

Hybrid intra-hour DNI forecasts with sky image processing enhanced by stochastic learning

Yinghao Chu, Hugo T.C. Pedro, Carlos F.M. Coimbra^{*}

Department of Mechanical and Aerospace Engineering, Jacobs School of Engineering, Center for Renewable Resource Integration and Center for Energy Research, University of California, 9500 Gilman Drive, La Jolla, CA 92093, USA

Received 3 May 2013; received in revised form 15 October 2013; accepted 19 October 2013
Available online 16 November 2013

Communicated by: Associate Editor Christian A. Gueymard

Abstract

We propose novel smart forecasting models for Direct Normal Irradiance (DNI) that combine sky image processing with Artificial Neural Network (ANN) optimization schemes. The forecasting models, which were developed for over 6 months of intra-minute imaging and irradiance measurements, are used to predict 1 min average DNI for specific time horizons of 5 and 10 min. We discuss optimal models for low and high DNI variability seasons. The different methods used to develop these season-specific models consist of sky image processing, deterministic and ANN forecasting models, a genetic algorithm (GA) overseeing model optimization and two alternative methods for training and validation. The validation process is carried over by the Cross Validation Method (CVM) and by a randomized training and validation set method (RTM). The forecast performance for each solar variability season is evaluated, and the models with the best forecasting skill for each season are selected to build a hybrid model that exhibits optimal performance for all seasons. An independent testing set is used to assess the performance of all forecasting models. Performance is assessed in terms of common error statistics (mean bias and root mean square error), but also in terms of forecasting skill over persistence. The hybrid forecast models proposed in this work achieve statistically robust forecasting skills in excess of 20% over persistence for both 5 and 10 min ahead forecasts, respectively.

© 2013 Elsevier Ltd. All rights reserved.

Keywords: Direct normal irradiance; Artificial neural networks; Genetic algorithms; Sky imaging; Smart solar forecasting

1. Introduction

While the solar irradiance at the ground level offers the opportunity of producing clean and abundant power, the variability of ground irradiance levels caused by atmospheric processes compromises the reliability of the solar resource for power production and increases the associated integration costs to the power grid. The solar beam component of solar irradiance, also referred as Direct Normal Irradiance (DNI), is critical for all concentrated solar technologies and is particularly sensitive to fluctuating values of

atmospheric cloud cover and aerosol content. Rapid fluctuations of DNI irradiance can be observed daily in most locations, and these fluctuations can bring the ground DNI levels from several hundreds of W/m^2 to zero in a few seconds. These very steep irradiance level ramps are normally associated with cloud cover, but aerosol content may also affect DNI substantially, albeit at longer time scales.

From the point of view of energy management, the power grid needs to be balanced in real-time, and only limited low-cost storage and spinning reserves are generally available. At increasingly higher solar penetration levels, sudden power drops caused by atmospheric processes can adversely affect local grid stability and power quality, with possible domino effects throughout adjacent power nodes.

^{*} Corresponding author. Tel.: +1 858 534 4285.

E-mail address: ccoimbra@ucsd.edu (C.F.M. Coimbra).

Accurate solar, wind and load forecasting are essential components of a smarter, more dynamic power grid that is prepared to absorb increasingly higher levels of weather-dependent renewable resources. In this work we develop and analyze a number of different forecasting methodologies for short-term prediction of DNI. We focus on two important time horizons for power plant operation, real-time dispatch, and trading (5 and 10 min ahead).

The simplest forecast model for short-term horizons is the persistence model, which is still widely used and works well for periods of low solar irradiance variability. Unfortunately, the accuracy of persistence models is reduced substantially when solar irradiance variability increases. Statistically, persistence models tend to perform well over long periods of time because invariant conditions are statistically common (either consistently clear or consistently overcast days are common in many micro-climates), but persistence modeling misses all important ramp rates that are more likely to cause grid integration problems. Because we are mostly interested in the ability to predict these irradiance and/or power ramps, there is a critical need for the development of robust methods that can take into account the presence and positioning of the clouds in respect to Sun.

Some forecast models based on historical time series analysis (e.g., auto-regressive or stochastic learning methods) show marginal improvements over persistent models (Mellit and Kalogirou, 2008; Martín et al., 2010; Sfetsos and Connick, 2000; Cao and Lin, 2008; Mellit and Pavan, 2010; Mellit et al., 2010; Marquez and Coimbra, 2013a; Marquez and Coimbra, 2013b; Pedro and Coimbra, 2012; Marquez et al., 2013a). Among these models, methods based on Artificial Neural Networks (ANN) are some of the most widely used (Mellit and Kalogirou, 2008; Martín et al., 2010; Cao and Lin, 2008; Mellit and Pavan, 2010; Mellit et al., 2010; Marquez and Coimbra, 2011; Pedro and Coimbra, 2012; Marquez et al., 2013a; Marquez et al., 2013b). Elizondo et al. (1994) used ANNs to predict daily solar radiation, and early applications of ANNs to forecast DNI can be found in the work by Lopez in 2005 (López et al., 2005).

Most of the models cited above were constructed without considering spatio-temporal information on clouds, which are known to be the factor affecting ground irradiance levels the most. Some models that incorporate sky cover information use remote sensing (Hammer et al., 1999; Perez et al., 2010; Marquez et al., 2013b). However, the temporal resolution (typically 30 min) and the spatial resolution (in the order of kilometers) of satellite images are inappropriate for accurate intra-hour forecast of solar irradiance. The lack of spatial resolution from satellite images can be overcome by using local high-frequency image-capturing devices, such as Total Sky Imagers (TSIs).

A few forecasting models (Marquez and Coimbra, 2013a; Chow et al., 2011; Ruano et al., 2008; Marquez et al., 2013a) use information about cloud cover extracted from TSIs. Marquez and Coimbra (2013a) described a deterministic DNI forecast methodology based on TSI images. For forecast horizons in the range of 3–15 min,

their model was able to achieve significant improvement over the persistence model on several days characterized as high DNI variability periods.

Studies of integrating cloud tracking techniques with machine learning methods for global irradiance forecast have emerged in the recent years (Marquez et al., 2013a,b). In the present work, a smart DNI forecasting model is developed based on Artificial Neural Networks, which uses cloud coverage time-series extracted from a TSI in combination with high-quality DNI time-series as inputs to forecast DNI for the specific horizons of 5 and 10 min. ANN-based forecasting models require a learning process, in which the ANN parameters are fitted to the learning data. Because of seasonal differences in the data used in this work we divided the data into two subsets: a “clear weather” subset identified by low DNI variance and a “cloudy weather” subset characterized by high DNI variance. A genetic algorithm (GA) was used to search for the optimal definition of the ANN-based forecasting models. Because these models depend on the season, we used the best forecasting model from each season to create a season-independent smart model, whose performance is evaluated on an independent testing set.

This paper is organized as: in Section 2 we discuss the data collection, quality control and separation of data by seasons. We explain the details of the methods in Section 3, including TSI image processing and generation of the cloud index time-series (CI), the deterministic model, the validation methods for the ANN, the GA optimization, the smart model and the testing procedures to assess the models’ performance. Results are discussed in Section 4 and conclusions are presented in Section 5.

2. Irradiance data and sky images

Experimental data for this study were collected at our observatory in Merced (latitude = 37.36, longitude = −120.43), CA during the Spring, Summer and Fall of 2011 (04/28/11–10/15/11).

The DNI data was measured with Eppley’s Normal Incidence Pyrheliometer (NIP) at a sampling period of 30 s. Sky images were recorded with the Yankee Environmental Systems (YES) Total sky imager (TSI 880) every 20 s. The TSI is composed of a rotating hemispherical mirror which reflects the sky image onto a CCD camera. The images are saved in an online database from where they are retrieved and processed. The processing techniques used to transform the sky image into numerical cloud cover indices are explained in the next section. TSI built-in software computes the daily sun percentage (SP) index, which gives an estimate for the fraction of daily images in which the sun is shining. The daily SP values for the period analyzed are plotted in Fig. 1. These values were used to classify the weather type for a given day of interest.

Data quality is essential to a good forecasting model. For this reason, sky images showing misaligned shadowband or excessive amount of dust (deposited on the TSI’s mirror)

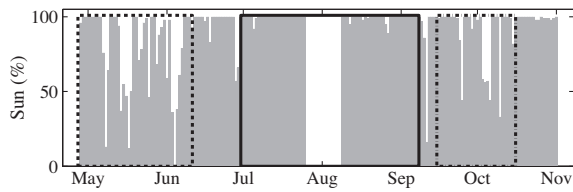


Fig. 1. The daily sun percentage (SP) plot for the period from 04/28/11 to 10/30/11. Three subsets are created based on this plot: low DNI variance season (identified by the solid-line rectangle), high DNI variance season (identified by the dash-line rectangle), and independent testing set (identified by the dash dot line).

are removed. With respect to the DNI, besides the Eppley's NIP, a rotating shadowband radiometer (the MFR-7 from YES) was at the same location to provide redundant irradiance measurements for data quality control.

Merced has a Mediterranean climate with rainy winters and dry summers as supported by the SP values in Fig. 1. In summer the daily SP is close to 100%, while spring is cloudy and the daily SP is relatively low for most of the time. Fall is composed of significant amounts of both high and low SP days. The conclusion from this analysis is that the DNI data and sky images can be categorized into two very distinct phases: the low DNI variance phase ("clear time"), characterized by low variability of the DNI time-series associated to clear-sky conditions, and the high DNI variance phase associated with "cloudy" weather conditions.

Motivated by the very different levels of DNI variability between these phases, we developed different forecasting models specific for each phase: low DNI variance models estimated from summer data, and high DNI variance models estimated from spring data (see data partition in Fig. 1). In order to create the models, spring and summer data were separated into two disjointed subsets: the learning set and the seasonal testing set. The learning set is used in model estimation, and the seasonal testing set is used to assess the forecasting models' performances for their respective season. Because the models estimated with one season's data may be sub-optimal for the other season, we used fall data (which contains both high and low variance periods of DNI) as an independent testing set to evaluate and compare the forecasting models independently of the season. In total we have five different subsets that are listed in Table 1.

3. Methods

In this section we discuss the key steps to create the smart forecasting models: the TSI image processing

methodology that generates the cloud cover index (CI) time-series, which is the building block of both deterministic and ANN-based forecasting models, and the optimization of the seasonal models through the GA, which uses two different validation methods as optimization objective functions. The testing process used to assess the forecast accuracy is discussed at the end of this section.

3.1. Clear-sky model

The different forecasting models developed in this work require knowledge of clear-sky irradiance. In this work we developed an empirical expression specific to our data, in which the clear-sky DNI is given by a polynomial expression that depends on the cosine of the solar zenith angle θ :

$$B_{clr} = \sum_{n=0}^8 a_n (\cos(\theta))^n \quad (1)$$

where the coefficients a_n were obtained using the least squares method (LSM) for 20 days of clear-sky DNI data selected manually from all our data. The LSM returns the following coefficients:

$$a_n = \{42.72, -146.9, 172.8, -50.98, -57.49, 58.79, -23.93, 5.898, -0.01045\}$$

We model the clear sky DNI in this way for three reasons: (1) Most clear sky models in literature work very well to model GHI but not so well for DNI, as DNI is much more sensitive to model inputs. (2) Additional input data (atmospheric aerosol optical depth, water vapor column, etc.) are required for DNI calculations. (3) Although this expression is particular to Merced, the methodology used to create the curve fitting is general and should work well for any other location. To implement this forecast elsewhere, only the coefficients in Eq. (1) need to be updated with local ground telemetry data.

3.2. Persistence model

The simplest forecasting model is the persistence model. Despite its simplicity, it can be very accurate for periods of low DNI variability. It is also useful to benchmark other models. In this work the persistence model assumes that the clear-sky index B/B_{clr} remains constant between t and $t + \Delta t$. Mathematically, the persistence model is given by:

$$\hat{B}(t + \Delta t) = \frac{B_{clr}(t + \Delta t)}{B_{clr}(t)} B(t) \quad (2)$$

Table 1
Analysis periods.

Season	Sub set	Label	Range	Data points (approx.)
Low variability	Learning period	LV_l	07/01/11–08/25/11	19,000
	Seasonal testing	LV_t	08/26/11–09/07/11	5000
High variability	Learning period	HV_l	04/28/11–06/05/11	14,000
	Seasonal testing	HV_t	06/05/11–06/10/11	2500
Independent testing	–	IT	09/15/11–10/15/11	7000

3.3. TSI image processing

The algorithm to convert the TSI images into numerical information for the forecasting models follows directly from previous work done by our research group (Marquez and Coimbra, 2013a). This algorithm can be summarized in four main steps: projection of the original image onto a rectangular grid, computation of the velocity vectors, identification of cloud pixels, and the calculation of cloud indices.

In the first step, in order to remove the image distortion caused by the convex mirror, all images are projected from the convex mirror space onto a flat rectangular space. The projection method converts the pixels' locations of the original image into polar coordinates:

$$x = r_i \cos(\phi) \quad y = r_i \sin(\phi) \quad (3)$$

where r_i is the radial distance from the zenith pole (the center of the image) to the pixels' coordinate (x, y) . Ideally, the r_i is linearly related to the tangent of the zenith angle. Due to the image distortion, the relationship between r_i and $\tan(\theta)$ is in fact nonlinear. The relationship between r_i and $\tan(\theta)$ is approximated by fitting the sun's apparent radial distance ($r_{i, \text{sun}}$) to the tangent solar zenith angle ($\tan(\theta)$) through a fourth order polynomial obtained from pairs of $r_{i, \text{sun}}$ and $\tan(\theta)$. The polynomial coefficients b_n are obtained using the LSM for data computed from images of three selected days:

$$b_n = \{0.13, -0.49, 0.46, 0.65, -0.0031\} \quad (4)$$

An example of the projected image obtained by this process is shown in Fig. 2b.

In the second step, pairs of consecutive images are processed using Particle Image Velocimetry (PIV) to compute the flow direction of the clouds. The PIV algorithm (developed by Mori and Chang (2003)) partitions the image into interrogation windows (32×32 pixels). The correlation between two consecutive interrogation windows is analyzed through the Minimum Quadratic Difference method (Mori and Chang, 2003) to determine the displacement. The velocities are obtained by dividing the displacements by the separation time between the pair of images resulting in a velocity field as exemplified in Fig. 2b. To characterize the mean cloud motion, we cluster velocity vectors using

the k -means clustering method. The velocity magnitudes usually cluster around two means: one near or equal to zero and the other with larger magnitude. The first results from windows where the clouds are absent or stationary and is therefore disregarded. The second is selected as the representative velocity for the cloud motion.

In the third step, a hybrid cloud classification algorithm (Li et al., 2011) that uses an adaptive threshold scheme is used to identify clouds. As mentioned above, the TSI 880 has built-in image processing software that performs tasks such as cloud classification and SP calculation. However, we concluded that the cloud classification from the TSI returns poor results and lacks flexibility. For this reason we coded our own cloud classification procedure based on the hybrid cloud classification method proposed by Li et al. (2011). This method first transfers each pixel of the image into a normalized red to blue ratio λ :

$$\lambda = (r - b)/(r + b) \quad (5)$$

where r is the red pixel intensity and b is the blue pixel intensity. Then the method identifies whether the histogram of λ is uni-modal or bi-modal. A fixed threshold cloud classification is applied to the uni-modal cases. The fixed threshold is set to 0.275, which is statistically estimated from twenty manually selected images. For the bi-modal case, the minimum cross entropy (MCE) threshold cloud classification is applied. MCE selects the threshold as the minimizer of cross entropy between the image and its segmented image. Further details regarding the MCE threshold can be found in (Li and Lee, 1993; Li and Tam, 1998; Li et al., 2011). An example of the cloud decision obtained by this method can be seen in Fig. 2c.

In the final step, a set of grid elements (X_1, X_2, \dots, X_6) originating from the sun's position and oriented in the reverse direction of the clouds' representative velocity is placed over the images (shown in Fig. 2c). In this work we set the number of grid elements to six, each with an area of 20×20 pixels. Then we compute the cloud indices CI_i as the percentage of pixels classified as cloud for element i . A limitation of using a TSI for solar forecasting is that the shadowband hides part of the sky (see Fig. 2). Therefore, part of a grid element hidden by the shadowband mask or

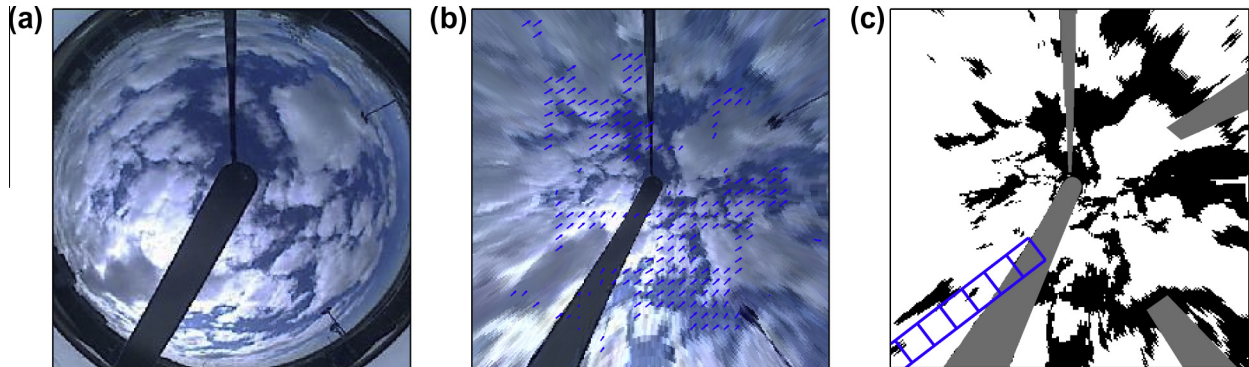


Fig. 2. Main steps of image processing. (a) The original image. (b) The image projected onto a rectangular grid, and the velocity vectors computed by the PIV algorithm. (c) The cloud classified image, where white pixels represent cloud, black pixels represent clear-sky, and gray pixels are obstacle which are excluded from the analysis.

parts that lie outside of the image are not taken into account in the cloud coverage calculations. This process was applied to all the images under study, resulting in six CI_i time-series. These time-series were then synchronized with the DNI data and used as inputs for the forecasting models.

One of the major problems in this algorithm concerns the cloud classification at periods of high air mass (when sun elevation is low). During these periods, higher scattering and diffusion of the solar beam is recorded in the TSI image as a large set of bright pixels in the circumsolar region. These pixels are often classified as clouds even in the case of completely clear sky. This adverse effect is significantly reduced when sun elevation increases. Therefore, to avoid the incorrect behavior of the cloud classification algorithm, we excluded data points whose solar zenith angle θ satisfied $\cos(\theta) < 0.6$.

3.4. Deterministic models

Marquez and Coimbra (2013a) proposed a straightforward linear relation between each CI_i time-series and forecast values of DNI:

$$\hat{B}_i(t + \Delta t) = 0.9 (1 - CI_i) \quad (6)$$

where the factor 0.9 kW/m^2 is an empirical constant for peak DNI for the period studied in that paper. The obvious caveat for this model is that it only works well for times around the solar noon. To overcome this issue we used a more generalized relation:

$$\hat{B}_i(t + \Delta t) = B_{clr}(t + \Delta t) (1 - CI_i) P_o \quad (7)$$

where $\hat{B}_i(t + \Delta t)$ is the predicted DNI for the forecast horizon Δt using the cloud index CI_i . B_{clr} is the clear-sky DNI computed by Eq. (1). P_o is the percentage of DNI decrease caused by clouds. For example, P_o is equal to one for opaque clouds that block solar beam completely. Since the output of the cloud classification algorithm is a binary cloud map (opaque cloud/clear-sky), we assume that all the detected clouds are opaque and P_o is set to one all the time.

Given that there are 6 CI time-series, Eqs. (6) and (7) return 6 values (one for each CI). This creates a decision problem, which is resolved by computing the forecast error for the learning periods (HV_I and LV_I) and choosing the model with the lowest error. This analysis revealed that the best deterministic forecast is obtained with CI_1 and CI_3 for the 5 and 10 min forecasts, respectively.

3.5. ANN-based forecasting model

Artificial Neural Network (ANN) is an Artificial Intelligence method widely used in the forecasting and modeling of solar irradiance (Mellit and Kalogirou, 2008). The key elements of the neural network are the neurons, which are interconnected and placed in layers. The layers between the input layer (the first) and the output layer (the last) are called hidden layers. In the case of feed forward ANNs

(like the ones used here) only forward connections are allowed. Neurons on hidden layers sum the weighted inputs received from the previous layer and add a bias or threshold to the sums. For each neuron, an activation function (a nonlinear transformation like the Sigmoidal function) is applied to the sum and generates the neuron's output. This output is then used as the input for neurons in the following layer.

Once the ANN architecture is determined (the number of layers and neurons) there are several undetermined free-parameters. These parameters correspond to the weight and bias values in each neuron and they are found by training the ANN. As an universal approximating function (Marquez and Coimbra, 2011; Bishop, 1994), ANNs must be trained with relevant data in order to determine the proper weights and biases. In the present work, ANNs were trained with the Bayesian regularization process that uses the Levenberg–Marquardt optimization to minimize the error between the ANN's output and the targets by adjusting the weights and biases. This is an iterative process that stops once discrepancy is lower than a preset value. Alternative training processes are discussed in the literature (Haykin, 2008; Lakhmi and Martin, 1998). The ANN training process for a given ANN architecture is defined symbolically as:

$$(ANN_u(L, N), I, T) \xrightarrow{\text{Training}} ANN_B \quad (8)$$

where ANN_u represents the untrained feed-forward neural network with L hidden layers and N neurons per layer. The inputs I and targets T are the data in the training process. And the forecasting model resulting from the ANN is:

$$\hat{B}(t + \Delta t) = ANN_B(I_n(t)) \quad (9)$$

where I_n refers to new inputs not used in the training phase.

In this study, the input variables I are a subset of the available variables listed in Table 2. The first six variables are the cloud cover indices discussed above, and the remaining variables are the time-lagged values of DNI ranging from 0 to 20 min in steps of 5 min. The target variable T depends on the forecast horizon for which the ANN is being trained:

$$T = \begin{cases} B(t + 5) & \text{for 5 min forecast horizon} \\ B(t + 10) & \text{for 10 min forecast horizon} \end{cases}$$

One of the concerns when training ANNs is the problem of over-fitting. This happens when the solution found in the training optimization process fits the training data very well but leads to an ANN that performs poorly when applied to new data. The issue of over-fitting can be addressed through a validation process. The validation process can be integrated with the optimization tools (discussed in Section 3.6) to find out the optimal definition (architecture, inputs and training) for the ANN forecasting model. In this work we employed two validation methods: cross validation (CVM) (Kohavi, 1995; Jain et al., 2000; Geisser, 1993; Lendasse et al., 2003; Efron and Gong,

Table 2

Variables available as ANN inputs: B is the measured DNI time-series and CI is the cloud coverage time-series. The index of CI is the number of units on the ladder, where 1 is closest to the sun and 6 is most distant from the sun. B at time t and 4 lagged values of B were also considered.

Index	Cloud cover indices						Irradiance data				
	1	2	3	4	5	6	7	8	9	10	11
Time-series	$CI_1(t)$	$CI_2(t)$	$CI_3(t)$	$CI_4(t)$	$CI_5(t)$	$CI_6(t)$	$B(t)$	$B(t-5)$	$B(t-10)$	$B(t-15)$	$B(t-20)$

1983), and randomized training and validation (RTM). CVM is a popular method used in model estimation to determine whether a hypothesis model fits the data of interest. On the other hand, RTM is a new training process proposed in this work. As far as we know, RTM has not been used in other works.

CVM divides the learning data (LV_i or HV_i depending on the season) into K disjoint subsets S_i with $i = 1, \dots, K$. For each subset S_i , there is a corresponding subset of inputs I_i and targets T_i . The K in this study is set to ten as suggested in previous studies (Kohavi, 1995; McLachlan et al., 2005). After partitioning the data, one subset is reserved for validation and the remaining are used to train the ANN. This process is repeated K times, each time taking a different subset as the validation set. For the i th run:

$$(ANN_u(L, N), \bar{I}_i, \bar{T}_i) \xrightarrow{\text{Training}} \overline{ANN}_i \quad (10)$$

where \overline{ANN}_i is the $ANN_u(L, N)$ trained on the subsets S_j , $j = \{1, \dots, K\} | j \neq i$. The training yields the forecasting model:

$$\hat{B}_i = \overline{ANN}_{Bi}(I_i) \quad (11)$$

and the root mean square error (RMSE) for the i th validation run is computed by comparing the predictions \hat{B}_i for the i th input subset I_i against the targets T_i . The performance for the ten CVM validation runs is taken as the average and variance of the K RMSEs. A good model should have a low average and small variance for the RMSE values. High RMSE variance identifies models whose performance is very dependent on the training data and therefore prone to over-fitting issues.

RTM is a learning process which selectively chooses part of the learning period for training. It is based on the premise that not all elements of the available data are equally informative for ANN training (e.g. outliers or repeated data points). Moreover, this strategy searches for the optimal composition of the training sets by means of some optimization algorithm, in this case is the GA (as explained in Section 3.6). The RTM includes five steps:

1. the learning period (LV_i or HV_i depending on the season) is divided into separated training and validation sets (70% for training, the remaining for validation);
2. the training set is further divided into 100 divisions of equal size;
3. several divisions of the training set are selected randomly to train the ANN;

4. the trained ANN is validated on randomly selected subsets (between 5% and 100%) of the predetermined validation set;
5. the validation process is conducted K times (set to be 10 in this study) and the average and variance of the 10 RMSEs are computed and used to assess the model's performance.

3.6. Genetic algorithm (GA) for model optimization

We considered 11 variables (Table 2) as possible inputs for the ANN-based forecasting model. Together with the possible number of hidden layers L and neurons per hidden layer N , training and validating all possible models to identify the best one is computationally prohibitive, therefore we employed the GA to determine the optimal model.

The genetic algorithm (GA) is inspired by the idea of natural selection and survival of the fittest. First introduced by Holland (1992) GAs are widely used as an efficient optimization tool (Mellit and Kalogirou, 2008; Reeves and Rowe, 2002; Pai and Hong, 2005; Marquez and Coimbra, 2011; Pedro and Coimbra, 2012). The GA is an iterative process that scans elements of the search space (the individuals) in order to find the global optimum. The iterative process is driven by the fitness of each individual. In general, the GA starts with a random set of individuals that cover the search space uniformly. Each individual is classified according the objective function, and the best are selected as parents for the following generation individuals which are created through cross-over and mutation. The iterative process continues until the fitness of the best individual converges with the average fitness for the entire population.

The GA optimization objective function in this study computes the models' performance by means of the validation methods (CVM or RTM) explained above. For instance, when GA is implemented with CVM, each individual (ANN model with specific topology) is cross-validated on the learning data through CVM, and the validation performance is calculated as the fitness. The GA was applied with both validation methods and the optimal resulting models $ANN[CVM]$ and $ANN[RTM]$ explicitly identify which method they refer to.

In this study, the GA chromosome length is fixed. The chromosome for $ANN[CVM]$ optimization contains 13 elements. The first 11 are binary and decide whether an input vector is selected or not as an ANN input (see in Table 2). The last two are integers, the first represents the number of layers ($L = \{1, 2, 3, 4\}$), and the second represents the

number of neurons ($N = \{1, \dots, 20\}$). The limits for L and N are related to computational cost. However the optimal chromosomes returned by GA always have $L < 3$ and $N < 5$ (see Table 4). These values are not close to the search space limits, which reassures us that these limits are appropriate. The chromosome for ANN[RTM] optimization contains 113 elements. The first 13 are the same as those discussed above. The rest are binary that decide whether a division of the training set is used to train the ANN or not.

The GA optimizations are submitted as parallel jobs to the San Diego Supercomputer Center (SDSC) Gordon Supercomputer. One generation's computation takes 16 cores working for about 1 h. Each optimization job (total 50 generations) takes about 2 days to complete. Once the training is performed, new forecasts are produced by regular PCs with minimal computational cost.

3.7. Smart forecasting model

Once persistence, deterministic and ANN-based forecasting models were determined for both periods (high and low DNI variability) we defined a smart forecasting model that uses the best model from each period to create a season-independent model. The smart forecasting model is defined by:

$$\hat{B}(t + \Delta T) = \begin{cases} Model_{LV}(t) & \text{if } \sum_i CI_i = 0 \\ Model_{HV}(t) & \text{otherwise} \end{cases} \quad (12)$$

where $Model_{LV}$ is the best forecasting model for the low DNI variance season and $Model_{HV}$ is the best forecasting model for the high DNI variance season. Eq. (12) is a piece-wise function where cloud cover indices determine which forecasting model is used: when the sum of all CI_i indices is zero (no clouds present in the grid elements X_i) it applies $Model_{LV}$, otherwise it applies $Model_{HV}$. The threshold in Eq. (12) is fixed. When all $CI_i = 0$, we assume there are no clouds moving towards the sun. Consequently, we can expect low DNI variance in the next forecast horizon.

3.8. Testing process

All the models were tested with the independent data set IT defined in Table 1. Testing is necessary to demonstrate how models perform when deployed in real-time where they encounter new input data. For this evaluation, the persistence model (Eq. (2)) was used as a benchmark for all other models.

Three statistical metrics for the models' error were used to assess their performance:

- mean bias error (MBE), the mean of all the errors, which indicates the systematic bias of the model

$$MBE = \frac{1}{n} \sum_{j=1}^n (\hat{B}_j - B_j) \quad (13)$$

- RMSE, a frequently used measure of difference between model prediction and actual observation;

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (\hat{B}_j - B_j)^2} \quad (14)$$

- forecast skill (s), which represents the improvement of forecast accuracy against the reference persistence model:

$$s = \begin{cases} 1 - \frac{RMSE}{RMSE_p} & \text{if } RMSE < RMSE_p \\ \frac{RMSE_p}{RMSE} - 1 & \text{if } RMSE > RMSE_p \end{cases} \quad (15)$$

where $RMSE_p$ is the forecast RMSE of the persistence model on the testing set.

Three different data sets were used for this procedure: LV_t , HV_t and IT . We first tested the models for low and high DNI variability, LV_t and HV_t respectively. Based on those results we defined the best seasonal models $Model_{LV}$ and $Model_{HV}$. These two models and the persistence, deterministic and the smart models were then tested using the independent data set IT . Table 3 lists the testing procedure.

In order to test the ANN-based models properly, additional care is necessary because due to the random initialization of the ANN training process, ANN-based models trained with the same parameters on the same data set return different forecast results for every time they are re-trained. This is caused by the existence of multiple local minima in the ANN training search space. Thus, results from a single training process cannot be treated as a repeatable. To avoid this issue we would have to initialize manually the weight and bias values in the Levenberg–Marquardt optimization which is not feasible. Alternatively, we trained the ANN multiple times (using the default random initialization) and use the average of these runs as the result for the ANN forecasting model. In summary, the results presented below for any ANN-based model were obtained in the following manner:

1. Train the ANN with the model definition optimized by the GA. ANNs developed with CVM are trained throughout the whole learning period while the ANNs developed with RTM are trained on the optimization training set found by the GA.

Table 3

The testing schedule shows which period was used in the testing of the different models. $Model_{LV}$ is the best forecasting model for the LV_t testing set and $Model_{HV}$ is the best forecasting model for the HV_t testing set.

Testing set	LV_t	HV_t	IT
Models	Persistence Deterministic $ANN_{LV}[CMV]$ $ANN_{LV}[RTM]$	Persistence Deterministic $ANN_{HV}[CMV]$ $ANN_{HV}[RTM]$	Persistence Deterministic Smart $Model_{LV}$ $Model_{HV}$

2. Apply the ANN to the testing set and determine the forecast outputs \hat{B} .
3. Repeat steps 1 and 2 thirty times and compute the performance values MBE, RMSE and s acquired for each ANN.
4. Compute mean and standard deviation for the performance values.

Persistence and deterministic models return a single value, so their forecasts do not have corresponding standard deviation.

4. Results and discussion

4.1. ANN model optimization

The optimal set of input variables, hidden layers, and the number of neurons per layer for the ANN based forecasting models are shown in Table 4 for the different forecast horizons, validation methods and seasons. These results were obtained upon convergence of the average fitness of the population with the fitness of the best individual, which indicates that the GA converges to a solution.

These results indicate that for the high DNI variance season, CI values from grid units located closer to the sun are very important for shorter forecast horizons. This agrees with the findings of Marquez and Coimbra (2013a). The same is not observed for the low DNI variance season, when weather conditions are stable and generally there are no clouds in the sky. Table 4 also shows that $B(t)$ (variable number 7), which is the latest available value of DNI at the time of the forecast is always selected regardless of the season or forecast horizon. This is expected given that the last available value of DNI is the best indicator of its future value.

4.2. Seasonal models

As mentioned above, in order to access the performance of a new forecasting model correctly, the model must be applied to data that was not used in the estimation phase. This was done for the seasonal models using the testing sets LV_t and HV_t . The results are listed Table 5.

These results show that the deterministic model outperforms the persistence model for the HV period. However, its performance is inconsistent, and its forecast skill is negative on the LV testing set. Several reasons can explain this:

cloud classification misidentifies clouds due to mirror glazing or forward lobe Mie scattering near the circumsolar region, the model is unable to accurately determine cloud's optical path, and it oversimplifies the relationship between cloud coverage and DNI ramps.

The results for the low DNI variance season suggest that the model $ANN_{LV}[RTM]$ achieves the lowest overall RMSE. In fact, this is the only model that beats the persistence model on this season for both forecast horizons. On the other hand, the model obtained by means of CVM, $ANN_{LV}[CVM]$, is clearly inferior to $ANN_{LV}[RTM]$ with a higher average RMSE (even higher than the persistence RMSE for the 10 min horizon) and higher variance of RMSE.

For the high DNI variance season the results are reversed: the $ANN_{HV}[CVM]$ model shows an average RMSE significantly lower than $ANN_{HV}[RTM]$. In this period both $ANN_{HV}[CVM]$ and $ANN_{HV}[RTM]$ exhibit significant decreases in RMSE with respect to the persistence model regardless of the forecast horizon. The evaluation of forecast skill shows that the ANN forecasting models benefit from integrating the CI time-series and lagged DNI data by clearly outperforming the persistence model with improvements ranging from 5% to 10% for the low variability season and 20% to 25% for the high variability season.

In terms of bias, the persistence model shows the smallest MBE in most cases. However, given that all models exhibit small MBE values we did not consider MBE values when determining the best seasonal models. Thus, base solely on forecast skill, the two models included in the smart forecast were $ANN_{LV}[RTM]$ and $ANN_{HV}[CVM]$.

4.3. Smart forecast

With the smart forecasting model defined from the analysis in the previous section, five models (smart, $ANN_{LV}[RTM]$, $ANN_{HV}[CVM]$, deterministic and persistence) were applied to the independent data-set. The performance metrics for these models are presented in Table 6, with results showing that the smart forecasting model outperforms all other models regardless of the forecast horizon.

Given that there are both low and high DNI variance days in the independent testing set, we studied the daily performance of the forecasting models. Fig. 3 shows daily RMSE for the smart and persistence models for 5 and 10 min horizons. The error level varies greatly from day

Table 4

ANN inputs and parameters selected by the GA for the different seasons, forecast horizons and validation methods.

Season	Methods	5 min forecast			10 min forecast		
		Inputs' indices	L	N	Inputs' indices	L	N
Low DNI	CVM	4 5 7 8 11	2	1	3 4 5 7 8	1	3
	RTM	2 4 5 7 8 9 10 11	2	1	1 3 4 5 6 7 8	2	1
High DNI	CVM	1 4 7 8 9	2	3	2 3 4 5 7 8	1	3
	RTM	1 7	1	4	2 3 4 5 6 7	2	2

Table 5

Forecast performance for the different season-dependent models as a function of the forecast horizon and the validation method. Std represents the standard deviation. Boldface font identifies the best performance values.

Test set	Models	5 min forecast			10 min forecast		
		MBE (Std) [W/m ²]	RMSE (Std) [W/m ²]	s (Std) [%]	MBE (Std) [W/m ²]	RMSE (Std) [W/m ²]	s (Std) [%]
Low DNI variance Season (<i>LV_t</i>)	<i>ANN_{LV}</i> [CVM]	2.4 (4.8)	48.3 (16.5)	5.0 (16.4)	−5.7 (2.3)	65.9 (8.1)	−8.0 (8.2)
	<i>ANN_{LV}</i> [RTM]	0.1 (0.0)	44.8 (0.0)	7.7 (0.0)	−1.9 (0.0)	54.6 (0.5)	9.1 (0.1)
	Persistence	−0.1	48.5	0.0	0.3	60.0	0.0
	Deterministic	−2.1	90.1	−46.2	−37.8	87.7	−31.6
High DNI variance Season (<i>HV_t</i>)	<i>ANN_{HV}</i> [CVM]	−4.1 (0.9)	109.3 (1.9)	25.8 (1.3)	−9.9 (0.5)	132.0 (0.8)	24.7 (0.5)
	<i>ANN_{HV}</i> [RTM]	−5.8 (0.4)	114.4 (1.4)	22.4 (0.9)	−11.7 (1.1)	140.3 (1.8)	20.0 (1.0)
	Persistence	−1.6	147.4	0.0	−3.3	175.4	0.0
	Deterministic	−3.3	142.7	3.2	−5.2	151.8	13.5

Table 6

Forecast performance for the different models applied to the independent testing set (*IT*) as a function of the forecast horizon. Std represents the standard deviation. Boldface font identifies the best performance values.

Models	5 min forecast			10 min forecast		
	MBE (Std) [W/m ²]	RMSE (Std) [W/m ²]	s (Std) [%]	MBE (Std) [W/m ²]	RMSE (Std) [W/m ²]	s (Std) [%]
Smart model	−0.2 (0.1)	88.6 (0.5)	20.9 (0.4)	−1.9 (0.4)	103.3 (0.6)	22.5 (0.4)
<i>ANN_{LV}</i> [RM]	−14.1 (0.0)	104.8 (0.1)	6.5 (0.0)	34.6 (0.1)	141.3 (0.1)	−5.6 (0.0)
<i>ANN_{HV}</i> [CVM]	−5.5 (0.1)	90.7 (0.6)	19.0 (0.6)	−7.5 (0.6)	103.9 (0.7)	22.1 (0.5)
Deterministic	−10.7	150.5	−25.6	−8.3	144.0	−7.4
Persistence	2.1	112.0	0.0	3.6	133.4	0.0

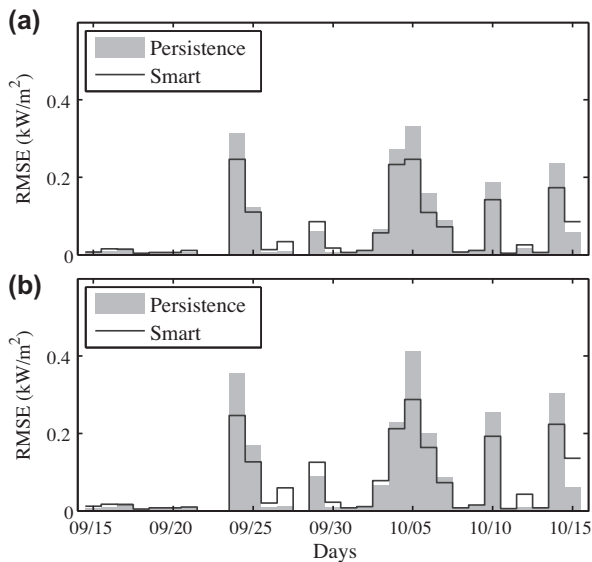


Fig. 3. Plot of daily RMSE of the smart and persistence forecasts. Each bar stands for one testing day. (a) The 5 min forecast RMSE of the smart and persistence models. (b) The 10 min forecast RMSE of the smart and persistence models.

to day depending on the weather conditions. An alternative way of analyzing these results is by plotting the daily

RMSE of the smart models against the daily $RMSE_p$ of the persistence models. The resulting scatter plot is depicted in Fig. 4 for both 5 and 10 min forecast horizons. Given that the daily RMSE for the persistence model is a good indicator of each day's DNI variance level, the x-axis was used to distinguish between high and low variability days (the limits for each period are indicated by the vertical dashed lines in Fig. 4). For days with low variability ($RMSE_p < 0.05 \text{ kW/m}^2$), the smart forecasting model does not beat the persistence model consistently (the points are above the identity line). However, the smart forecast error on those days is very low.

The analysis of the daily RMSE (Figs. 4 and 5) indicates that the forecast performance of models varies under different meteorological conditions. To quantitatively study the forecast performance under different levels of irradiance variability, we introduce the irradiance variability V , which is defined as the standard deviation of the step changes of the clear sky index k_t (Marquez and Coimbra, 2013b):

$$V = \sqrt{\frac{1}{N} \sum_{t=fh}^t (k_t - k_{t-1})^2} \quad (16)$$

where fh is the forecast horizon, and the time step is 1 min.

During similar V periods, the statistical quantities of forecast evaluation are comparable, and the ratio of

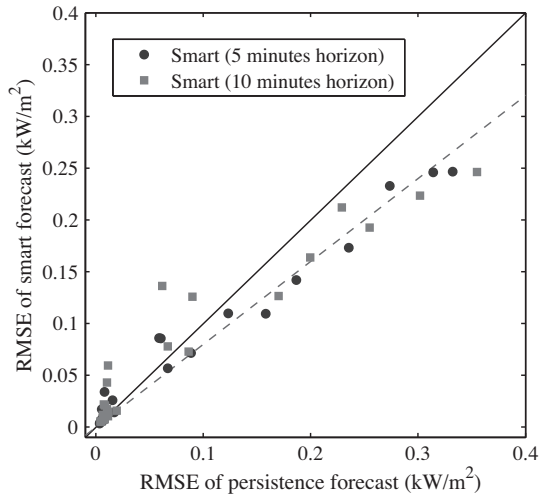


Fig. 4. Plot of daily RMSE of the smart forecast against the persistent forecast. Each marker stands for one testing day. The solid line with slope 1 suggests an equal performance for the smart and persistence forecasts. Points below the dash line with slope 0.8 represent days in which the forecast skill of the smart model exceeds 20%.

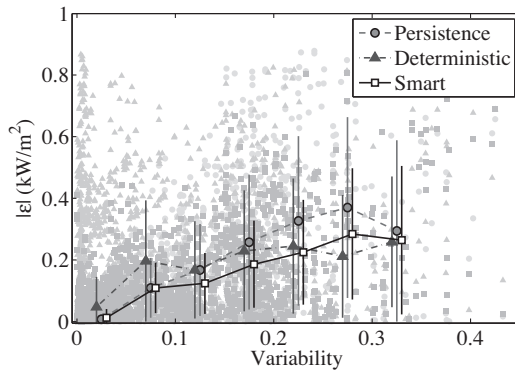


Fig. 5. Plot of absolute forecast error ($|\varepsilon|$) versus variability V for 5-min-ahead forecasts. Round markers represent the persistence forecast, triangular markers represent the deterministic forecast, and square markers represent smart forecast. The heavier markers are the mean $|\varepsilon|$ taken within the variability bin with width of 0.05. The length of each error bar represents for one standard deviation of all data within each bin.

absolute forecast errors ($|\varepsilon|$) to V is a statistical invariant for different forecasting models (Marquez and Coimbra, 2013b). This ratio for persistence, deterministic and smart forecasts are plotted in Figs. 5 and 6 for 5 and 10 min ahead forecasts respectively. The figures show that the persistence model is most accurate during low V periods ($V < 0.1$) with absolute persistence forecast error $|\varepsilon_p|$ close to zero. During these periods, the DNI time-series is very stable and ideal for persistence forecasts. However, the performance of the persistence model degrades considerably when V increases.

The deterministic model shows superior performance for very high V periods for both 5 and 10 min forecasts. Nevertheless, its performance is inconsistent resulting in high absolute forecast errors ($|\varepsilon_d|$) for low V periods. This matches our previous discussion in Section 4.2.

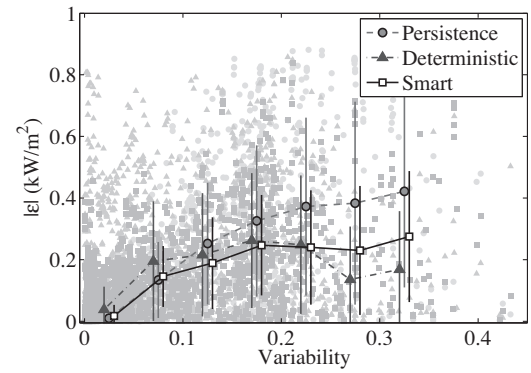


Fig. 6. Plot of absolute forecast error ($|\varepsilon|$) versus variability V for 10-min-ahead forecasts. Round markers represent the persistence forecast, triangular markers represent the deterministic forecast, and square markers represent smart forecast. The heavier markers are the mean $|\varepsilon|$ taken within the variability bin with width of 0.05. The length of each error bar represents for one standard deviation of all data within each bin.

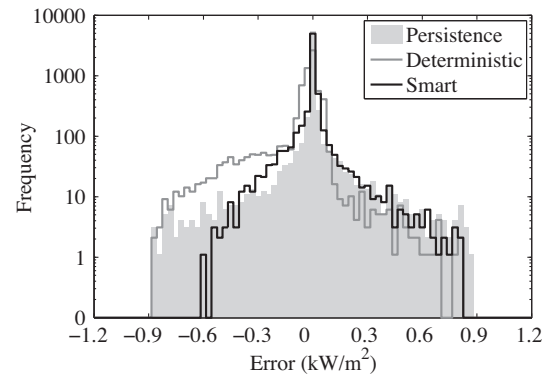


Fig. 7. Plots of error (ε) distribution for 5 min ahead forecasts.

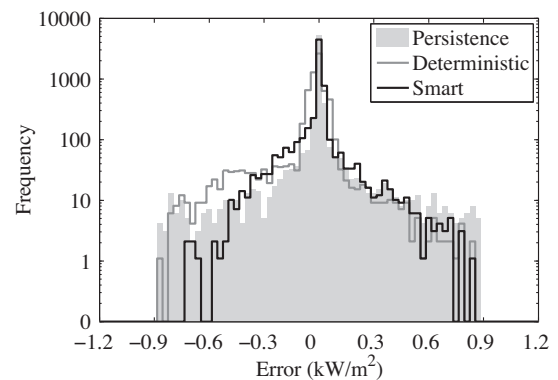


Fig. 8. Plots of error (ε) distribution for 10 min ahead forecasts.

The smart forecast, which considers the input variables of both persistence and deterministic models, balances the distinct behaviors of both forecasts and generates the most consistent forecast for both horizons. The smart forecast is most accurate during intermediate variability conditions ($0.10 < V < 0.25$). Moreover, its performance is nearly equivalent to the persistence forecast during low V

periods, but inferior to the deterministic forecast during high V periods.

To further understand the models' performances, their error distributions are studied (Figs. 7 and 8). The standard deviation, Pearson's skewness, and kurtosis (Kenny, 1961; Joanes and Gill, 1998) for each distribution are obtained through statistical analysis and listed in Table 7. Skewness, which quantifies the level of the distribution's bias, is defined as the difference between the mean and mode divided by the standard deviation. For example, positive skewness indicates that the right tail is longer or fatter than the left tail. Kurtosis, which quantifies the "peakedness" of a distribution and heaviness of its tail, is defined as the distribution's fourth central moment divided by the fourth power of its standard deviation. For example, a high kurtosis distribution has a sharper peak and longer, heavier tails.

The distribution of ε_p leans slightly to the right side, resulting in positive skewness. It has a moderate level of deviation but the highest kurtosis, indicating it has the sharpest peak. This is the result of excellent performance (ε_p is close to zero) of the persistence forecast during low V periods, which comprises more than 80% of the whole testing period. This matches the assumption that the persistence forecast performs best during low V periods. However, persistence forecast errors increase rapidly as V increases, resulting in a relatively high standard deviation for the ε_p distribution.

Contrary to the persistence forecast, the lowest kurtosis of the ε_d distribution indicates a lower and rounder peak. This reflects the less satisfactory performance of the deterministic model during low V periods. The ε_d distribution shows a negative skewness with highest absolute value among all distributions. This validates the hypothesis that misidentified clouds and erroneous optical path degrades the performance of the deterministic forecast. Both factors cause the predictions of non-existing ramps and the overestimation of ramp rates, resulting in a higher occurrence of negative ε_d . Therefore, ε_d distribution shows a heavy left tail and the highest standard deviation. The RMSE for 10 min ahead forecast is smaller than RMSE for the 5 min ahead forecast. This can be explained by the fact that 10 min ahead forecast relies mostly on grid units farther away from the sun. As a result, this forecast is less vulnerable to mirror glaring and forward scattering in the circum-solar region. Consequently, the 10 min forecast for ε_d has a lower standard deviation, less negative skewness, and higher kurtosis than its 5 min counterpart.

The smart forecast achieves the lowest standard deviation and skewness for the smart forecast error (ε_s) distribution of both 5 and 10 min forecasts, showing that although the smart forecast uses the same inputs as the deterministic model, it is able to identify and reduce the forecast bias. Because the smart forecast achieves lower $|\varepsilon_s|$ during high V periods than the persistence forecast, thereby reducing the occurrence of high absolute value errors with an increase of occurrence of moderate sized errors. This results in a shorter tail and rounder shoulders for the ε_s distribution. In addition, the smart forecast has nearly equivalent performance as the persistence forecast during low V periods, so the ε_s distribution has a similarly sharp peak to the ε_p distribution. Consequently, the kurtosis of the ε_s distribution is lower than the ε_p distribution but still remarkably higher than the ε_d distribution. The minimization of the occurrence of the ε_s with high absolute value, especially on the left tail, narrows the distribution and lowers the standard deviation of ε_s .

5. Conclusions

We presented several new models for intra-hour DNI forecast. ANN-optimized models, which were based on cloud cover indices extracted from TSI images as inputs, were first developed for low and high variability DNI seasons. A genetic algorithm (GA) was employed to obtain the optimal ANN-based forecasting models for each season. Two validation schemes, the Cross-Validation Method (CVM) and a new, Randomized Training Method (RTM), were used to ensure that the ANN models were not over-fitted. The methods proposed here were developed and applied to over 6 months of 30 s diurnal sampling data, which included co-located sky imaging and high-quality DNI measurements.

By hybridizing the two best ANN models for each season with an adaptive threshold scheme, a season-independent smart forecasting model was developed. Application of the hybrid model to the independent testing set shows that the proposed smart forecasting methodology achieves the lowest mean absolute error during moderate variability periods, and its error distribution has the lowest bias, lowest standard deviation, and lowest occurrence of high absolute value errors among all forecasting models investigated. The combination of all these effects resulted in robust forecasting skill above the 20% level over persistence for 5 and 10 min forecast horizons.

Table 7
Statistical results of the analysis for the error distributions.

Distributions	5 min forecast			10 min forecast		
	Persistence	Deterministic	Smart	Persistence	Deterministic	Smart
Standard deviation (kw/m ²)	0.112	0.142	0.088	0.133	0.136	0.103
Pearson's skewness	0.019	−0.326	−0.002	0.027	−0.201	−0.018
Kurtosis	28.175	12.497	24.740	22.482	14.824	18.222

Acknowledgements

The authors gratefully acknowledge funding from the California Solar Initiative (CSI) Research, Development, Demonstration, and Deployment (RD&D) Program Grant III; and from the National Science Foundation (NSF) EECS-EPAS award No. 1201986, which is managed by Dr. Paul Werbos.

References

- Bishop, C.M., 1994. Neural networks and their applications. *Review of Scientific Instruments* 65, 1803–1832.
- Cao, J., Lin, X., 2008. Study of hourly and daily solar irradiation forecast using diagonal recurrent wavelet neural networks. *Energy Conversion and Management* 49, 1396–1406.
- Chow, C.W., Urquhart, B., Lave, M., Dominguez, A., Kleissl, J., Shields, J., Washom, B., 2011. Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. *Solar Energy* 85, 2881–2893.
- Efron, B., Gong, G., 1983. A leisurely look at the bootstrap, the jackknife, and cross-validation. *The American Statistician* 37, 36–48.
- Elizondo, D., Hoogenboom, G., McClendon, R.W., 1994. Development of a neural network model to predict daily solar radiation. *Agricultural and Forest Meteorology* 71, 115–132.
- Geisser, S., 1993. In: *Predictive Inference: An Introduction*, vol. 55. CRC press, New York, NY.
- Hammer, A., Heinemann, D., Lorenz, E., Lucke, B., 1999. Short-term forecasting of solar radiation: a statistical approach using satellite data. *Solar Energy* 67, 139–150.
- Haykin, S., 2008. *Neural Networks: A Comprehensive Foundation*, third ed. Pearson Prentice Hall, Englewood Cliffs, NJ.
- Holland, J.H., 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press.
- Jain, A.K., Duin, R.P.W., Mao, J., 2000. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 4–37.
- Joanes, D.N., Gill, C.A., 1998. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)* 47, 183–189.
- Kenny, J.F., 1961. *Mathematics of Statistics*. Nostrand.
- Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference on Artificial Intelligence*. Lawrence Erlbaum Associates Ltd, pp. 1137–1145.
- Lakhmi, C.J., Martin, N.M., 1998. *Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications*. CRC Press, Boca Raton, FL.
- Lendasse, A., Wertz, V., Verleysen, M., 2003. Model selection with cross-validations and bootstraps-application to time series prediction with RBFN models. *Artificial Neural Networks and Neural Information Processing ICANN/ICONIP* 2714, 573–580.
- Li, C.H., Lee, C.K., 1993. Minimum cross entropy thresholding. *Pattern Recognition* 26, 617–625.
- Li, C.H., Tam, P.K.S., 1998. An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognition Letters* 19, 771–776.
- Li, Q., Lu, W., Yang, J., 2011. A hybrid thresholding algorithm for cloud detection on ground-based color images. *Journal of Atmospheric and Oceanic Technology* 28, 1286–1296.
- López, G., Batlles, F.J., Tovar-Pescador, J., 2005. Selection of input parameters to model direct solar irradiance by using artificial neural networks. *Energy* 30, 1675–1684.
- Marquez, R., Coimbra, C.F.M., 2011. Forecasting of global and direct solar irradiance using stochastic learning methods, ground experiments and the NWS database. *Solar Energy* 85, 746–756.
- Marquez, R., Coimbra, C.F.M., 2013a. Intra-hour DNI forecasting methodology based on cloud tracking image analysis. *Solar Energy* 91, 327–336.
- Marquez, R., Coimbra, C.F.M., 2013b. Proposed metric for evaluation of solar forecasting models. *ASME Journal of Solar Energy Engineering* 135, 0110161–0110169.
- Marquez, R., Gueorguiev, V.G., Coimbra, C.F.M., 2013a. Forecasting of global horizontal irradiance using sky cover indices. *ASME Journal of Solar Energy Engineering* 135, 0110171–0110175.
- Marquez, R., Pedro, H.T.C., Coimbra, C.F.M., 2013b. Hybrid solar forecasting method uses satellite imaging and ground telemetry as inputs to ANNs. *Solar Energy* 92, 176–188.
- Martín, L., Zarzalejo, L.F., Polo, J., Navarro, A., Marchante, R., Cony, M., 2010. Prediction of global solar irradiance based on time series analysis: application to solar thermal power plants energy production planning. *Solar Energy* 84, 1772–1781.
- McLachlan, G.J., Do, K.A., Ambrose, C., 2005. In: *Analyzing Microarray Gene Expression Data*, vol. 422. Wiley-Interscience, Hoboken, NJ.
- Mellit, A., Eleuch, H., Bengham, M., Elaoun, C., Pavan, A.M., 2010. An adaptive model for predicting of global, direct and diffuse hourly solar irradiance. *Energy Conversion and Management* 51, 771–782.
- Mellit, A., Kalogiros, S.A., 2008. Artificial intelligence techniques for photovoltaic applications: a review. *Progress in Energy and Combustion Science* 34, 574–632.
- Mellit, A., Pavan, A.M., 2010. A 24-h forecast of solar irradiance using artificial neural network: application for performance prediction of a grid-connected pv plant at Trieste, Italy. *Solar Energy* 84, 807–821.
- Mori, N., Chang, K.A., 2003. Introduction to Mpiv. User manual and program. <<http://sauron.civil.eng.osaka-cu.ac.jp/~mori/softwares/mpiv>>.
- Pai, P.F., Hong, W.C., 2005. Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. *Electric Power Systems Research* 74, 417–425.
- Pedro, H.T.C., Coimbra, C.F.M., 2012. Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy* 86, 2017–2028.
- Perez, R., Kivalov, S., Schlemmer, J., Hemker, K., Renne, D., Hoff, T.E., 2010. Validation of short and medium term operational solar radiation forecasts in the US. *Solar Energy* 84, 2161–2172.
- Reeves, C.R., Rowe, J.E., 2002. In: *Genetic Algorithms-Principles and Perspectives: A Guide to GA Theory*, vol. 20. Kluwer Academic Publishers, Boston, MA.
- Ruano, A.E., Crispim, E.M., Ferreira, P.M., 2008. Prediction of the solar radiation evolution using computational intelligence techniques and cloudiness indices. *International Journal of Innovative Computing, Information and Control* 4, 1121–1133.
- Sfetsos, A., Coonick, A.H., 2000. Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques. *Solar Energy* 68, 169–178.