Hourly solar radiation forecasting using optimal coefficient 2-D linear filters and feed-forward neural networks

Article *in* Solar Energy · August 2008

DOI: 10.1016/j.solener.2008.02.003

Omer N. Gerek
Anadolu University
169 PUBLICATIONS **1,491** CITATIONS

SEE PROFILE

Mehmet Kurban
Bilecik Üniversitesi
80 PUBLICATIONS **626** CITATIONS

SEE PROFILE

Available online at www.sciencedirect.com

ScienceDirect

Solar Energy 82 (2008) 714–726

SOLAR ENERGY

www.elsevier.com/locate/solener

Hourly solar radiation forecasting using optimal coefficient 2-D linear filters and feed-forward neural networks

Fatih O. Hocaoğlu [*], Ömer N. Gerek, Mehmet Kurban

Anadolu University, Department of Electrical and Electronics Engineering, 26555 Eskişehir, Turkey

Received 27 April 2007; received in revised form 15 December 2007; accepted 7 February 2008
Available online 4 March 2008

Communicated by: Associate Editor Frank Vignola

Abstract

In this work, the hourly solar radiation data collected during the period August 1, 2005–July 30, 2006 from the solar observation station in Iki Eylul campus area of Eskisehir region are studied. A two-dimensional (2-D) representation model of the hourly solar radiation data is proposed. The model provides a unique and compact visualization of the data for inspection, and enables accurate forecasting using image processing methods. Using the hourly solar radiation data mentioned above, the image model is formed in raster scan form with rows and columns corresponding to days and hours, respectively. Logically, the between-day correlations along the same hour segment provide the vertical correlations of the image, which is not available in the regular 1-D representation. To test the forecasting efficiency of the model, nine different linear filters with various filter-tap configurations are optimized and tested. The results provide the necessary correlation model and prediction directions for obtaining the optimum prediction template for forecasting. Next, the 2-D forecasting performance is tested through feed-forward neural networks (NN) using the same data. The optimal linear filters and NN models are compared in the sense of root mean square error (RMSE). It is observed that the 2-D model has pronounced advantages over the 1-D representation for both linear and NN prediction methods. Due to the capability of depicting the nonlinear behavior of the input data, the NN models are found to achieve better forecasting results than linear prediction filters in both 1-D and 2-D.

Keywords: Solar radiation; Forecasting; Linear filter; NN

1. Introduction

Hourly solar radiation data forecasting has significant consequences in most solar applications such as energy system sizing and meteorological estimation. Accurate forecasting improves the efficiency of the outputs of these applications. Classically, the solar radiation data can be regarded as a random time series produced by a stochastic process, and its prediction depends on accurate mathematical modeling of the underlying stochastic process. Using an accurate model, the prediction is mathematically the conditional expectation of the data given the model and the past data samples. On the other hand, the computation of such conditional expectation, which is in general nonlinear, requires the knowledge of the distribution of the samples including higher order statistics. Since the available or recorded data is finite, such distributions can be estimated or fit into pre-set stochastic models such as Auto-Regressive (AR) (Maafi and Adane, 1989), Auto-Regressive Moving-Average (ARMA) (Mellit et al., 2005), Markov (Amato et al., 1986; Aguiar et al., 1988), or learning adaptive systems such as Neural Networks (NNs) (Cao and Cao, 2006). NNs can be trained to predict results from available examples, and they are also able to deal with nonlinear problems.

The determination and optimization of mathematical model parameters is normally done by training algorithms.

[*] Corresponding author. Tel.: +90 222 321 3550; fax: +90 222 323 9501.
E-mail address: fohocaoglu@anadolu.edu.tr (F.O. Hocaoğlu).

0038-092X/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.solener.2008.02.003

Once the training is complete, the predictor can be settled to a fixed function for further prediction or forecasting. In recent years, a number of researchers have used NNs for prediction of hourly global solar radiation data (Sfetsos and Coonick, 2000, Cao and Cao, 2006.). There are several studies about modeling the solar radiation data in the literature (Chena et al., 2007; Cucumo et al., 2007., Kaplanis, 2006; Aguiar and Collares-Pereira, 1992; Kaplanis and Kaplani, 2007). In these works, the data is treated in its raw form as a 1-D time series therefore the inter-day dependencies are not exploited.

This paper presents a novel and low-complexity approach for hourly solar radiation forecasting. The approach is based on a new representation which renders the data in a matrix to form a 2-D image-like model, as explained in Section 2. Although 2-D rendering of a time-series data was already used for some other types of 1-D signals, this kind of an approach is novel in the area of solar radiation signal processing. In fact, the 2-D representation of solar radiation data was first introduced in (Hocaoğlu et al., 2007) using a limited number of prediction methods. In this work, a comprehensive and comparative study is carried out to test the 2-D representation efficiency with several linear and neural prediction filters. The single direction (1-D) and multiple direction (2-D) prediction templates are tested, compared, and optimized to find the optimal template for the representation.

The proposed 2-D representation clearly enables the visualization and comprehension of seasonal inter-day dependencies along the same hour segments of the days. This is due to the fact that the column elements of consecutive days corresponding to the same hour of the day are located as vertically neighbor in 2-D data. Signal compression theory indicates that locating relatively correlated samples in geometrically near positions in a representation greatly improves predictability of the data. This intuitive observation is also justified in the case of solar radiation data by the conducted experiments, which yield the fact that full 2-D prediction templates provide better prediction compared to 1-D prediction. Mathematically, the stochastic part of the solar radiation data is due to transient atmospheric phenomena. These non-predictable variations are often secondary to the oscillatory variations determined by solar geometry which are entirely predictable as a function of latitude, date, and time. The latitude of a geometrical place stays constant, leaving the varying parameters as date and time. The idea of 2-D rendering is motivated by the fact that there is a pronounced periodicity of the solar radiation data with periods of exactly 1 day. If the data is presented in 1-D form along an axis of hours, the daily oscillations and date-wise (seasonal) oscillations become occluded and the data get difficult to interpret. It is illustrated here that splitting the date and time variables into two separate axes better exploits the predictable sinusoidal behavior along both time and date parameters.

After describing the 2-D rendering method in Section 2, the first experiments conducted to test the 2-D model efficiency utilized optimal linear image prediction filters (Gonzalez and Woods, 2002), as explained in Section 3. In order to take into account the adaptive nature of complex (due to non-predictable fluctuations) and non-stationary (oscillatory) time series, neural networks are also applied to the forecasting problem in Section 4. The training algorithms for NNs are also discussed in this Section. In Section 5, performance evaluation methods for predictors are described and the numerical forecasting results that are obtained from both optimal linear filters and neural network models are presented in Section 6.

Throughout this work, our own solar recordings were used for the implementation of the proposed methods. This was possible due to the wind and solar observation station established at Iki Eylul campus of Anadolu University, Turkey, in order to determine the wind and solar energy potentials of the region. The data collected in this observation station between the dates of July 1, 2005 and September 30, 2006 were evaluated via the CALLaLOG 98 software and by algorithms implemented in MATLAB. The data is saved in a data logger at 1 h time intervals (Kurban and Hocaoglu, 2006). The continuity of data acquisition was constantly monitored in our labs. Therefore, the quality of the data was assured. The accuracy of the data was also cross-checked by the data obtained from the National Meteorological Station values for Eskisehir region.

2. A novel representation for the solar radiation data

In this section, the novel 2-D representation (Hocaoğlu et al., 2007) is explained. In order to visualize the efficiency of our new representation, one year data (Agust 1, 2005-July 30, 2006) data are first considered as one-dimensional time series, and then as a 2-D image signal. In order to observe one year behavior of radiations, the data samples are reordered to span from January to December, and then plotted in Fig. 1. Next, the same data are rendered as a two-dimensional matrix (Eq. (1)) and plotted as a surface mesh in Fig. 2, where the dimensions x, y, z, are hours of days, days of year, and solar radiation magnitudes, respectively. In Fig. 1, it is visually very difficult to grasp which parts correspond to day-time hours, which parts correspond to night hours, and what the seasonal behaviors are

$$\mathbf{X}_{\mathrm{rad}} = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} \tag{1}$$

The 2-D representation provides significant insight about the radiation pattern as a function of time. The informational insight is apparent from the sample surface-plots and image visualizations (in gray-scale) presented in Figs. 2 and 3. These two plots correspond to the same rendering style with different visualizations. Both of the visualizations provide the mentioned intuitive infor-

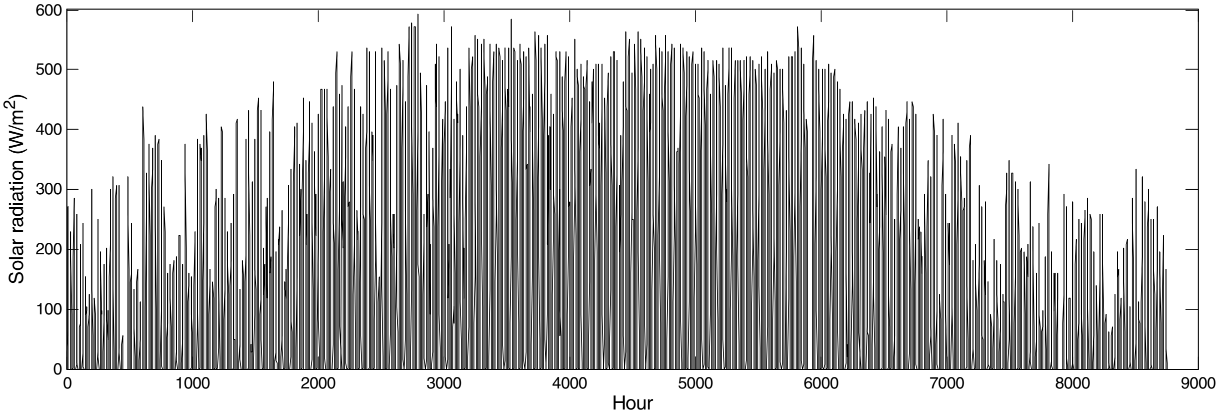

Fig. 1. Hourly solar radiation data in 1-D time-plot.

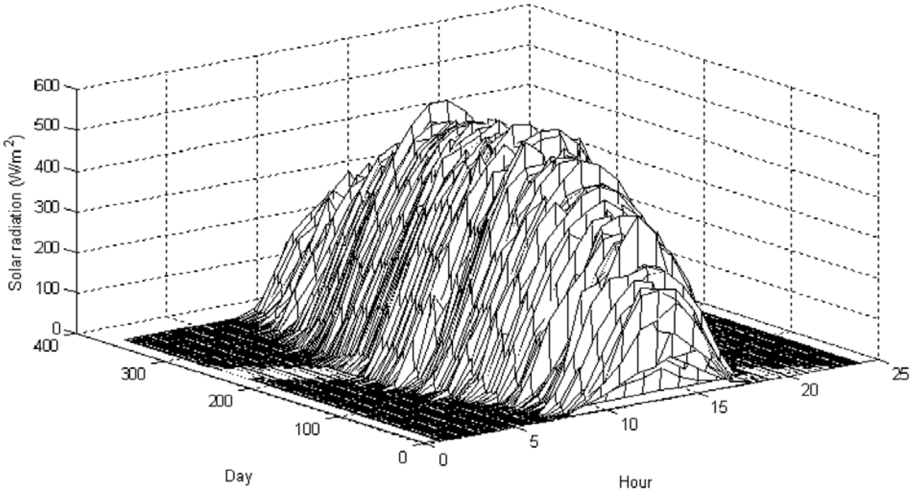

Fig. 2. Hourly solar radiation data presented in 2-D surface form with "days" and "hours" axes.

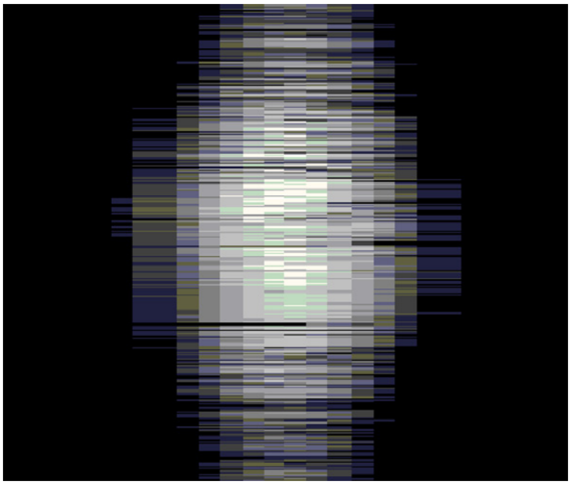

Fig. 3. Image view of solar radiation data rendered in gray-scale 2-D form.

mation. The 2-D gray-scale "image" interpretation can be processed further using image processing techniques, and structural and pattern information can be extracted.

By inspecting the image version of the data in Fig. 3, it is easy to interpret daily and seasonal behavior of solar radiation. Dark regions of the image indicate that there is no sun shine on the horizontal surface. The transition between black and white indicates that solar radiation falling on the horizontal surface is increasing or decreasing. During winter time, the dawn to dusk period is shorter, producing a narrower protruding blob. Conversely, the white blob is wider during summer times, indicating that the day-time is longer. Hence, the width behavior of the white blob indicates the seasonal changes of sun-light periods.

In this representation, the horizontal and vertical correlation within the 2-D data is quite pronounced. This implies that, given the vertical correlation among the same hours of consecutive days, it is beneficial to use 2-D prediction for hourly forecasting.

The resulting 2-D shape also gives rough information about the extraterrestrial solar radiation. The terrestrial solar radiation is, in fact, the extraterrestrial radiation pattern that passes through the nonlinear and time-varying effects of the atmosphere and then added by non-predictable (random) atmospheric phenomena such as clouds.

Therefore, the constructed model provides more accurate information regarding the actual terrestrial solar radiation due to the consideration of the "observed" data that includes meteorological effects such as clouds, fog, etc. The difference between the proposed prediction methods and the actual recordings are observed to be lower than the solar radiation data normalized by the extraterrestrial solar radiation values. Furthermore, such normalization of the data produced samples with very low correlation, and this caused all available statistical prediction methods to fail. As a result, extraterrestrial solar radiation normalization is avoided.

In this work, the prediction efficiency of the proposed model is illustrated with 2-D optimum linear prediction filters and neural networks.

3. Optimal 2-D linear prediction filter design

The proposed 2-D representation has significant advantages which makes prediction and forecasting more robust and accurate. In Hocaoğlu et al. (2007), a preliminary set of linear prediction filters were tested with the 2-D representation. In this work, the experiments are extended to evaluate and compare linear prediction filters with different template sizes and structures. Throughout this section, we first construct the basis of 2-D linear prediction. Due to predictive image coding literature, it is known that a single or multi-dimensional time series can be efficiently modeled by linear predictive filters. The prediction domain is determined by the dimension of the time series. As an example, consider a 2-D three coefficient prediction template structure as given in expression 2

$x_{i,j}$	$x_{i,j+1}$
$x_{i+1,j}$	$\hat{x}_{i+1,\,j+1} = ?$

(2)

where the prediction value is determined for the lower-right pixel value using the "past" pixels as

$$\hat{x}_{i+1,j+1} = x_{i,j} \cdot a_1 + x_{i,j+1} \cdot a_2 + x_{i+1,j} \cdot a_3 \tag{3}$$

The template structure slides over the 2-D data in raster-scan, i.e., the j parameter slides horizontally for each row, and the row count i slides vertically. In this study, i indicate the days, j indicates the hours, and $x_{i,j}$ indicates the solar radiation at ith day and jth hour. At each location during the mentioned sliding operation (corresponding to different i and j), the prediction is made for the coordinate (i,j) using data from coordinates corresponding to past samples. The term "linear prediction" comes from the fact that the prediction result is obtained as a linear function of the input past data samples. The alternative to linear prediction can be considered as construction of a nonlinear predictor that takes the input samples and evaluates a nonlinear function of the input as $\hat{x}_{i+1,j+1} = f_{\mathrm{nonlinear}}(x_{i,j}, x_{i,j+1}, x_{i+1,j})$. Neural Networks are inherently nonlinear systems that are known to perform well for estimation. Therefore, the results of the linear predictors are compared to those of Neural Networks (NNs) in this work. For both linear prediction systems and NNs, the prediction functions are determined according to training phases that optimize either the linear coefficients (a's) or the nonlinear function $f_{\mathrm{nonlinear}}()$ to make the difference between the prediction value $\hat{x}_{i+1,j+1}$ and the actual recorded value $x_{i+1,j+1}$ small.

In 2-D linear prediction, each past sample within the sliding template is scaled by an appropriate coefficient that is optimized according to minimization of the squared prediction error. Therefore, the prediction $\hat{x}_{i+1,j+1}$ is constructed as a linear combination of past samples, and the combination coefficients are optimized in the sense of minimum squared error between the prediction $\hat{x}_{i+1,j+1}$ and the actual value $x_{i+1,j+\ 1}$ for all possible i and j values. Via the optimization algorithm (that is described below), the linear filter coefficients: a_1, a_2 and a_3 are optimized, and the minimum prediction error result for optimum $\hat{x}_{i+1,j+1}$ at a particular (i,j) coordinate is estimated as

$$\varepsilon_{i+1,j+1} = \hat{x}_{i+1,j+1} - x_{i+1,j+1} \tag{4}$$

Consequently, the total error energy corresponding to the whole image prediction error can be calculated as

$$\varepsilon = \sum_{i=2}^{m} \sum_{j=2}^{n} \varepsilon_{i,j}^2 \tag{5}$$

where m and n correspond to the width and height of the image, which are, for the solar data, 365 and 24, respectively. The filter coefficients that minimize this function can be found from the solution of Eq. (6)

$$\frac{\partial \varepsilon}{\partial a_1} = \frac{\partial \varepsilon}{\partial a_2} = \frac{\partial \varepsilon}{\partial a_3} = 0 \tag{6}$$

The solution to six yields the following matrix–vector equation:

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \tag{7}$$

where R_{ij} is the correlation value between x_i and x_j within the prediction template, a_i is a linear filter coefficient, and r_k corresponds to the correlation between the predicted pixel, $x_{i+1,j+1}$, and each of the pixel samples within the prediction template. Particularly, r_1 is the correlation between $x_{i+1,j+1}$ and $x_{i,j}$, r_2 is the correlation between $x_{i+1,j+1}$ and $x_{i+1,j}$, and r_3 is the correlation between $x_{i+1,j+1}$ and $x_{i,j+1}$. The whole expression can also be compactly written as $\mathbf{R} \cdot \mathbf{a} = \mathbf{r}$. Eventually, the optimal filter coefficients can be obtained as

$$\mathbf{a} = \mathbf{R}^{-1} \cdot \mathbf{r} \tag{8}$$

where a is a vector containing the filter-tap coefficients, r includes the correlation values of the target pixel to the pixels in prediction template, and R includes correlation values within the prediction template (Gonzalez and Woods, 2002). Similar analysis holds for 1-D prediction, and it has been classically applied to speech compression. The performance comparisons of various sizes of 2-D prediction are presented in Section 5. The results indicate that, by using larger and truly 2-D prediction templates, better prediction performance can be achieved as compared to 1-D prediction.

4. Artificial neural networks

An alternative method to use in the proposed 2-D representation is the adaptive or nonlinear method that converges to a predictor for the solar radiation data. The classical and well known method for this purpose is the application of NNs which have been widely used in many areas, such as control, data compression, forecasting, optimization, pattern recognition, classification, speech, vision, etc. NNs have been utilized to overcome the limitations of the conventional and linear approaches to solve complex and nonlinear problems. In principle, NNs can be trained to solve problems that are difficult to model analytically (Haykin, 1999). Since NNs are not restricted to linear correlations, they can be used for nonlinear phenomena.

In a classical NN structure, each input (data sample within a template window) is multiplied by *a*, so called, connection weight. Then the products are summed and passed through a nonlinear transfer function to generate an output. The result is usually obtained as the thresholded output. The transfer function is an algebraic equation that can be linear or nonlinear. The training phase corresponds to updating and determination of the connection weights. One complete update calculation involving all the training patterns is called an epoch. At the end of each epoch, the system updates the parameters of the network. Typically, several such epoch passes are necessary to converge to a desired optimum. There are different connection styles and learning algorithms. A popular and simple algorithm is the Back Propagation algorithm which, as expected, consists of two phases: Training phase and recall phase. Before the training phase, the weights of the network are randomly initialized. Then, the output of the network is calculated and compared to the desired value. At each step during training, the error of the network is calculated and used to adjust the weights of the output layer by means of gradient methods (Haykin, 1999). If the network contains more than one layers of connections, the network error is also propagated backward and used to update the weights of the previous layers. Once the weights are determined after several training steps, the recall phase may run, where the network output computations are performed using assigned the input data and weights from the training phase.

The training phase is of interest, and it specifies the success of the resulting network. In back propagation, there are two different methods of updating the weights. In the first method, weights are updated for each of the input patterns using an iteration method. In the second method, an overall error for all the input output patterns of training sets is calculated (Yousefizadeh and Zilouchian, 2001). Once the weight update values are determined, the new weights and biases can be obtained using Eq. (9)

$$W_{ij,n} = U_n + \alpha W_{ij,n-1} \tag{9}$$

where $W_{ij,n}$ is a vector of current weights and biases, α is the momentum factor rate which determines how the past weights will reflect to the current value, and U_n is the update function. The determination of the update function is not unique, and it depends on the adopted error minimization method which can be chosen according to the problem and data set of interest. As an example, in the conjugate gradient-based minimization algorithms, the update is along conjugate direction where the error function decreases fastest. This method generally produces faster convergence than the celebrated steepest descent algorithms. There are other algorithms that produce faster convergence such as the Newton's method or the Levenberg–Marquard algorithm which considers the second derivative of the error function. The research and literature about NN realizations is vast, and it is not possible to cover all types of NN methods for a single application. Nevertheless, most applications are based on four basic and commonly encountered update types, namely (1) basic back propagation algorithm (BBP), (2) Conjugate gradient algorithm (CG), (3) Quasi-Newton algorithm (QN), and (4) Levenberg–Marquard (LM) algorithm. In this work, an implementation from each of these update types are performed, compared, and discussed. Brief explanations of each method are provided here.

4.1. Basic back propagation algorithm (Gradient descent algorithm)

The basic back propagation training algorithm works by altering the values of weights with a fixed length vector in the direction of the negative gradient to minimize an error function. Therefore, it is also called the gradient descent algorithm. The error function is the halved sum of squares of element-wise error terms as given in Eq. (10)

$$E = \frac{1}{2}\sum_{i=1}^{n}(t_i - o_i)^2 \tag{10}$$

where t_i is the target value (desired output) and o_i is the actual output and n is the number of input (or output) data dimension. It should be noted that Eq. (10) is valid for the networks that have one output vector. Consequently, the update function U_n (as in Eq. (9)) is defined by Eqs. (11) and (12):

$$g_n = \eta\frac{\partial E}{\partial W_{ij,n}} \tag{11}$$

$$U_n = -g_n \tag{12}$$

where w_{ij} are the weights of connections between neurons (or nodes) i and j, and η is the, so called, learning rate. A disadvantage of back propagation algorithm is its slow convergence. There are a lot of studies to increase the speed of convergence in training phase (Kamarthi and Pittner, 1999; Stäger and Agarwal, 1997; Osowski et al., 1996; Moller, 1993; Looney, 1996; Kinnebrock, 1994). Among the category of fast algorithms, the methods use standard numerical optimization techniques such as conjugate gradient, Quasi-Newton, and Levenberg–Marquard. These algorithms are discussed in the following subsections.

4.2. Conjugate gradient algorithms

These techniques focus on the local gradient of the error function, and its second derivative. The first derivative measures the slope of the error surface at a point, while the second one measures the curvature of the error surface at the same point. This information is critical for determining the globally optimal update direction. Since these methods make use of the second derivatives of the function to be optimized, they are typically referred to as second order methods. In particular, the conjugate gradient method is commonly used in training BP networks due to its speed and simplicity. Over the years, a large number of conjugate gradient functions (such as the Fletchere Reeves method) have been proposed. Conjugate gradient algorithm used in this study can be defined by using Eqs. (13) and (14)

$$U_0 = -g_0 \tag{13}$$

$$U_n = -g_n + \beta_n \times U_{n-1} \tag{14}$$

where g_n is defined as in Eq. (11) and hence the update function U_n varies at each step. Various versions of conjugate gradients can be constructed by different determination methods for the β_n constant. In the case of Fletcher–Reeves update, β_n is estimated using Eq. (15)

$$\beta_n = \frac{g_n^{\mathrm{T}} \times g_n}{g_{n-1}^{\mathrm{T}} \times g_{n-1}} \tag{15}$$

where$\{\cdot\}^{\mathrm{T}}$ denotes matrix or vector transposition.

This is the ratio of the norm squared of the current gradient to the norm squared of the gradient calculated at the previous step. Detailed information about Conjugate Gradient algorithms and its applications to NNs can be obtained from (Hagan et al., 1996).

4.3. Quasi-Newton algorithms

Newton's method is an alternative to the conjugate gradient methods for fast optimization. The basic step of Newton's method is

$$W_{ij,n} = W_{ij,n-1} - A_n^{-1} g_n \tag{16}$$

Here, A_n^{-1} is the Hessian matrix (containing second derivatives) of the performance index at the current step values of the weights and biases. Newton's method often converges faster than conjugate gradient methods. However, it is analytically more complex and expensive to compute due to the computation of the Hessian matrix.

A sub-class of Newton's method is called the Quasi-Newton (or secant) method, for which the calculation of second derivatives is not required. This method iteratively refines an approximate Hessian matrix, which eventually converges to the true Hessian matrix. The refined update depends on a function of the error gradient, therefore it is computationally efficient.

4.4. Levenberg–Marquardt algorithm

The most commonly used alternative to the above methods is the Levenberg–Marquardt (LM) algorithm. It is arguably the most popular method in application based studies. It can be considered as an alternative to the conjugate methods for second derivative optimization. In this algorithm, the update function, U_n, can be calculated using Eq. (17)

$$U_n = -[J^{\mathrm{T}} \times J + \mu I]^{-1} \times J^{\mathrm{T}} \times e \tag{17}$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The parameter μ is a scalar number and I is the identity matrix. The update becomes identical to the basic back propagation (with a small step size) whenever the μ parameter is large. Typically, μ value should be decreased after each successful step and should be increased only when a tentative step increases the error term (or performance function). Therefore, the performance function is guaranteed to reduce or get bounded at each iteration (Hagan and Menhaj, 1994).

While building a NN model should be selected, applied, evaluated and their performance should be verified for the problem that is dealt with. In this work, the algorithms mentioned above are all applied and the results are compared with each other. Eventually, Levenberg–Marquard algorithm is observed to produce more accurate results in the sense of lower RMSE. Algorithm selection procedure for this study is mentioned in Section 5 together with quantification methods for prediction results.

5. Prediction performance assessment

The prediction success of a mathematical model normally depends on the correlation among samples of the data. A strong correlation is an indicator of successful prediction with small prediction error. It is also critical to observe or determine which data samples are correlated with each other. In this work, it is argued that hourly variations of recorded solar radiation are correlated as well as the recordings corresponding to the same hour along consecutive days. Therefore, the proposed 2-D model causes both data samples with high correlations to become neighbors to improve prediction efficiency.

The importance of correlation in a time series for prediction can be explained as follows. If a series of n measurements of Y and Z are considered and written as y_i and z_i ($i = 1, 2,...,n$), then the sample correlation coefficient (ρ) can be used to estimate the correlation quantity of Y and Z as in Eq. (18)

$$\rho = \frac{n\sum_{i=1}^{n} y_i z_i - (\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} z_i)}{\sqrt{n(\sum_{i=1}^{n} y_i^2) - (\sum_{i=1}^{n} y_i)^2} . \sqrt{n(\sum_{i=1}^{n} z_i^2) - (\sum_{i=1}^{n} z_i)^2}} \tag{18}$$

The correlation coefficient is a *measure* of linear association between two variables. This measure automatically provides information regarding the success of a prediction algorithm. A successful predictor should yield a prediction error data with very low (almost zero) correlation coefficient. Values of the correlation coefficient are always between −1 and +1. A correlation coefficient of +1 indicates that two variables are perfectly related in a positive linear sense; a correlation coefficient of −1 indicates that two variables are perfectly related in a negative linear sense and a correlation coefficient of 0 indicates that there is no linear relationship between the two variables. The correlation coefficient (and its squared term, ρ^2) provides a measure to determine how certain one can be in making predictions from a certain model/graph. Since $0 \leqslant \rho^2 \leqslant 1$, the highest correlation is achieved at 1, corresponding to a full linear relation: $x_{ij} = ax_{kl} + b$ where a is the slope and b is the intercept. Within the concept of linear prediction, ρ immediately corresponds to the slope (a), and the intercept (b) must be taken as zero (assuming a prediction error with zero mean).

The methodology adopted here to determine the success of the prediction method for solar radiation is two folds. First, the prediction error is calculated using the root mean-square error (RMSE). Lower RMSE corresponds to better prediction. Next, the correlation coefficients corresponding to: (1) data samples, (2) prediction error samples, and (3) between the prediction and actual data, are provided. Experimental results indicate that the input data samples have strong correlation coefficient values for physically near samples. For instance, the correlation coefficient between neighboring hour samples are over 0.9. What is more interesting is the vertical correlation inside the 2-D representation. Experiments indicate that correlation coefficient between samples corresponding to the same hour of two consecutive days is also over 0.9. This information gives a strong hint about why the 2-D representation and corresponding 2-D predictors would exploit the dependencies better than 1-D predictors. Finally, it was also observed that the prediction values for each of the above methods have very high correlation to the actual values ($\rho > 0.96$) for 2-D predictors. This high correlation indicates a strong resemblance between the prediction and the actual value. The correlation coefficients for 1-D predictors, however, were found slightly less ($\rho > 0.91$). Specific experimental results are presented in Section 6.

6. Experimental results

The correlation coefficients between solar radiations at any hour of any day i ($X_{\mathrm{H}i,\mathrm{D}i}$) and day j ($X_{\mathrm{H}j,\mathrm{D}j}$) are calculated and tabulated for up to 3 h (horizontal) and two days (vertical) difference. This corresponds to a vertical distance of up to two pixels and horizontal distance of up to three pixels in the 2-D representation. The results are presented in Table 1. It can be seen that the correlation decreases as the pixel difference increases in both directions. It is worthy to note, however, that the amount of correlation decrease in the vertical direction (along days) is quite competitive to the horizontal correlation. Experimentally, it was observed that the decrease in correlation coefficient value was rapid in both horizontal and vertical directions for geometric differences above 4 pixels. As a result, prediction templates that are wider (and taller) than ones presented here produce inferior results.

In Table 1, the terms "$x_{ij} - x_{kl}$" indicate correlations between x_{ij} and x_{kl}, and ρ is calculated from Eq. (18). As explained above, the critical observation from Table 1 is that there are strong correlations between the radiations not only for consecutive hours, but also for the hours of consecutive days. As a matter of fact, the correlation between two consecutive days at the same hour is *stronger* than the correlation between 2 h of the same day separated by 2 h. Therefore, when constructing a prediction model, the data from the previous day must be taken into account for higher accuracy. It is clear that solar data readings separated by 2 h vary considerably during day–night transitions, therefore extraterrestrial normalization might improve the correlation (Vignola et al., 2007). However, it is suggested here that correlation in the other dimension (daily variations across the same hour) automatically provides a simple normalization for the prediction.

In order to reduce computational complexity and to avoid incorporation of poorly correlated (far away) samples that reduce the performance, relatively short 1-D and 2-D prediction filters are used in this work. The utilized 1-D and 2-D filter templates are given in Figs. 4 and 5, respectively. The first set of filters (in Fig. 4) is purely single directional, whereas the second set (in Fig. 5) contains filter samples from different rows and columns, making them inherently two-dimensional. These

Table 1
Correlation coefficients

Pixels	ρ	ρ^2
x_{44}–x_{34}	0.911	0.830
x_{44}–x_{24}	0.894	0.799
x_{44}–x_{14}	0.898	0.806
x_{44}–x_{43}	0.938	0.879
x_{44}–x_{42}	0.807	0.651
x_{44}–x_{41}	0.630	0.397
x_{44}–x_{33}	0.870	0.760
x_{44}–x_{22}	0.740	0.548

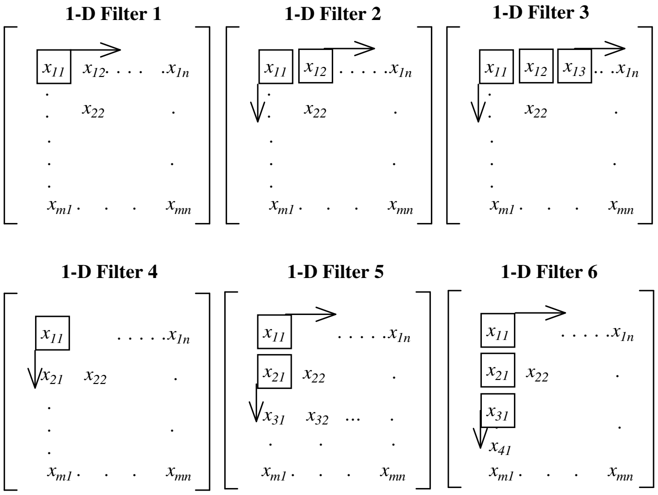

Fig. 4. Single direction (either horizontal or vertical) prediction filter templates. Template domains are indicated with bounded boxes.

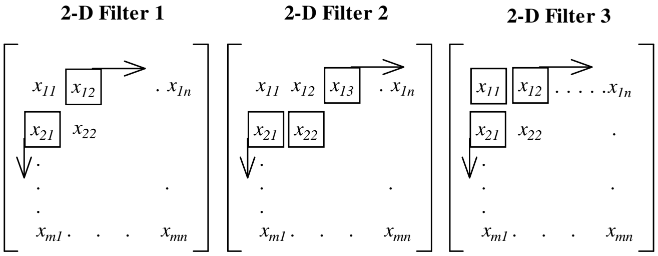

Fig. 5. 2-D multiple direction prediction filter templates with domain samples from both vertically and horizontally different columns and rows. Template domains are indicated with bounded boxes.

templates are also widely used in predictive image and signal coding.

For the minimum RMSE linear prediction, the optimal coefficients given in Table 2 are analytically determined for each filter by solving Eq. (8). The 2-D image data is fed to the prediction system, and RMSE's are obtained for each hour. The prediction domain was selected as one of the templates given in Figs. 4 and 5. The filter values and structures are selected according to the correlation analysis results of Table 1. These domain templates are used for both the linear prediction optimization, and the NN models. The models all have multiple inputs (past samples) selected from the domain template, and they have a single output, corresponding to the prediction. Up to three inputs were proposed for the NN structure, therefore 1-input, 2-input, and 3-input NN structures were developed as shown in Figs. 6–8. Since the input and output templates of the NNs are selected as the same set used in linear predictive filter optimization, direct comparison between optimum linear filtering and NNs was possible.

Table 2
Optimal filter coefficients

	a	b	c
1-D Filter1	1.043	N/A	N/A
1-D Filter2	−0.600	1.533	N/A
1-D Filter3	−0.170	−0.337	1.431
1-D Filter4	0.941	N/A	N/A
1-D Filter5	0.382	0.581	N/A
1-D Filter6	1.093	−0.099	−0.743
2-D Filter1	0.389	0.6033	N/A
2-D Filter2	−0.504	0.693	0.799
2-D Filter3	0.238	−0.445	1.168

N/A: Not available.

Specifically, since there are two variants of single input single output domain templates (one vertically neighbor,

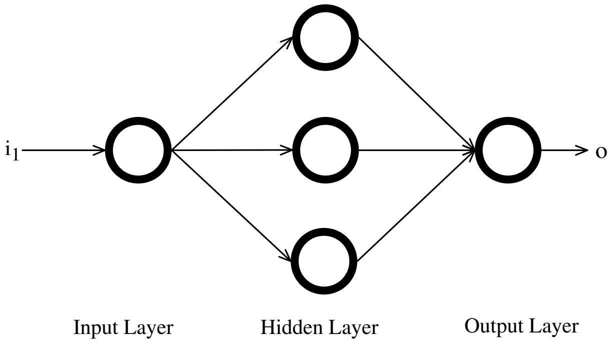

Fig. 6. An example one-input (i_1) and one-output (o) NN structure that implements the time varying and dynamic function:($o = f(i_1)$.

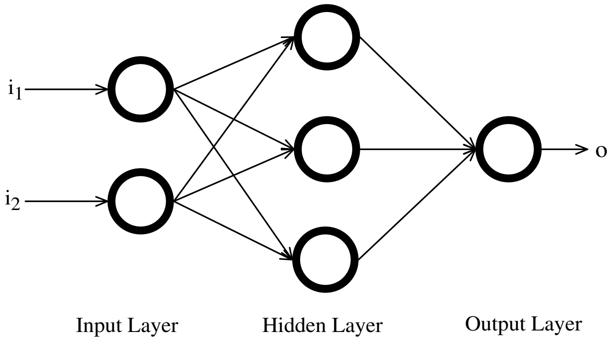

Fig. 7. A two-input (i_1, i_2) and one-output (o) NN structure to implement a 2-D function $o = f(i_1, i_2)$.

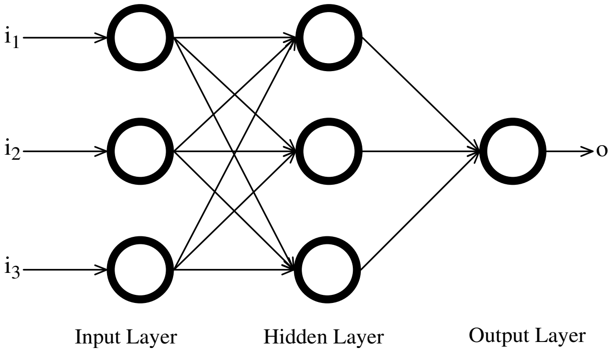

Fig. 8. A three-input (i_1, i_2, i_3) and one-output (o) NN structure to implement a 3-D function $o = f(i_1, i_2, i_3)$.

and one horizontally neighbor), the networks labeled as NN1 and NN4 have the structure as given in Fig. 6. Similarly, three networks were possible with two inputs, so the structure of NN2, NN5 and NN7 are as given in Fig. 7. Finally, three input networks labeled as NN3, NN6, NN8 and NN9 have structures as given in Fig. 8. For all NN cases, ten months solar radiation data are used for training and 2 months are used for testing. The surface plot of the test data is given in Fig. 9.

In all of the above NN structures, the sigmoid function was adopted and all of the learning methods described in Section 4 (which are basic back propagation, conjugate gradient method, Quasi-Newton algorithm, and Levenberg–Marquard) were tested with various neuron sizes in the hidden (center) layer. It was experimentally found that the fastest convergence with the smallest prediction error was obtained by the Levenberg–Marquard (LM) method during the learning process with three neurons at the hidden layer. For example, the prediction error RMSE values for these methods with NN8 (which has the most successful domain template shape) were 72.45 for Basic BP, 45.94 for CG, 37.03 for QN, and 35.42 for the LM algorithms. For all structures the algorithm ran 100 epochs. To show the sufficiency of this epoch number, the convergence as a function of epoch for NN8 is illustrated in Fig. 10. From here on, the LM algorithm will be adopted as the learning algorithm, the number of hidden neurons will be selected as three.

During the learning process, it was observed that most of the learning is completed within 10 epochs (as seen from Fig. 10). After the learning phase, the network is tested by the data corresponding to the remaining solar radiation acquisition samples (Fig. 9). An exhaustive cross test is performed for various NN domain templates that were previously described. The prediction performances were observed to be close (<3%) for all 2-D templates with NN8 having the lead. On the other hand, the 1-D templates were found inferior. In Fig. 11, the prediction error values for NN8 are rendered in 2-D. For the data presented in this figure, two important observations can be made; (1) the energy of the prediction error is much lower than the solar data, itself, and (2) the prediction error samples have no correlation with the input samples (calculated correlation coefficient was 0.0056, practically corresponding to zero correlation). The uncorrelatedness of the prediction error samples indicates that the prediction almost totally exploits the predictable part of the data by producing a residue corresponding to a white noise.

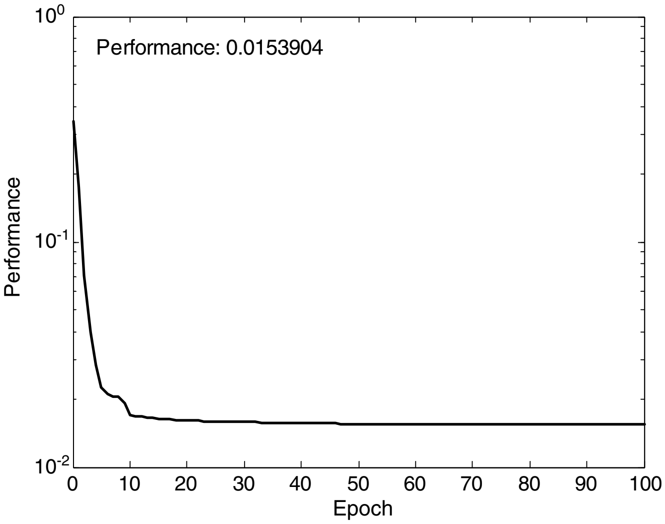

Fig. 10. Performance graph of NN8 for varying epoch numbers.

For the same network structure (NN8), the actual solar radiation samples are correlated with the prediction out-

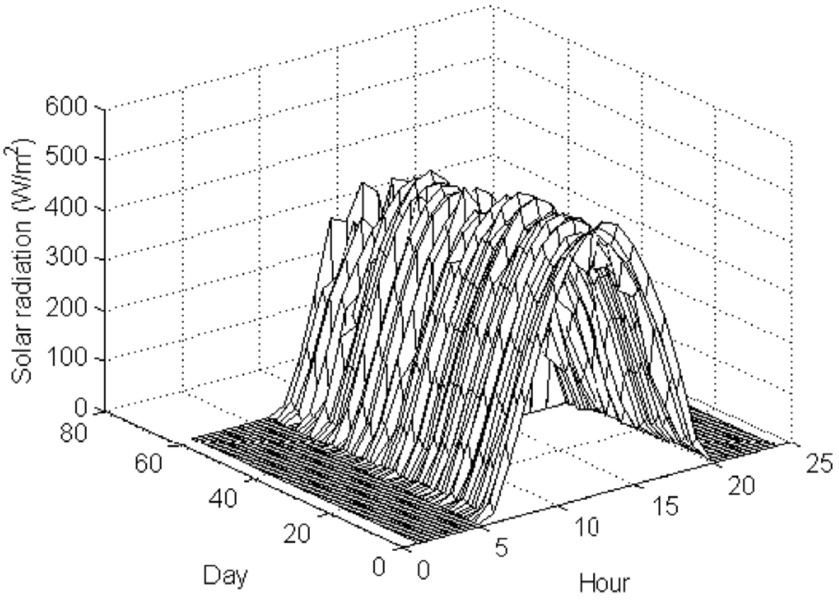

Fig. 9. 2-D Surface (or mesh) plot of test data.

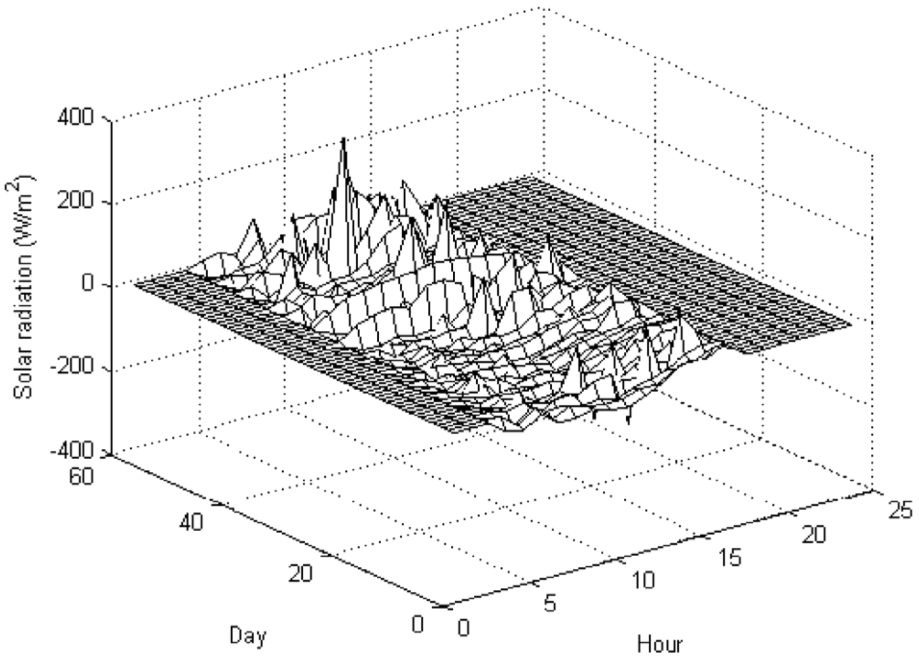

Fig. 11. Prediction error (in 2-D image form) for the test data in Fig. 9 using NN8.

puts of the network. The correlation analysis is performed separately for the training and test data, and presented in Figs. 12 and 13, respectively. The figures are presented in the form of actual data – versus – predicted data plots. It is clear from Figs. 12 and 13 that prediction results are closely matching actual data along the diagonal axis. Therefore, the slope of the linear fit for Figs. 12 and 13 are close to 45 degrees, and the scatter is narrow along the matching diagonal axis. These observations indicate a high prediction success for the proposed NN algorithm. The same analysis is also carried out for the linear prediction filter, and the result for Filter 3 variant (which was experimentally the most successful) is presented in Fig. 14. Again, the plot contains a narrow match along the diagonal axis, indicating a successful prediction with high correlation and small deviation. This observation is also verified by the small prediction error samples for the same filter which are presented in 2-D form in Fig. 15. It can be noted

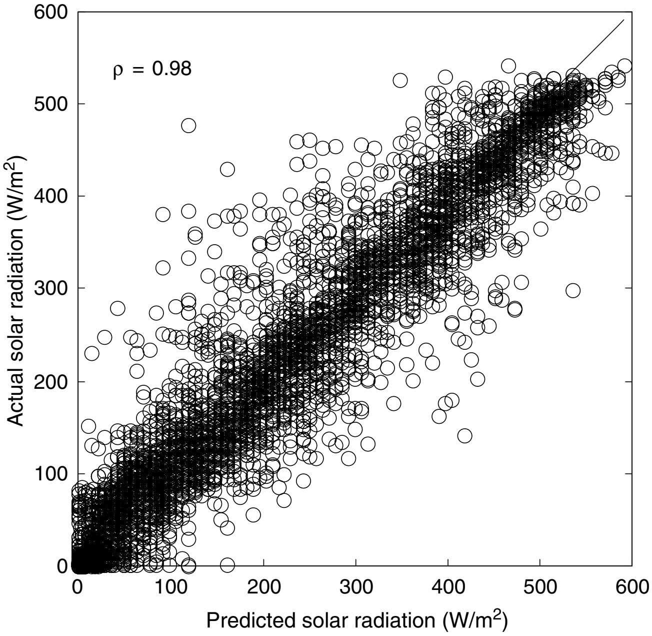

Fig. 12. Correlation between the actual and predicted solar radiation values based on *training* data for NN8.

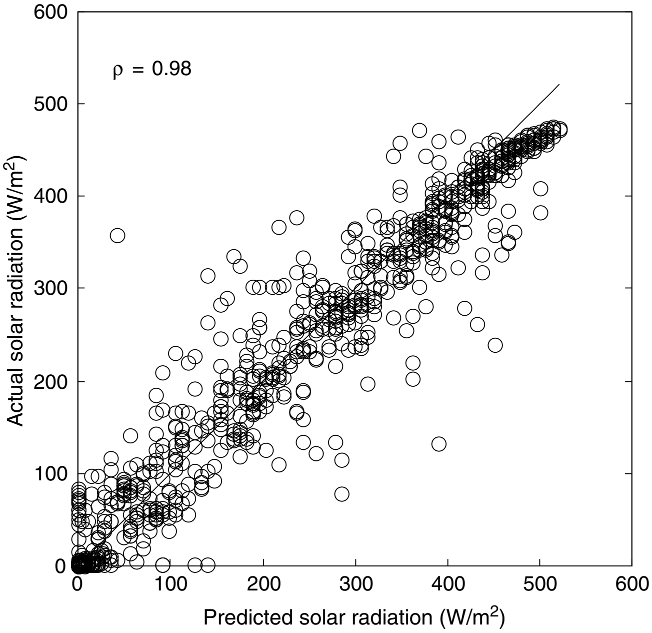

Fig. 13. Correlation between the actual and predicted solar radiation values based on *testing* data for NN8.

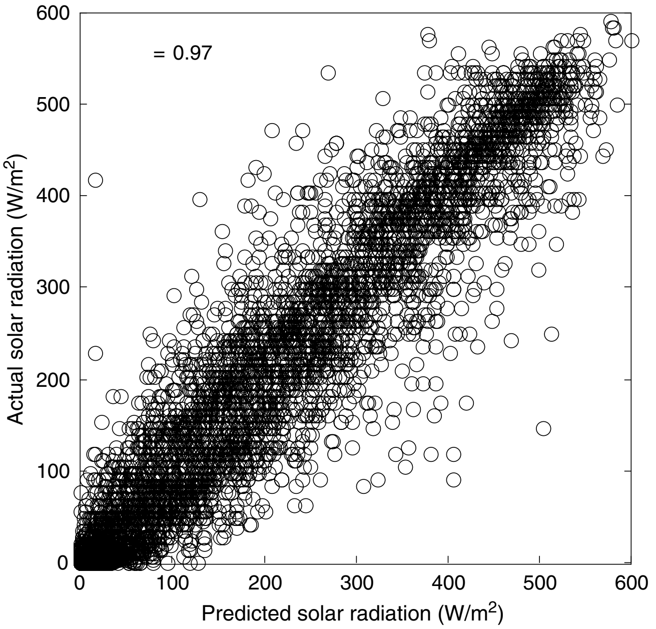

Fig. 14. Correlation between the actual and predicted solar radiation values for *2-D Filter 3* linear prediction filter.

that the values of the surface function (corresponding to the prediction error) in Fig. 15 are significantly lower than the actual values presented in Fig. 2.

Using the above techniques, RMSE values and correlation coefficient values between actual data and predicted samples are obtained. The experimental results for all linear predictive filters and NNs are presented in Table 3, which completely provides the sufficient information for the comparisons of 1-D versus 2-D, and linear prediction versus NN methods. In this table, the RMSE values obtained using 2-D predictors are clearly *smaller* than the values obtained using 1-D predictors. This shows that the proposed 2-D representation gives better insight to the solar pattern than the regular 1-D interpretation. In this table, it can also be seen that NNs provide better prediction results as compared to the linear prediction filters. This is an expected result due to the ability of NN's to adapt to nonlinearities that are inherent in seasonal change of data.

In order to test the robustness of the proposed methods, the acquired solar radiation data was deliberately disturbed down to complete blank for three consecutive days (5, 6, 7th days of February). Due to the generalized modeling success of the NN and linear filter methods, the new results for the perturbed data stayed almost the same. The filter coefficients of the linear predictive filters were similar to the original data (less than 0.2% different in magnitude) and the produced prediction error was the same as the original prediction error (less than 0.4% different in RMSE). Since the internal structure of the NNs could not be observed, only the prediction performances were compared, and it was observed that the prediction error RMSEs are, again, less than 0.4% different compared to the original results. As an example, the RMSE value for 2-D filter NN2 is calculated as 40.38 using disturbed data, whereas it was calculated as 40.23 using original data. The results indicate that the NN training phase and linear filter optimization method are fairly robust to input data discontinuities. Nevertheless, since the acquired data was constantly monitored in our laboratory, the quality of the recorded data was assured, and this analysis was only performed to explain the robustness of the described methods in case of perturbed or erroneous data.

A point of concern is the selection of the training and test data. The test data (depicted in Fig. 9) was arbitrarily selected as the 8th and 9th months of the year. These months correspond to seasonal transition from summer

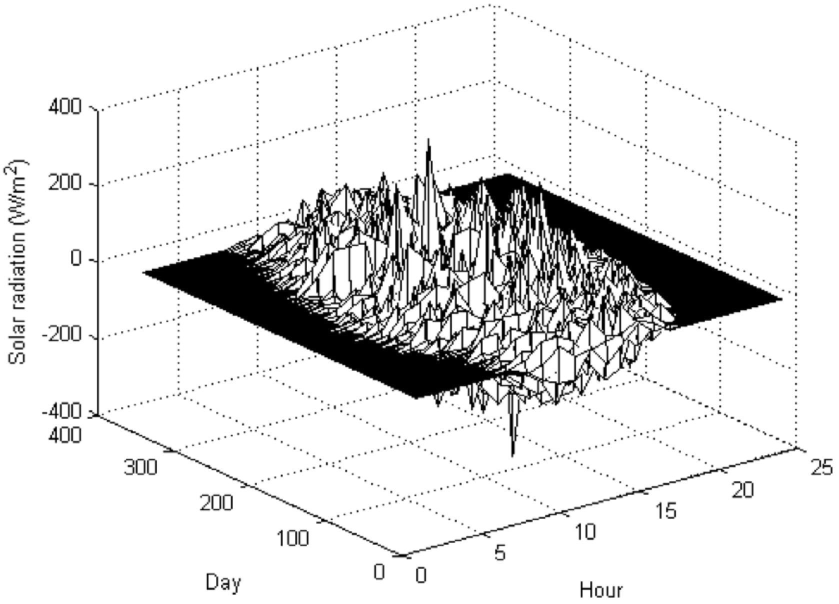

Fig. 15. Prediction error (in 2-D image form) obtained from *2-D Filter 3*.

Table 3
The RMSE and correlation coefficients (ρ) values obtained from the NN and optimum linear filters

	Linear filters		Neural networks			
			Training data		Testing data	
	RMSE	ρ	RMSE	ρ	RMSE	ρ
1-D Filter 1/NN1	58.53	0.94	58.80	0.93	62.64	0.94
1-D Filter 2/NN2	45.02	0.96	41.86	0.97	39.79	0.98
1-D Filter 3/NN3	44.25	0.96	44.70	0.96	43.73	0.97
1-D Filter 4/NN4	67.05	0.91	69.30	0.90	74.32	0.91
1-D Filter 5/NN5	61.94	0.92	63.95	0.92	67.63	0.92
1-D Filter 6/NN6	76.49	0.88	60.83	0.93	63.64	0.93
2-D Filter 1/NN7	46.56	0.96	45.58	0.96	47.98	0.96
2-D Filter 2/NN8	40.23	0.97	35.42	0.98	34.57	0.98
2-D Filter 3/NN9	39.37	0.97	42.15	0.97	43.15	0.97
1-D Filter 1/NN1	58.53	0.94	58.80	0.93	62.64	0.94

to winter with a moderate amount of sunny days and occluded days. The test results (provided herein) are not found to change with different choices of test months. This is also clear from the fact that the training set errors and test set error results are similar in magnitude. Hence, unless unexpected atmospheric events (e.g. totally cloudy days during summer time, or vice versa) are not persistently present for a long duration (e.g. 2 weeks), the model gives similar results by different choices of test months. In order to take into account "expected" cloudy days during a season, and incorporate this information into the constructed model, longer (several years) data collection might be necessary.

7. Conclusions

In this work, a novel 2-D approach, which was recently proposed (Hocaoğlu et al., 2007) for hourly solar radiation forecasting, is comprehensively evaluated through linear and neural network prediction models. The hourly solar radiation data are rendered as a 2-D image and its correlation properties are examined. It is initially observed that 2-D representations give more insight to the solar pattern than the regular 1-D interpretation. Furthermore, the 1-D and 2-D interpretations are found to have significant correlation efficiency differences. In the first stage, six 1-D and three 2-D optimal linear prediction filters are designed with different prediction domain templates. Optimal filter coefficients are obtained and prediction results are compared in the sense of RMSE. It was observed that the correlation coefficients between 2-D template domain elements and the target element are always better than the correlation coefficients obtained using 1-D templates. The correlation property reflects to the prediction error values obtained using 2-D and 1-D filter templates, and the prediction errors for 2-D templates are found to be smaller. Similar representation efficiency is observed by feed-forward NNs with the same domain template structures used in linear prediction. To enable the same templates, three types of NN structures are designed. The same data are applied to NNs and prediction errors are compared. It is observed that 2-D templates for NNs outperform 1-D templates, in general. It is also observed that the NNs particularly perform better than linear prediction filters for both 1-D and 2-D. This is due to the nonlinear and time-varying nature of the solar radiation data. It is concluded that the 2-D representation has many potential uses for solar radiation data modeling. Dynamical time varying behavior of the 2-D model may also be analyzed through adaptive filters. Such analysis can be regarded as future works of this study.

Acknowledgements

This work is supported in parts by TUBITAK (Turkish NSF) Grant no: 107M212, and Anadolu Univ. Research Fund Contract no: 040258.

References

Amato, U., Andretta, A., Bartolli, B., Coluzzi Cuomo, B.V., Fontana, F., Serio, C., 1986. Markov process and Fourier analysis as a tool to describe and simulate solar irradiation. Solar Energy 37, 197–201.

Aguiar, J., Collares-Perrira, M., Conde, S.P., 1988. Simple procedure for generating of daily radiation values using library of Markov transition matrices. Solar Energy 49, 229–279.

Aguiar, R., Collares-Pereira, M., 1992. A time dependent autoregressive Gaussian model for generating synthetic hourly radiation. Solar Energy 49, 167–174.

Cao, J.C., Cao, S.H., 2006. Study of forecasting solar irradiance using neural networks with preprocessing sample data by wavelet analysis. Energy 31, 3435–3445.

Chena, R., Kanga, E., Jia, X., Yanga, J., Wang, J., 2007. An hourly solar radiation model under actual weather and terrain conditions: a case study in Heihe river basin. Energy 32, 1148–1157.

Cucumo, M., De Rosa, A., Ferraro, V., Kaliakatsos, D., Marinelli, V., 2007. Experimental testing of models for the estimation of hourly solar radiation on vertical surfaces at Arcavacata di Rende. Solar Energy 81, 692–695.

Gonzalez, R.C., Woods, R.E., 2002. Digital Image Processing, second ed. Prentice-Hall, Englewood Cliffs, USA, pp. 461–463.

Hagan, M.T., Demuth H.B, Beale M.H., 1996. Neural Network Design. PWS Publishing, Boston, MA.

Hagan, M.T., Menhaj, M.B., 1994. Training feedforward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks 5, 989–993.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River, NJ.

Hocaoğlu, F.O., Gerek, Ö.N., Kurban, M., 2007. A novel 2-D model approach for the prediction of hourly solar radiation. LNCS Springer 4507, 741–749.

Kamarthi, V., Pittner, S., 1999. Accelerating neural network training using weight extrapolations. Neural Networks 12, 1285–1299.

Kaplanis, S., 2006. New methodologies to estimate the hourly global solar radiation: comparisons with existing models. Renewable Energy 31, 781–790.

Kaplanis, S., Kaplani, E., 2007. A model to predict expected mean and stochastic hourly global solar radiation *I*(*h;nj*) values. Renewable Energy 32, 1414–1425.

Kinnebrock, W., 1994. Accelerating the standard backpropagation method using a genetic approach. Neurocomputing 6, 583–588.

Kurban, M, Hocaoglu, F.O., 2006. The construction of hybrid (Wind-Solar) power plant model by determining the wind and solar potential in the Iki Eylul Campus of A.U., Anadolu University Scientific Research Project, Project no: 2004/040258 (in Turkish).

Looney, C.G., 1996. Stabilization and speedup of convergence in training feedforward neural networks. Neurocomputing, 7–31.

Maafi, A., Adane, A., 1989. A two state Markovian model of global irradiation suitable for photovoltaic conversion. Solar and Wind Technology 6, 247–252.

Mellit, A., Benghanem, M., Hadj Arab, A., Guessoum, A., 2005. A simplified model for generating sequences of global solar radiation data for isolated sites: using artificial neural network and a library of Markov transition matrices approach. Solar Energy 79, 469–482.

Moller, M.F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6, 525–533.

Osowski, S., Bojarczak, P., Stodolski, M., 1996. Fast second order learning algorithm for feedforward multilayer neural networks and its applications. Neural Networks 9, 1583–1596.

Sfetsos, A., Coonıck, A.H., 2000. Unıvariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques. Solar Energy 68, 169–178.

Stäger, F., Agarwal, M., 1997. Three methods to speed up the training of feedforward and feedback perceptrons. Neural Networks 10, 1435–1443.

Vignola, F., Harlan, P., Richard, Kmiecik M., 2007. Analysis of satellite derived beam and global solar radiation data. Solar Energy 81, 768–772.

Yousefizadeh, H., Zilouchian, A., 2001. Neural Network Architectures, Chap. 3, Intelligent Control Systems Using Soft Computing Methodologies. CRC Press, USA, pp. 41–50.