ and $e_{rms}(n) > E_2$ and $d(n) > E_3$ allocate a new hidden unit $K+1$ with:
      - $\lambda_{K+1} = e(n)$
      - $\boldsymbol{c}_{K+1} = \boldsymbol{x}(n)$
      - $\beta_{K+1} = \alpha \|\boldsymbol{x}(n) - \boldsymbol{c}_r(n)\|$
   else
      - adapt the measurement noise coefficient:
        $\bar{\hat{\sigma}}_v^2(n) = \bar{\hat{\sigma}}_v^2(n-1) + \frac{1}{l_v+1}(\hat{\sigma}_v^2(n) - \hat{\sigma}_v^2(n-l_v))$
      - tune the network parameters:
        $\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \boldsymbol{k}(n)e(n)$
      - adapt the process noise coefficient:
        $\bar{\hat{\sigma}}_\eta^2(n-1) = \bar{\hat{\sigma}}_\eta^2(n-2) + \frac{1}{l_\eta+1}[\hat{\sigma}_\eta^2(n-2) - \hat{\sigma}_\eta^2(n-(l_\eta+1))]$
      - update the error covariance matrix:
        $\boldsymbol{P}(n) = \left[I_{z\times z} - \boldsymbol{k}(n)\boldsymbol{a}^T(n)\right]\boldsymbol{P}(n-1) + \sigma_\eta^2(n-1)\boldsymbol{I}_{z\times z}$
   end
4. Check the criterion to prune hidden units:
      - compute the hidden unit outputs:
        $o_i(n) = \lambda_i \phi\left(\|\boldsymbol{x}(n) - \boldsymbol{c}_i\|\right), i = 1, \cdots, K$
      - compute the normalized outputs:
        $\overline{o}_i(n) = \frac{o_i(n)}{\max\{o_i(n)\}}, i = 1, \cdots, K$
      - if $\overline{o}_i(\cdot) < \delta$ for $\xi$ consecutive observations then
        prune the $ith$ hidden unit and reduce the dimensionality of the related matrices
   end
5. $n = n+1$ and go to step 1.
```

3 Neural Network Based Irradiation Forecasting

Tests are based on data acquired from January 2011 to December 2011 during PV plant standard working. The considered 14 kWp PV plant, equipped with polysilicon solar panels south oriented and tilt angle 27 deg., is located in Jesi (AN), Italy. It is composed by 8 strings of Renergies $220P/220$ polysilicon panels [19] where each string is connected to a SMA Sunny Boy $1700IT$ solar inverter [21]. A lithium battery pack which is composed by the series of two sub-module with 80 ThunderSky modules 40 Ah, a Battery Management System (BMS) and a battery charger for each module [24]. A solar inverter (model SIAC soleil 10kW) is connected to this pack [20]. All communication is done through the TCP/IP protocol, using serial to TCP/IP converters. All devices are connected to a server, where it is located the software to manage the whole system.

For the above PV plant, two different situations are considered: in the first one the proposed algorithm is tested without a pre-trained net; this situation can occur if the solar irradiation has to be predicted on a plant without previous information on it. In the second situation, the MRANAEKF learning algorithm starts with a pre-trained net

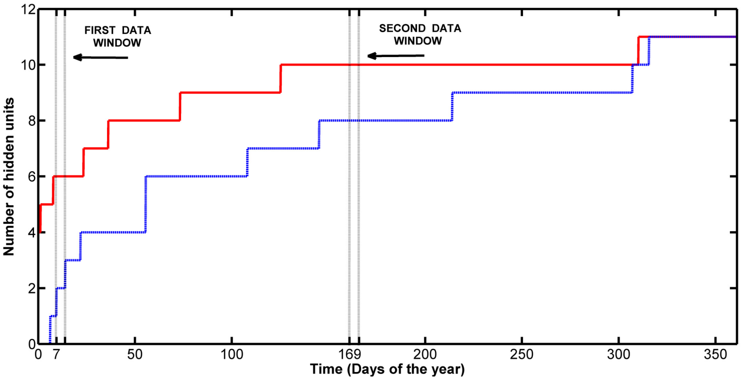

Fig. 1. Evolution of hidden neurons due to growing and pruning for the considered data set (Year 2011). The continuous red line is the pre-trained MRANAEKF network while the dashed blue line is the no pre-trained MRANAEKF network. The first data window is from day 7 ($144th$ sample) to day 9 ($216th$ sample); the second data window is from day 169 ($4032th$ sample) to day 171 ($4106th$ sample).

based on few historical information found on the WEB such as solar irradiation profile of clear sky days and cloudy days for the specified location, panel orientation and tilting. The information have been taken directly from the website of PVGIS [18]. This is a common operating condition, when no sensors and measures are available for the plant before the forecast starts. To measure the performance of the proposed algorithm, the Root Mean Square of the Error $e(\cdot)$ (RMSE) and its Standard Deviation (SD) have been calculated. Only hours with daylight (irradiance greater than zero) are considered for the calculation of the RMSE; night values with no irradiance are excluded from the evaluation. The set of experimental data is composed of 8000 pairs of input and output samples. Sampling time is 1 h and the data have been normalized, between 0 and 1, in order to have the same range. In particular the set of experimental data is given by the pairs $(\boldsymbol{x}(n), y(n))$, $n = 1, 2, \ldots$ where $\boldsymbol{x}(n)$ is composed by the hourly weather forecast given by sunny, clean, partly cloud and overcast sky conditions taken on the WEB [12,1], the number of day of the year, the hour of the day, the air temperature and $y(\cdot)$ is the solar irradiation measured by the SMA Sunny Sensorbox [21]. The considered algorithm requires careful selection of the threshold parameters E_1, E_2, E_3 and of parameter M, as defined in Eqs. (4)–(6), which control the growth characteristics of the network; i.e. if small thresholds are chosen more units are added to the NN. The parameters δ and ξ control the pruning strategy (Eq. (15)); it is important to take into account the system non stationarity to select these parameters. In other words, slowly dynamic variations imply a bigger δ and a smaller ξ. The parameters α, p_0, $\sigma_v^2(0)$ and $\sigma_\eta^2(0)$ related to the AEKF algorithm used to update the network parameters of Eq. (10) are chosen by trial and error. In the considered experimental tests the numeric values of these parameters are selected, as: $E_1 = 0.01$, $E_2 = 0.02$, $E_3 = 0.4$, $M = 50$, $\delta = 0.0005$, $\xi = 2000$, $\alpha = 1.2$, $p_0 = 0.2$, $\sigma_v^2(0) = 0.03$, $\sigma_\eta^2(0) = 0.03$. Samples of the performed prediction tests are given in Fig. 1 through 4. In particular the Fig. 1

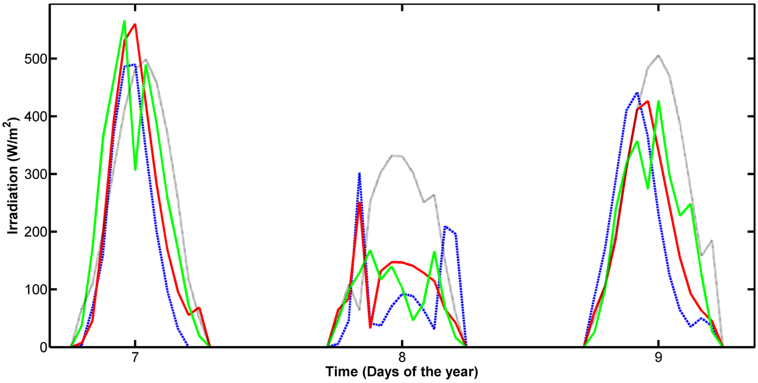

Fig. 2. First data window: day 7 to 9 of 2011 (7–9 January)

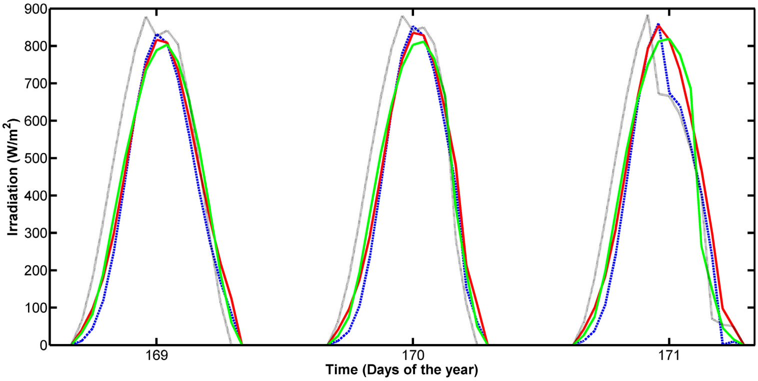

Fig. 3. Second data window: day 169 to 171 of 2011 (19–21 June)

shows the hidden neurons evolution history for the MRANAEKF algorithm as it learns sequentially from data. In this figure two data windows have been highlighted to compare the performance of the MRANAEKF RBF NN with respect to a classical RBF NN algorithm. As will be showed in Figs. 2 and 3, the first data window is relative to the starting learning period of the MRANAEKF (days 7 – 9) and the second data window is relative to the "steady state" operation (days 169 – 171). Results are compared with a previously trained classical RBF network with the same inputs and output. The training data set of the classical RBF network is composed by 5000 pairs of input-output and it is relative to the last 20 days of each month of 2010 as described in [7]. A sample of the performed tests is shown in Figs. 2 and 3, where continuous red line represents the irradiation predicted by the pre-trained MRANAEKF, the dashed blue line is the irradiation predicted by the no pre-trained MRANAEKF network, the dotted black line is the irradiation predicted by the classical RBF NN and the continuous green line is the measured irradiation. Results for the data set relative to the year 2011 have been

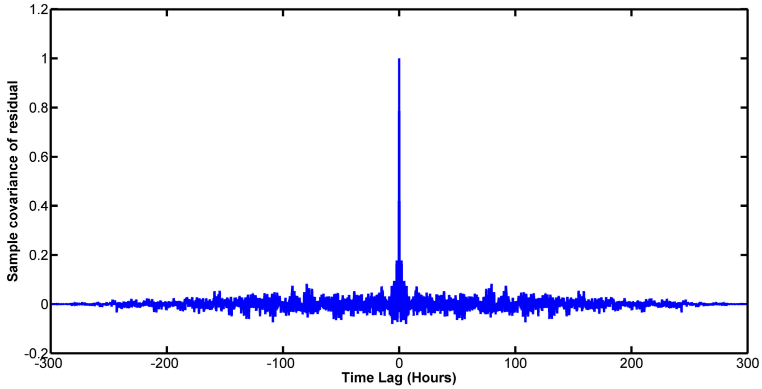

Fig. 4. Sample covariance of the residuals obtained by the prediction performed by the pre-trained MRANAEKF network.

Table 1. Comparison of prediction results (values are in $[W/m^2]$) of the MRANAEKF with (a) and without (b) pre-training and the classical fixed RBF network. The considered whole data set is relative to the year 2011. Data windows are highlighted in Fig. 1.

	MRANAEKF (a)		MRANAEKF (b)		RBF	
DATA	RMSE	SD	RMSE	SD	RMSE	SD
Whole data set	65.2	60.2	70.5	68.2	75.1	74.2
First data window	71.1	69.8	76.2	75.2	81.2	79.7
Second data window	50.4	48.3	55.9	53.4	59.3	57.1

summarized in Table 1. The network pre-trained with basic information on PV plant design and historical data available on the WEB [18] has shown good capability to generalize the prediction of the irradiation of a specific plant in the first two days of working and, with the growing on the net, the forecast becomes more accurate; the net starting without neurons (no pre-trained) needs up to 4 days to show a good performance. Both networks have shown better performance with respect to the classical RBF NN. The whiteness test on the sample covariance of the prediction errors has been used for the network validation. A sample is shown in Fig. 4.

4 Concluding Remarks

In this paper a minimal resource allocating network algorithm has been analyzed for on-line hourly site-specific irradiance forecast. This algorithm increases the number of RBFN hidden neurons depending on the input-output data and an adaptive extended Kalman filter is used to update all the parameters of the network. A pruning strategy is also considered to remove those hidden units which end up to give a contribution to the network output. Two ways of use the MRANAEKF algorithm have been proposed:

with a pre-trained net based only on historical information found on the WEB and with a no pre-trained network. The results indicate that both situations give networks with better prediction accuracy with respect to a classical fixed RBF NN previously trained with a large data set. As future developments, the authors are actually considering the possibility to integrate a “winner neuron” strategy to minimize the computational effort of the overall algorithmic architecture. This will make the developed system even more appealing from a real-time implementation perspective.

References

1. http://www.3bmeteo.com (2012)
2. Antonini, P., Ippoliti, G., Longhi, S.: Learning control of mobile robots using a multiprocessor system. Control Engineering Practice 14(11), 1279–1295 (2006)
3. Ben Salah, C., Ben Mabrouk, A., Ouali, M.: Wavelet autoregressive forecasting of climatic parameters for photovoltaic systems. In: 2011 8th International Multi-Conference on Systems, Signals and Devices (SSD), pp. 1–6 (March 2011)
4. Cavalletti, M., Ippoliti, G., Longhi, S.: Lyapunov-based switching control using neural networks for a remotely operated vehicle. International Journal of Control 80(7), 1077–1091 (2007)
5. Cavalletti, M., Ippoliti, G., Longhi, S.: Intelligent control for a remotely operated vehicle. International Journal of Systems Science 40(11), 1099–1114 (2009)
6. Chen, S., Cowan, C., Grant, P.: Orthogonal least squares learning algorithm for radial basis function networks. IEEE Transactions on Neural Networks 2(2), 302–309 (1991)
7. Ciabattoni, L., Ippoliti, G., Longhi, S., Cavalletti, M., Rocchetti, M.: Solar irradiation forecasting using RBF networks for PV systems with storage. In: 2012 IEEE International Conference on Industrial Technology (ICIT), pp. 699–704 (March 2012)
8. Corradini, M., Ippoliti, G., Longhi, S., Marchei, D., Orlando, G.: A quasi-sliding mode observer-based controller for PMSM drives. Asian Journal of Control, 1–8 (2012), http://dx.doi.org/10.1002/asjc.555
9. Dorvlo, A.S., Jervase, J.A., Al-Lawati, A.: Solar radiation estimation using artificial neural networks. Applied Energy 71(4), 307–319 (2002)
10. Giantomassi, A., Ippoliti, G., Longhi, S., Bertini, I., Pizzuti, S.: On-line steam production prediction for a municipal solid waste incinerator by fully tuned minimal RBF neural networks. Journal of Process Control 21(1), 164–172 (2011)
11. Hunt, K., Sbarbaro, D., Zbikowski, R., Gawthrop, P.: Neural networks for control systems – A survey. Automatica 28(6), 1083–1112 (1992)
12. http://www.ilmeteo.it (2012)
13. Jetto, L., Longhi, S., Venturini, G.: Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots. IEEE Transactions on Robotics and Automation 15(2), 219–229 (1999)
14. Karner, O.: ARIMA representation for daily solar irradiance and surface air temperature time series. Journal of Atmospheric and Solar-Terrestrial Physics 71(8-9), 841–847 (2009)
15. Lange, M., Focken, U.: Physical approach to short term wind power prediction. Springer, New-York (2006)
16. Lorenz, E., Hurka, J., Heinemann, D., Beyer, H.: Irradiance forecasting for the power prediction of grid-connected photovoltaic systems. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 2(1), 2–10 (2009)
17. Palensky, P., Dietrich, D.: Demand side management: Demand response, intelligent energy systems, and smart loads. IEEE Transactions on Industrial Informatics 7(3), 381–388 (2011)

18. http://re.jrc.ec.europa.eu/pvgis (2012)
19. http://www.renergiesitalia.it (2012)
20. http://www.sielups.com (2012)
21. http://www.sma-italia.com (2012)
22. Sundararajan, N., Saratchandran, P., Li, Y.: Fully tuned radial basis function neural networks for flight control. Kluver Academic, London (2002)
23. Szabat, K., Orlowska-Kowalska, T.: Performance improvement of industrial drives with mechanical elasticity using nonlinear adaptive Kalman filter. IEEE Transactions on Industrial Electronics 55(3), 1075–1084 (2008)
24. http://en.winston-battery.com (2011)