

Models for solar radiation prediction based on different measurement sites

Massimo Lazzaroni ^{a,b,*}, Stefano Ferrari ^{c,*}, Vincenzo Piuri ^c, Ayşe Salman ^d, Loredana Cristaldi ^e, Marco Faifer ^e

^a *Department of Physics, Università degli Studi di Milano, Milan, Via Celoria, 16, 20133 Milano, Italy*

^b *I.N.F.N. Milano, Via Celoria, 16, 20133 Milan, Italy*

^c *Department of Computer Science, Università degli Studi di Milano, Via Bramante, 65, 26013 Crema, Italy*

^d *Department of Computer Engineering, Doğuş University, Istanbul, Acibadem, Kadikoy, 34722 Istanbul, Turkey*

^e *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci, 133, 20133 Milan, Italy*

Received 14 October 2013

Received in revised form 5 November 2014

Accepted 10 November 2014

Available online 17 December 2014

1. Introduction

The electric power system is mainly composed of units for energy production i.e. generators, loads and a power

grid that connects them. Actual configuration principally includes large central generators which, through the transformers, inject electrical power in the transmission grid. The world energy infrastructure is nowadays subjected to an important transformation such as the growing number of distributed small generation units, based on different technologies, directly connected to the power grid. These small generation units put side by side to the large and traditional ones are defining a grid based on the so called distributed generation. This kind of network architecture

* Corresponding authors at: Department of Physics, Università degli Studi di Milano, Milan, Via Celoria, 16, 20133 Milano, Italy (M. Lazzaroni).

E-mail addresses: massimo.lazzaroni@unimi.it (M. Lazzaroni), stefano.ferrari@unimi.it (S. Ferrari), vincenzo.piuri@unimi.it (V. Piuri), asalman@dogus.edu.tr (A. Salman), loredana.cristaldi@polimi.it (L. Cristaldi), marco.faifer@polimi.it (M. Faifer).

implies new problems concerning the management. In fact in traditional network the stability of the power system was achieved by means of the direct control of few large conventional power generators. By introducing distributed generation this approach cannot be followed, since the small generation units are basically not controllable by the network system operator. In particular this scenario is critical when units based on renewable energy resources are used, since they can only provide power as long as the source of energy is available. In many situation the energy production is mainly utilized directly by the producer or by nearby buildings. When energy production exceed the necessity the excess flows into the power grid of the utilities. In order to implement an electric grid allowing a large amount of distributed energy sources, a different approach to the problem of the network stability is required. It is clear that in this scenario the possibility to predict the plant's power production during the day greatly helps the management of such a power system. Besides, the prediction of energy production becomes extremely important for the trading in the free energy market.

Among the renewable energy sources, the photovoltaic (PV) technology, which allows to obtain electric energy from solar radiation [1], has the benefit of the low environmental impact. On the other end, its main weakness is that its availability cannot be fully controlled. Many aspects need to be considered such as geographic position, local climate, weather and global efficiency of the panel [2–12]. Among these, the position and the climate influence on the solar radiation can be easily obtained from astronomical and statistical data, but the weather is characterized by a high variability and depends on many physical factors. According to [13], the forecasts required by the activity related to the grid management can be divided in two categories. The first is related to grid stability problem (intra-hour, hour ahead, and day ahead), while the second concerns planning and assets optimization on medium and long-term (monthly and yearly forecasts, respectively). Since the main factor for solar radiation availability is the local weather, approaches based on weather forecast have been widely used in literature. These are based on data obtained from satellite observations and ground stations. The geographic and time availability of data are the main aspects that have to be taken into account. Besides, the sampling rate of the measurement (both in time and in space) have to be related to the granularity of the forecast.

The solar radiation prediction can be based on data obtained by several data sources, characterized by the type of data they produce, as well the space-time granularity they provide. These data source are, for example: Numerical Weather Prediction models, Satellite-base forecast, All-sky imagers, Ground measurements.

In this paper we consider the problem of the solar radiation prediction from data captured by ground weather stations, in two different operational conditions: the local (when the data are captured at the plant site) and the remote prediction (when the weather station is distant from the plant site).

Several forecasting approaches have been used in literature. Among these, the most effective in producing hour-ahead predictions are based on empirical regression,

neural networks [14,15] and time-series models (e.g., ARMA, ARIMA, Non-Linear autoregressive models) [16–18]. In particular, in [14] several prediction techniques have been challenges to forecast the daily solar radiation in Algeria using the historical data in the period from 1981 to 2001, and a wavelet-neural network resulted the best. In [15] a good review of the literature on the use neural networks for solar radiation prediction is presented. It highlight how the different approaches differ for the type of solar radiation predicted, for the input parameters and for the prediction error used. In [16] the historical data of the period 2003–2009 for the solar radiation in Bogot (Colombia) have been used to determine an autoregressive model in order to provide a long period forecast useful for assessing the productivity of solar plants. A similar study have been conducted in [17], where the autoregressive models of the solar radiation of several Nigerian cities have been built using the 1986–1990 data. Instead in [18] a Non-linear Autoregressive model is used to model the solar radiation in Edinburgh (UK) using 30 s averages of radiation measured for 8 h per day for one month.

For large plants, the investment to provide the plant with the instrumentation (and the management) for on-site weather measurement is appropriate, since it can provide data for both a reliable production forecast and an effective maintenance scheduling [19–22,11,12]. For small plants, instead, this cost is questionable and the exploitation of data coming from a remote public weather station can be appealing. In fact, this plants are mainly intended for fulfilling the local needs, and only the excess of production is sent to the grid. Hence some requirements on the prediction accuracy can be relaxed, since only the integral of the contribution of the small plants is important for the grid management. On the other hand, there is a clear interest for a reliable scheduling of the maintenance that guarantees the efficiency of the plant as far as a system with an high level of Dependability features, i.e. Reliability, Availability, Maintainability, Safety (RAMS requirements) [23–28].

In our previous works [29–31], several models (namely, Autoregressive models, Support Vector Machine, and Extreme Learning Machine) have been challenged in the task of predicting the global horizontal illuminance. In the present work, instead, another physical characteristics – the global horizontal radiation – will be considered and two experiments, already introduced in [32,33], will be described. The first experiment consists in a short term prediction of the solar radiation using only data measured in the near past. For this scope, a two-year hourly dataset of the global horizontal radiation will be used to feed some autoregressive models in order to obtain a one-hour forecast. In particular, the dataset has been collected in two years by the MeteoLab [34,35], Milan, Italy. In the second experiment, the same prediction will be performed using data of a different set-up. Data from two public weather stations [36] in the neighborhood of Milano, separated by ten kilometers in a straight line, are used for assessing the effectiveness of short term prediction of the production of a plant using measurements taken in a nearby measurement station. In particular, a 3-year hourly dataset will be used to model the time series of the global horizontal

radiation using two computational intelligence models, namely the Support Vector Regression (SVR) and the Extreme Learning Machine (ELM). The performance of the prediction models will be compared with those of a naïve predictor, the persistence model, and of a simple predictive model, namely the k -Nearest Neighbor (k -NN) model.

The paper is organized as follows. In Section 2 the model used in the prediction are described, while the data, the structure, and the results of the experiments are reported in Sections 3 and 4, for the local and the remote prediction, respectively. The results will be compared and discussed in Section 5. Finally, conclusions and future research directions are reported in Section 6.

2. Prediction models

A time series is composed of a sequence of observation $\{x_t\}$ sampled by a sequence of random variables $\{X_t\}$. Usually, the ordering value is related to the time, while the observations are related to a phenomenon that varies with the time and are taken in equally spaced instants.

Autoregressive models are designed to work with this kind of data and the standard algorithm used to estimate the parameters that model the given time series relies on the availability of the elements of the series. Once the model parameters have been estimated, a prediction for the future elements of the series can be computed. For the experiments described in this work, we challenged the most used Autoregressive models described in Section 2.1.

Although not specifically designed for modeling time series, both the SVR and the ELM paradigms (described respectively in Sections 2.2 and 2.3) can also fulfill the time series prediction task. In fact, they can model a mapping between an input and an output space, described by only a finite set of input–output pairs (possibly affected by error), called training set. Time series can hence be modeled as a mapping between some previously observed values and the value to be predicted. For instance, when using two previous samples to predict the next (i.e., looking for a $\mathbb{R}^2 \rightarrow \mathbb{R}$ mapping), the training dataset will be composed by triples of the form (x_{t-2}, x_{t-1}, x_t) , and the mapping, f , $\hat{x}_t = f(x_{t-2}, x_{t-1})$ will be assumed to approximate x_t . Both the two paradigms uses a linear combination of basis functions (usually Gaussians) to model the mapping:

$$f(x) = \sum_{i=1}^L \beta_i G(x; \mu_i, \sigma_i) + b \quad (1)$$

where L is the number of basis functions, G is the Gaussians function, μ_i , σ_i , and β_i are respectively the center, the width and the coefficient of the i -th Gaussian, and b is an optional bias. The scheme in Fig. 1 describes the structure of a predictor that uses D previous samples. Despite the similarity of their mathematical description, SVR and ELM differ for the learning algorithm, i.e. for the procedure that allow to obtain the model parameters $(L, \{\mu_i\}, \{\sigma_i\}, \{\beta_i\}, b)$ from the training set.

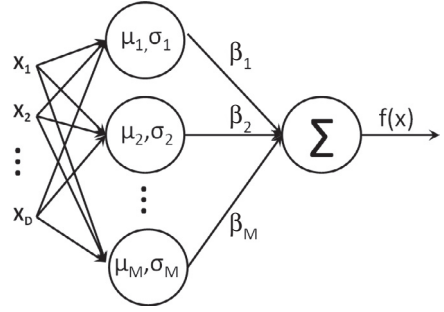


Fig. 1. The scheme of a kernel based predictor. The input are compared to the centers $\{\mu_i\}$ of all the units and their distance weighted with the widths $\{\sigma_i\}$ and used to produce the units' output through the Gaussian kernel. The units' contributions are then modulated through the units' weights $\{\beta_i\}$ and their sum constitutes the predicted value.

2.1. Autoregressive models

An Autoregressive (AR) model describes the values of a particular time series in terms of its past values [37]. In particular, the value of X_t is modeled as a combination of a part that is determined by the past values of the series and a part determined by an unpredictable event that happens at the time t (innovation). More formally, given a time series $\{X_t\}$, its autoregressive representation of order p , often denoted by $AR(p)$, is:

$$X_t = \alpha_0 + \sum_{k=1}^p \alpha_k X_{t-k} + \varepsilon_t \quad (2)$$

where α_0 is a constant, the innovation ε , is assumed to be white noise ($E(\varepsilon) = 0, E(\varepsilon^2) = \sigma^2$), and $\{\varepsilon_t\}$ are supposed to be normal independent and identically distributed (i.i.d.) random variables.

A Moving Average (MA) model describes the time series values in terms of linear combination of (unobserved) innovation values. A moving average representation of order q , often denoted by $MA(q)$, of the time series $\{X_t\}$ is:

$$X_t = \mu + \sum_{h=1}^q \beta_h \varepsilon_{t-h} + \varepsilon_t \quad (3)$$

The autoregressive and moving average models can be combined in the Autoregressive Moving Average (ARMA) model. An ARMA representation of autoregressive order p and moving average order q , $ARMA(p, q)$ is formally described as:

$$X_t = \alpha_0 + \sum_{k=1}^p \alpha_k X_{t-k} + \sum_{h=1}^q \beta_h \varepsilon_{t-h} + \varepsilon_t \quad (4)$$

When the time series is sampled from a stationary process, it can be represented by the above mentioned models. However, when the time series shows a trend or a seasonality, a more advanced class of models, namely the Autoregressive Integrated Moving Average (ARIMA) models, have to be used. The ARIMA model take into consideration also the difference series (i.e., the series resulting by computing the difference of time lagged series). In particular, the notation $ARIMA(p, d, q)$ is commonly used for indicating the ARIMA model with p , d , and q order of respectively

autoregression, differencing, and moving average. The formalization of this model is operated through the backward shift operator, B : $X_{t-1} = BX_t$. This allows to express X_{t-k} as $B^k X_t$. The ARIMA(p, d, q) representation of the time series $\{X_t\}$ is:

$$1 - \sum_{k=1}^p \alpha_k B^k \left((1 - B)^d X_t = 1 + \sum_{h=1}^q \beta_h B^h \right) \varepsilon_t \quad (5)$$

2.2. Support Vector Regression

Support Vector Machines (SVM) is a powerful method for classification [38,39] and regression [40]. In the latter domain, the method is usually named Support Vector Regression (SVR). In its original formulation, the regression function is obtained as the linear combination of some training samples, called Support Vectors (SV), but it can be extended to non-linear mapping through the use of suitable functions called kernels. The solution to the regression problem is obtained as the minimization of a suitable cost function, which can be chosen such that the optimization problem results to be convex. The cost function is usually composed of two parts, a functional, H_c , that takes into consideration the closeness of the solution, f , to the training data and a functional, H_s , that measures the smoothness of the solution; the two parts are balanced by means of a trade-off parameter, C :

$$\min_f H[f] = C \cdot H_c[f] + H_s[f] \quad (6)$$

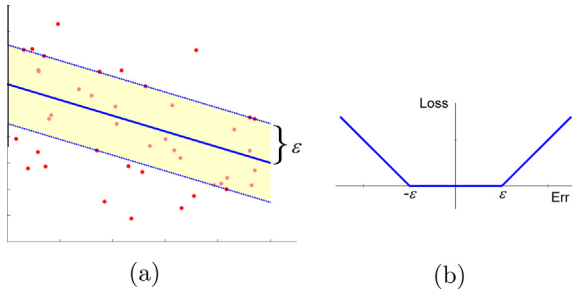


Fig. 2. Loss function computation in SVR. (a) The closeness of the solution (solid line) to the data points (circles) is computed up to a given accuracy, ε . (b) Those points that are closer than ε to the solution do not contribute to the loss, while the others contributes only for the part that exceed ε .

The closeness of the function to the data is often measured up to a given accuracy, ε , and all the points that are closer than ε to the solution do not contribute to the loss function, as depicted in Fig. 2. The cost function is hence ruled by three hyperparameters: the accuracy, ε , that represents the accepted distance between the training data and the solution; the trade-off, C , that balance the closeness of the solution to the training data and the robustness of the solution; and the width of the Gaussians used as kernels, σ , which in the basic SVR algorithm are constrained to have the same width. The convexity of the problem guarantees that the optimal solution (which identifies the SVs, $\{\mu_i\}$, and the corresponding coefficients, $\{\beta_i\}$) is unique.

2.3. Extreme Learning Machines

Neural networks constitutes a very variegated class of models for classification and function approximation [41,42]. Among these, the Radial Basis Function (RBF) networks are well-known, because of their simplicity and approximation power. In fact, they enjoy the universal approximation property (i.e., for every continuous function exists a RBF network that approximates arbitrarily well this function). The Extreme Learning Machine (ELM) is a RBF with a fixed architecture and randomly assigned hidden nodes parameters [43,44]. In particular, with the model described in (1), the parameters $\{\mu_i\}$ and $\{\sigma_i\}$ are randomly chosen with a given probability distribution. Given the training set $\{(x_j, y_j) | x_j \in \mathbb{R}^D, y_j \in \mathbb{R}, j = 1, \dots, N\}$, the output of the ELM network (1) gives rise to N equations that can be expressed in matricial notation as:

$$H\beta = \hat{Y} \quad (7)$$

where H is a $N \times L$ matrix such that $H_{j,i} = G(x_j; \mu_i, \sigma_i)$, $\beta = [\beta_1 \dots \beta_L]^T$, and $\hat{Y} = [\hat{y}_1 \dots \hat{y}_N]^T$. Given the training dataset and the hidden neurons parameters, under mild conditions, the weights β can be computed as:

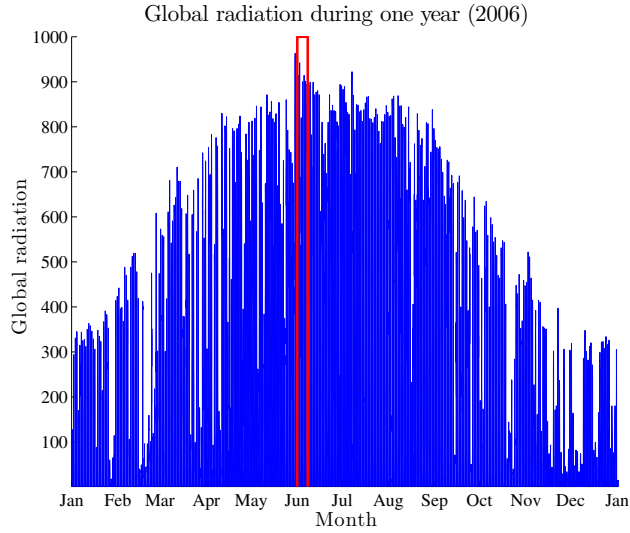
$$\hat{\beta} = (H^T G)^{-1} H^T \hat{Y} = H^{\dagger} \hat{Y} \quad (8)$$

where $H^{\dagger} = (H^T H)^{-1} H^T$ denotes the Moore–Penrose pseudo-inverse of the matrix H .

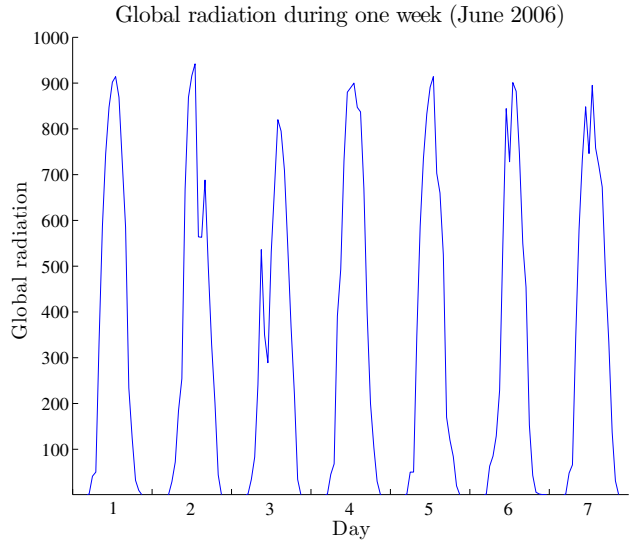
The ELM learning paradigm exploits the robustness of the solution with respect to the optimal value of the parameters of the neurons, and instead of spending

Year	M.	D.	H.	T [°C]	Hum. [%]	Global horiz. irradiance [W/m ²]	Diffuse horiz. irradiance [W/m ²]	Global horiz. illuminance [klux]
2005	12	22	9	1.7	71.9	25.9	22.5	1.5
2005	12	22	10	3.1	60.7	123.6	114.6	6.2
2005	12	22	11	4.0	52.6	216.6	205.5	12.0
2005	12	22	12	5.1	47.7	272.8	264.5	15.8
2005	12	22	13	6.4	48.5	284.2	119.8	16.1
2005	12	22	14	7.3	48.5	296.4	106.2	16.4
2005	12	22	15	8.0	49.2	209.1	96.4	11.8
2005	12	22	16	7.4	54.3	123.5	60.1	6.6
...

Fig. 3. Excerpt from the dataset used for the local prediction experiment.



(a)



(b)

Fig. 4. One year (a) and one week (b) of the measured global horizontal radiation. Note the trend in the year and in the day, but also the strong variability in the intraday values.

computational time for exploring the parameters' space, choose them by sampling a suitable distribution function (which encode the a priori knowledge on the problem), and compute the weights as the solution of the above described linear system. It can be shown that the solution $\hat{\beta}$ in (8) is an optimal solution in the least square sense, and has the smallest norm among the least square optimal solutions.

2.4. Persistence

In order to assess the performance of model in the short-term prediction of a time series, the persistence model, f_p , is often used. It is a naïve predictor that assumes

that the next value of the time series, x_t will be equal to the last known, x_{t-1} : $\hat{x}_t = f_p(x_{t-1}) = x_{t-1}$. It is obviously inappropriate for long-term prediction of time series of interest in real cases, but it can be used as a baseline forecast: it is supposed that any other model will perform better than the persistence model.

2.5. k -Nearest Neighbor interpolator

The k -Nearest Neighbor (k -NN) model is an instance-based or lazy learning paradigm used both for function approximation and classification [45]. It is used to predict the value of a function, f , in unknown points, given a sampling of the function itself (training data),

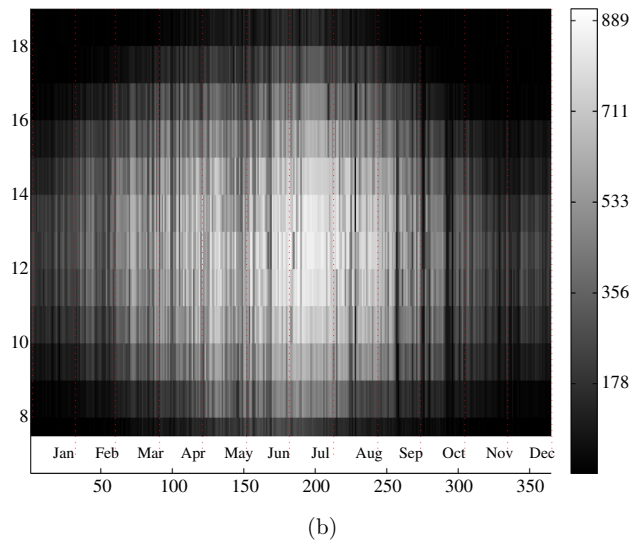
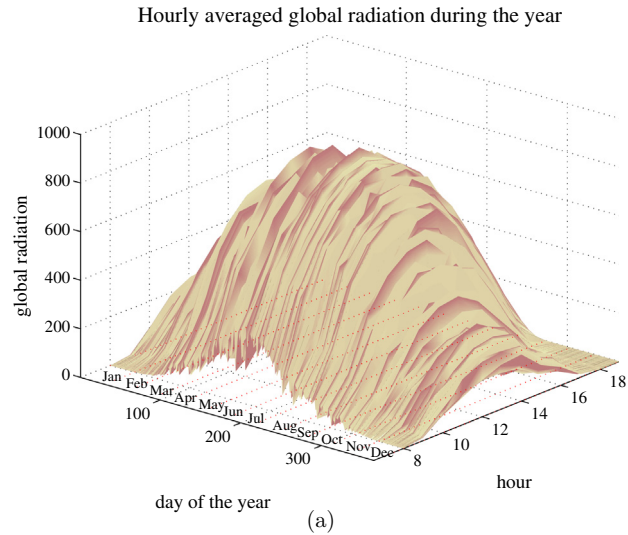


Fig. 5. The average global radiation for each day of the year and hour have been plotted as a surface. The roughness of the surface is due to variability, although a clear trend of the phenomenon can be acknowledged.

$\{(x_i, y_i) | y_i = f(x_i)\}$. For an unknown point, x , the value of $f(x)$ is estimated from the value of its k nearest neighbors, for a given k , using a suitable voting scheme or an average. The simplest scheme, often used in classification, estimates $f(x)$ as the most common output value among its neighbors, while in function approximation the average output value is often used. More complex schemes, such as the use of weighted averaging, or a sophisticated norm for computing the distance can be used as well. The k -NN can be used in time series prediction using some previously observed values for composing the input vectors.

3. Prediction based on local measurements

In the first experiment here described, the solar radiation prediction is operated on data captured at the plant

site. The dataset used has been collected by the MeteoLab [34,35] between October 2005 and October 2007 has been used. The MeteoLab station (located in Milan, Italy) measures and collects every ten minutes the following data: air temperature, relative humidity, global horizontal irradiance, diffuse horizontal irradiance, and global horizontal illuminance. The released dataset provides their hourly average (as illustrated in Fig. 3).

For this work, only the global horizontal radiation has been considered. Subsets of the available samples are reported in Fig. 4. In particular, Fig. 4a describes the global horizontal radiation measured in the year 2006, in Fig. 4b only one week is reported (the first week of June). It can be noticed that regularities are apparent both in the yearly and in the daily scale, but also that large deviations from the average behavior are possible, due to meteorological variability.

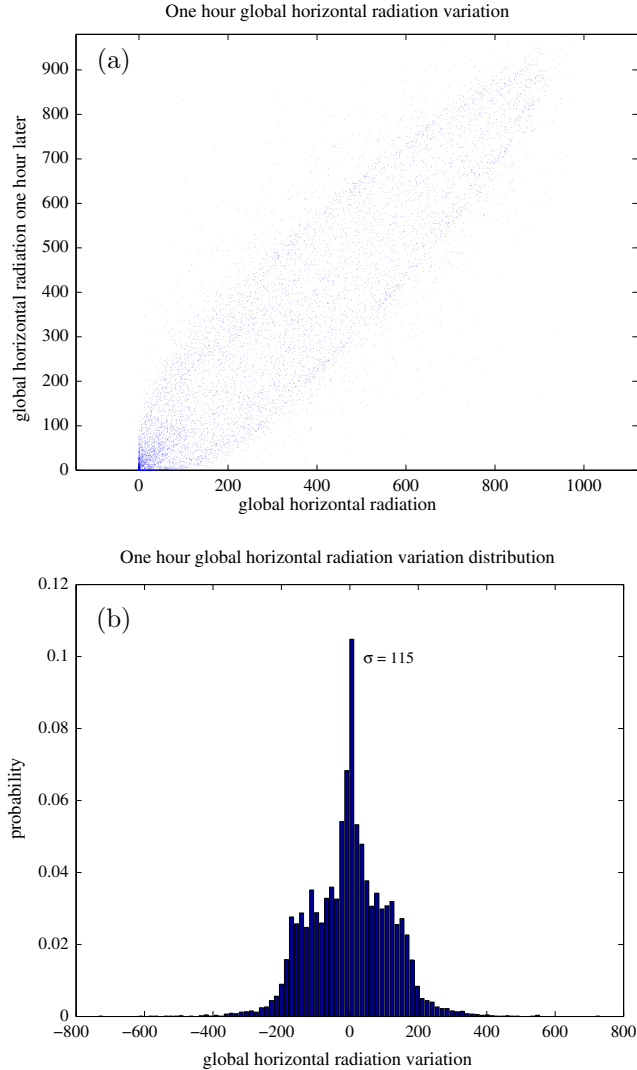


Fig. 6. The persistence predictor uses the global radiation value measured one hour earlier as predicted value. Panel (a) shows the relationship between the two measurements of the global horizontal radiation performed at the distance of one hour. Apparently, the samples are distributed along the identity line. In panel (b), the estimated probability density function of the difference between subsequent samples (with a standard deviation of 115).

As shown by surface depicted in Fig. 5, the global horizontal radiation varies both on daily and seasonal basis. The surface has been obtained by averaging the samples acquired in the same hour of the same day of the year. Although a trend is clearly recognizable, the variability of the global horizontal radiation (which depends also by fast changing meteorological phenomena) makes the surface very wrinkled.

Fig. 6, instead shows the relation between the global horizontal radiation acquired at two consecutive hours. In Fig. 6a the distribution of the points along the identity line supports the use of the persistence predictor. However, the maximum of the prediction error of the persistence can be considerably high since the length of the vertical section of the cloud of points is at least 300, where the maximum value of the radiation is about 900. The histogram in Fig. 6b resembles a mixture of two normal

distributions with the same mean. This is due to the fact that in the early and the late daylight hours the global radiation does not change very much (especially in winter). Hence, the consecutive samples acquired in those periods of time are quite similar, while the other moments of the day show a larger variability.

3.1. Dataset pre-processing

Since autoregressive models require equally time spaced values, the few values that are missing are interpolated using a simple rule that exploits the daily seasonality of the solar radiation. For each missing value, x_t , the set $\{x_{t-1}, x_{t+1}, x_{t-24}, x_{t+24}\}$, i.e., the set composed of the global radiation one hour earlier and later, and one day earlier and later is considered. The missing value is then replaced with the average of the collected values. Since the missing

data are few, the selected set has a meaningful number of elements even though some of the selected elements are missing too.

The resulting dataset is composed of 18,096 samples. Since the dataset covers a period of time of two years, and the autoregressive models require a training set composed of consecutive data, the first year has been used as training set. In this way, the yearly variability have a chance of being captured by the models. The data belonging to the second year has been randomly partitioned in the validation and training set. Hence, training, validation, and testing set are composed of, respectively, 9048, 4524, and 4524 samples.

3.2. Performance evaluation

For the evaluation of the performances, only the daylight hours data [8,19] has been considered. Besides, since the solar radiation cannot be negative, all the negative values predicted by the models are set to zero.

The prediction error has been evaluated as the average of the absolute error achieved on the testing data:

$$\text{Err}(f) = E(|x_t - f(x_t)|) \quad (9)$$

where $f(x_t)$ is the value for x_t predicted by the model f .

Although others error metrics can be used to assess the optimality of the model, the use of the absolute error is motivated by the target application for our models. In fact, we are interested in using the prediction to economical evaluation and the absolute error allows to directly relate the prediction error with the economical cost.

3.3. k -NN models prediction

The performance of a k -NN predictor depends on several hyperparameters, which operate only in the prediction stage, since the k -NN predictor does not requires other training process than just storing the training values. In particular, the behavior of the k -NN predictor is ruled by the number of the considered neighbors, k ; the number of dimension of the input space, D , which corresponds to the number of previous values used for the prediction; the weighting scheme, i.e., the law to assign the weights for the weighted averaging prediction.

The following values for the k and D hyperparameters have been challenged:

$$k \in [1, 30] \text{ and } D \in [1, 10] \quad (10)$$

Three weighting schemes have been tried: equal weight, weight proportional to the inverse of the neighborhood rank, and weight proportional to the inverse of the distance. Only the Euclidean norm has been used to compute the distance in the input space.

For the sake of comparison, the rules for generating the training, validation and test set will be the same used for the autoregressive models, described in Section 3.1.

3.4. Autoregressive models prediction

In order to train an autoregressive predictor, a suitable value for the hyperparameters that rule the optimization

procedure (i.e., the autoregression order, p , the moving average order, q , and the differencing order, d), have to be chosen. Several combination of the hyperparameters values have been tried and their effectiveness have been estimated through cross validation. In particular, the AR models have been challenged with $p \in \{1, \dots, 100\}$; the ARMA models have been challenged with the combination of p and q for $p \in \{1, \dots, 50\}$ and $q \in \{1, \dots, 50\}$; and the ARIMA model have been challenged with the combination of the following values of p , d , and q :

$$p \in \{1, \dots, 30\}, \quad d \in \{1, \dots, 3\}, \quad q \in \{1, \dots, 30\} \quad (11)$$

Since the training of the ARMA and ARIMA models requires consecutive training data, for avoiding of considering two separated periods of time for evaluating the validation and training error (which involves the risk of biased estimation due to the seasonality of the phenomenon under study), the prediction on the data not used to train the predictor has been carried out first and then the predicted period has been randomly sampled to obtain the validation and testing data.

3.5. Local prediction results

The persistence and k -NN predictors, described in Section 2, have been coded in Matlab, while for the autoregressive models (AR, ARMA, and ARIMA) their implementation in R have been used. Their performances have been evaluated using the prediction error, $\text{Err}(f)$, described in (9). Since the persistence predictor configuration does not need any hyperparameters, the whole dataset described in Section 3.1 has been used to assess its performances. Instead, the training of the k -NN and the autoregressive models are regulated by a pool of hyperparameters. Hence, the training set has been used to compute the model's parameters for each combination of the hyperparameters, then the validation dataset has been used to identify the best model and the prediction error of that model on the testing set has been used to measure the performance of the class of the predictors.

As reported in Table 1, the persistence predictor has achieved an error $\text{Err}(f_p) = 88.3$, while the k -NN achieved an error $\text{Err}(f_{k\text{-NN}}) = 47.7$, for $D = 4$, $k = 17$, and using the inverted distance weighting scheme.

The AR model that scores the lower validation error has been trained using $p = 97$ and achieved $\text{Err}(f_{\text{AR}}) = 43.5$; the best ARMA model has been trained using $p = 28$ and $q = 22$, achieving $\text{Err}(f_{\text{ARMA}}) = 42.7$; the best ARIMA model, trained using $p = 23$, $d = 1$, and $q = 16$, achieved a testing error $\text{Err}(f_{\text{ARIMA}}) = 43.3$.

Table 1
Test error achieved by the predictors (local prediction).

Predictor	$\text{Err}(f)$	std
Persistence	88.3	74.2
k -NN	47.7	59.7
AR	43.5	56.9
ARMA	42.7	56.5
ARIMA	43.3	56.5

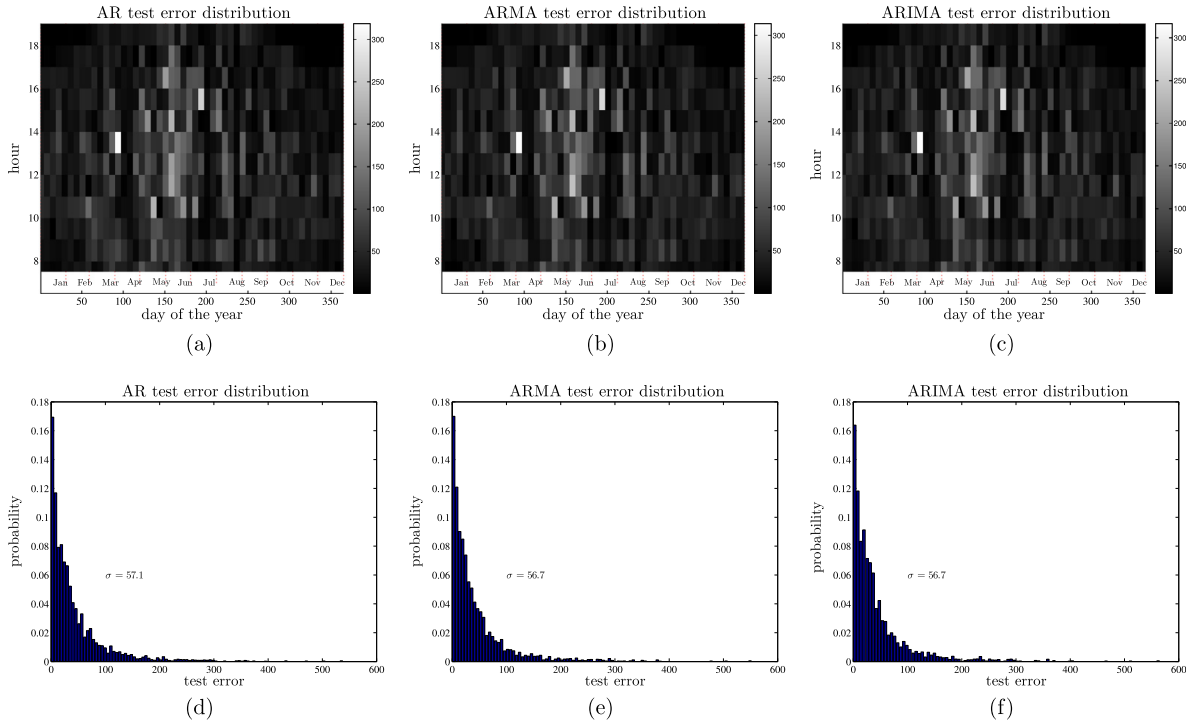


Fig. 7. Test error distribution. In panels (a)–(c), the test error produced by respectively the AR, the ARMA, and the ARIMA predictor are reported with respect to the day of the year and the hour. In panel (d)–(f), the estimated probability density function of the test error of the autoregressive models.

Fig. 7 shows the distribution of the prediction error of AR, ARMA and ARIMA models. Hardly some differences can be spotted in Fig. 7a–c, although Fig. 7d–f reveal a slightly compact histogram for ARMA and ARIMA. In particular, the error peaks in Fig. 7a–c are in the same position, probably due to some fast changing meteorological events happened in that period that modified the usual global radiation pattern.

4. Prediction based on remote measurements

The second experiment concerns the solar radiation prediction operated with data captured at a remote site. For this experimental session, two datasets collected by ARPA Lombardia [36] between October 2005 and September 2008 have been used. The datasets contain the hourly measurement of the global radiance in two sites (Lambrate and Rodano, Italy) separated by about 10 km.

Both the datasets shows a distribution similar to that illustrated in Figs. 4 and 5 and considerations reported in Section 3 similarly holds here. In Fig. 8, the relation between the global horizontal radiation acquired at two consecutive hours at the two sites is shown. Also in this case, considerations for the validity of the persistence predictor and the distribution of the difference in the values of the two sites are very similar to those made in the previous experiment. However, it is worth noting that here the standard deviation of the difference distribution (Fig. 8) is higher than in the previous case (127 instead of 115). This is reasonable, because the variation due to the meteorological phenomena is improved by the geographical displacement of the two sites.

4.1. Dataset pre-processing

Since our work requires the corresponding values of the two sites, each database has been purged of the samples that do not have a matching sample in the other database measured at the same time of the same day. After this operation, the datasets are composed of 22,961 samples which have been used for composing the input vectors for the prediction as described in Section 2. In particular, we tried to predict the global radiation in Rodano from measurements in Lambrate. Hence, each input vector has been composed by D consecutive samples from Lambrate, for $D \in \{1, \dots, 10\}$, taken at time $\{t - D, \dots, t - 1\}$, which has been related to the sample from Rodano at the time t . Besides, also the temporal information of t (hour of the day and day of the year) has been provided as input. The data has been randomly partitioned in training, validation, and testing set (using a proportion of 50–25–25%, respectively). In order to assign the same importance to all the components, the data have been normalized using the maximum of the measurement in the training set for the global radiation components, 23 for the hour of the day, and 364 for the day of the year.

4.2. Performance evaluation

The performances of the predictors have been evaluated using the same set-up described in Section 3.2.

4.3. k -NN models prediction

The k -NN models challenged have been the same described in Section 3.3.

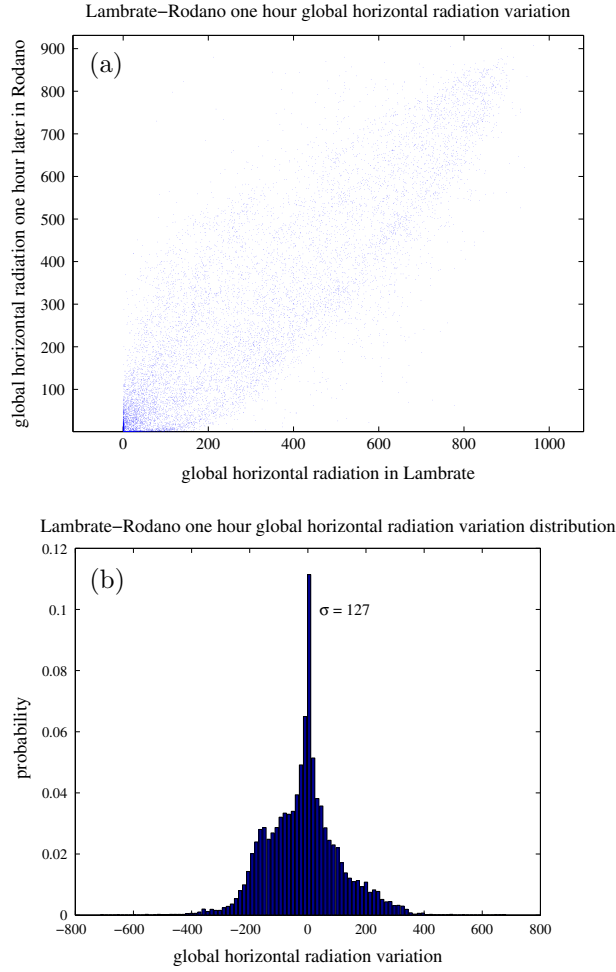


Fig. 8. The persistence predictor uses the global radiation value measured one hour earlier as predicted value. Panel (a) shows the relationship between the two measurements of the global horizontal radiation performed at Lambrate and that performed at Rodano one hour later. The samples are evidently distributed along the identity line. In panel (b), the estimated probability density function of the variation (which standard deviation is 127).

4.4. Prediction through ELM models

In order to train an ELM neural network as a time series predictor, the hyperparameters that regulate the optimization procedure (i.e., the probability distribution of the neuron parameters, μ_i and σ_i , the input space dimension, D , and the number of the neurons, L), have to be set to the proper value.

The dimensionality of the input training data, D (i.e., the number of previous samples considered for the prediction) has been chosen in the interval $[1, 10]$ like in (10), while networks of several sizes, L , have been challenged:

$$L \in \{10, 25, 50, 100, 250, 500, 1000, 2000, 3000\} \quad (12)$$

Table 2

Test error achieved by the predictors (remote prediction).

Predictor	Symbol	Err(f) (std)	Err(f^*)
Persistence	Err(f_P)	95.4 (84.2)	–
k -NN	Err(f_{k_NN})	41.4 (57.0)	53.1
ELM	Err(f_{ELM})	42.7 (57.0)	58.5
SVR	Err(f_{SVR})	40.5 (59.3)	57.2

Since the Gaussian has a meaningful output only in a neighborhood of its center, the distribution of the centers, μ_i , here indicated as the random variable A , is usually derived from the position of the input training data. In particular, three distributions have been tried for A : A_1 , uniform distribution in the bounding box of the input training data; A_2 and A_3 , respectively sampling with and without replacement from the input training data. The width of the Gaussian, σ , regulates the extent of its influence region (in regions further then 3σ from μ , the output is negligible). Since when the dimensionality of the input space increases the data become sparse (a problem often referred to as “curse of dimensionality”), for fairly

Table 3

Test error achieved by the ELM predictor.

# Trial	Err(f_{ELM})	Mean	Std
1	42.9	42.7	0.322
2	42.9		
3	42.2		
4	43.0		
5	42.6		

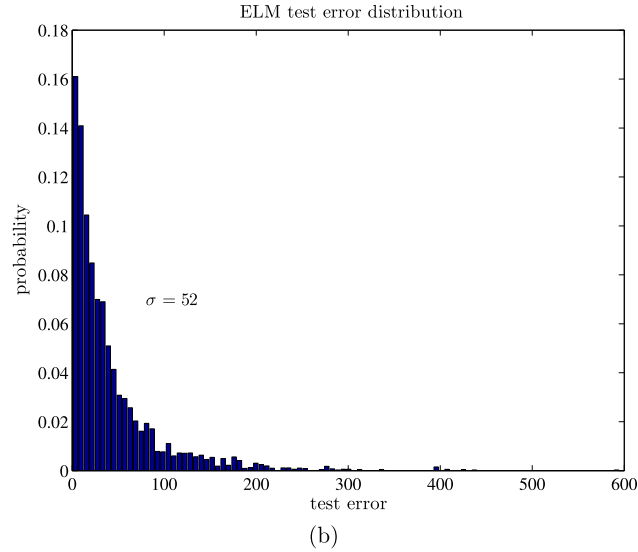
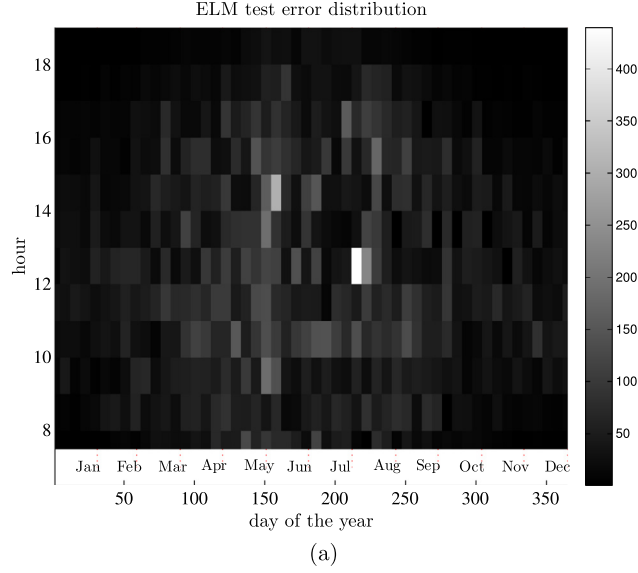


Fig. 9. ELM test error distribution. In panel (a), the average test error achieved in all the trials is reported with respect to the day of the year and the hour. The error is almost uniform on the domain, although it slightly follows the seasonal and daily variability. In panel (b), the estimated probability density function of the test error (which standard deviation is 52.0).

comparing the effects of the dimensionality, we chosen a set of relative values for the width, r , that are then customized to the actual value of D when the width of the neurons are computed. This is realized assigning to σ the relative width, r , multiplied by the diagonal of the bounding box of the input training data. The value challenged for r are:

$$r \in \{0.01, 0.05, 0.1, 0.5, 1\} \quad (13)$$

Once the proper value of σ has been computed for the considered dimensionality, the width of the neurons, $\{\sigma_i\}$ are sampled from $B \sim N(\sigma, \sigma/3)$ (i.e., $\{\sigma_i\}$ are distributed as a normal with mean σ and standard deviation $\sigma/3$). Negative values for σ_i , that are possible but unacceptable, are discarded and resampled.

Since the parameters of the network are chosen by chance, five trials with the same combination of the hyperparameters has been run and the performance of the parameter combination has been averaged.

4.5. Prediction through SVR

In order to train a SVR predictor, the hyperparameters that regulate the optimization procedure, have to be set to the proper value. Since the optimal values cannot be estimated *a priori*, several combinations have to be tried and their effectiveness assessed by cross validation.

The hyperparameters values that we challenged are:

- the input dimensionality, D : $[1, 10]$, as in (10);
- the accuracy, ε : $\{0.01, 0.1, 0.5, 1\}$;

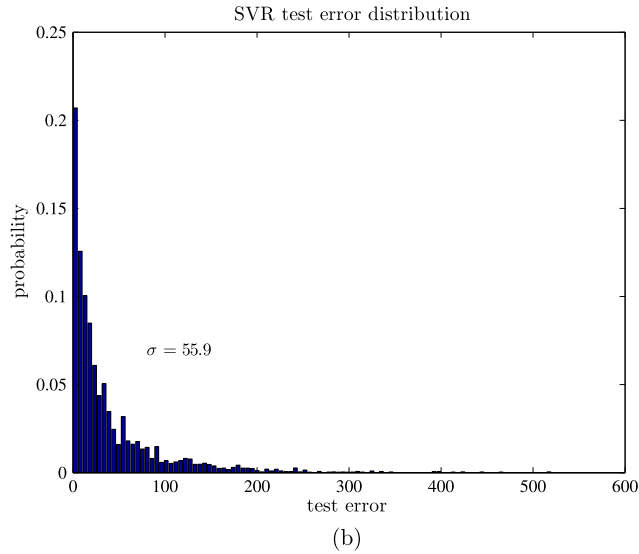
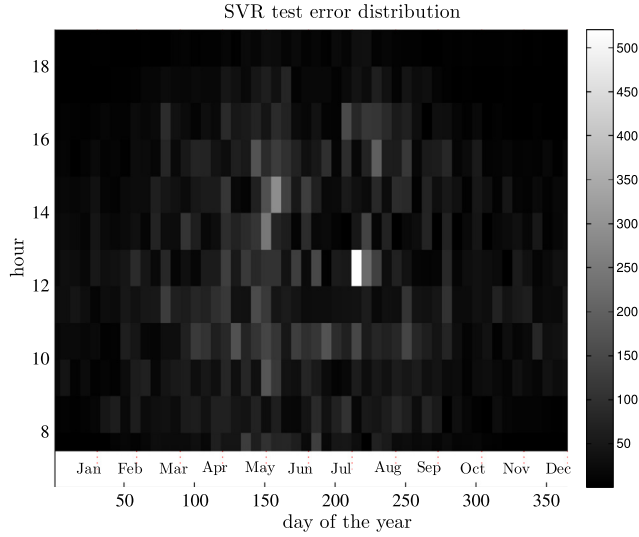


Fig. 10. SVR test error distribution. In panel (a), the test error is reported with respect to the day of the year and the hour. The error is almost uniform on the domain, although it slightly follows the seasonal and daily variability. In panel (b), the estimated probability density function of the test error (which standard deviation is 55.9).

- the regularization trade-off, C : $\{0.1, 1, 10, 100\}$;
- the width, σ : similarly to the ELM case, the proportionality factor r in (13) has been experimented to set σ depending on D .

4.6. Remote prediction results

The persistence, k -NN, and ELM predictors have been coded in Matlab, while for the SVR models we used the SVM^{light} [46], and their performances evaluated using the prediction error, $Err(f)$, described in (9). Since the persistence predictor configuration does not need any hyperparameters, the whole dataset described in Section 4.1 has been used to assess its performances. Instead, the training

of the k -NN, the ELM and SVR models are regulated by a pool of hyperparameters. Hence, similarly to what operated in the local prediction experiment (Section 3.5), the training set has been used to estimate the model's parameters for each combination of the hyperparameters, then the validation dataset has been used to identify the best model (i.e., the one that achieved the lowest prediction error on the validation dataset) and the prediction error of that model on the testing set has been used to measure the performance of the class of the predictors.

As reported in Table 2, the persistence predictor has achieved an error $Err(f_p) = 95.4$. This value should also be compared to the persistence measured at each site, which is 89.9 for Lambrate and 83.6 for Rodano. The fact

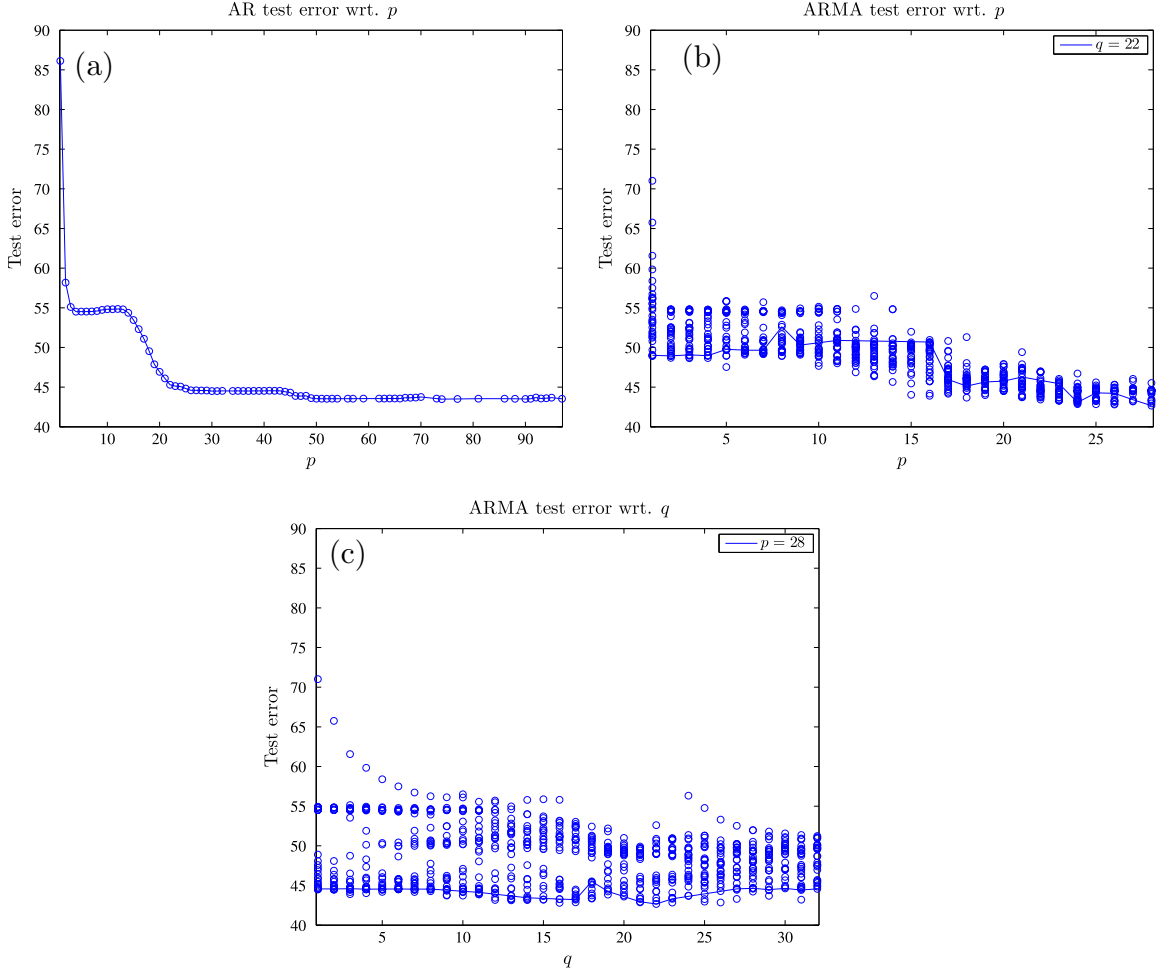


Fig. 11. AR, (a), and ARMA, (b) and (c), test error wrt. the hyperparameters.

that these three values are quite similar supports our working hypothesis, i.e., the data from one site can be used to predict the measurement on the other site.

In fact, as shown in Table 2, all the models have been able to halve the prediction error with respect to the persistence. In particular, the k -NN achieved an error $\text{Err}(f_{k\text{-NN}}) = 41.4$, for $D = 2$, $k = 9$, and using the inverted distance weighting scheme.

The best ELM model, which achieved an error of $\text{Err}(f_{\text{ELM}}) = 42.7$, resulted in the one trained using the following combination of hyperparameters: $D = 2$, $r = 0.1$, $L = 500$, and using the A_2 distribution for choosing the centers position. The performance achieved in each of the five trials for this model is reported in Table 3, with their average (42.7) and standard deviation (0.322). Since the standard deviation is very small with respect to the average, the ratio between the two values witnesses the stability of the learning.

The lowest error has been obtained by the best SVR model, which achieved an error of $\text{Err}(f_{\text{SVR}}) = 40.5$, using $L = 3853$ support vectors. The training has been realized with $D = 2$, $r = 0.1$, $\varepsilon = 0.01$, and $C = 1$.

The distribution of the test error with respect to the hour of the day and the period of the year for the ELM and SVR models are reported in Figs. 9 and 10, respectively. Since the test set does not include all the possible time combinations, the error has been reported averaging those of seven consecutive days. It can be noticed that both the distributions are very similar (although the SVR distribution is slightly smoother); the error is reasonably low in most of the domain, with few noticeable exceptions.

For the sake of comparison, we challenged the predictor on datasets purged of temporal references. The performance achieved in this situation has been reported in Table 2, as $\text{Err}(f^*)$. It can be noticed that the error significantly improves. Moreover, the dimension of the input space also increases ($D = 7$, for all the models).

5. Discussion

5.1. Distribution of the prediction error

Fig. 7 shows the distribution of the prediction error of AR, ARMA and ARIMA models for the local prediction

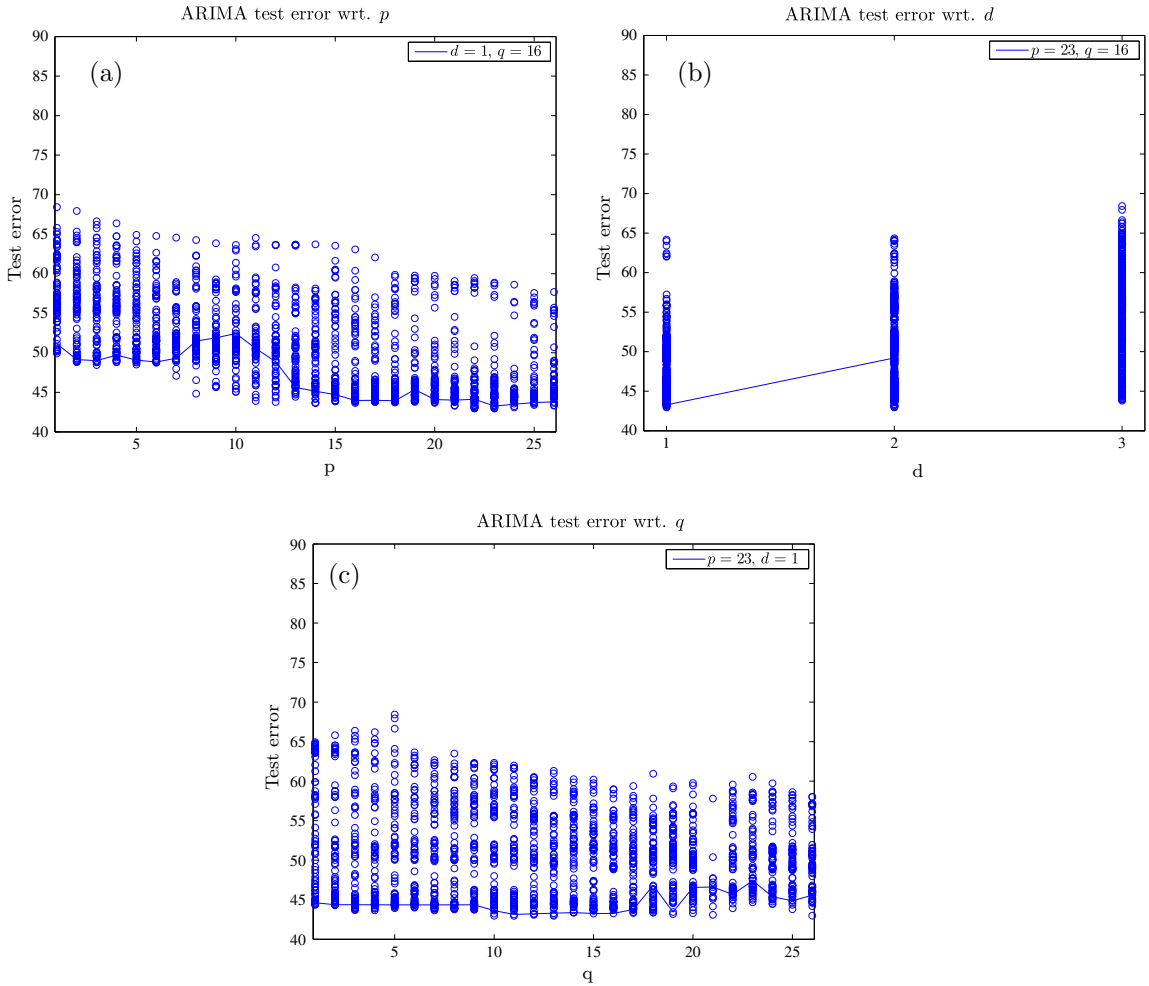


Fig. 12. ARIMA test error wrt. the hyperparameters.

experiment. They are very similar and, in particular, the peaks in the error in Fig. 7a–c are in the same temporal positions, which means that the corresponding values in the dataset are quite departed from the usual radiation pattern. Similar considerations are also valid for the comparison of Figs. 9 and 10, where the distribution of the error of ELM and SVR models for remote prediction are represented.

5.2. Sensitivity of the hyperparameters

While Fig. 7 shows the substantial equivalence of the autoregressive predictors, Figs. 11 and 12, show the robustness of the optimal pool of hyperparameters selected through cross validation. In these figures, the test-ing error for all the combinations of the hyperparameters are reported as circles.

In Fig. 11a, the performance of the AR models for each value of p is represented. It can be noted that the testing error decreases step-wise as the number of previous values used for the estimate increases, and, in particular, that drops in the error can be found around multiples of 24 h.

This is intuitively explained by the 24 h seasonality of the dataset. Although the error can possibly decrease for larger values of p , the search has been stopped because of the long time consumed by the computation of the model parameters for a large values of p . Besides, for large values of p , numerical errors can arise in the optimization routine, which results in failure in the estimate of the model parameters. In fact, it can be noted that in the graph some testing errors are missing (e.g., for $p = 60$).

In Fig. 11b–c, the performance of the ARMA models with respect to the values of p and q are represented. The solid line connects the error for each value of p and q when, respectively, $q = 19$ and $p = 24$, i.e., the behavior of the error when one hyperparameter is fixed to the optimal value. It can be noticed that, although the variance of the error tends to decrease with the increase of both p and q , the testing error does not exhibit a stable trend. This means that the solution is not very robust: for a different randomization of the validation and the testing sets, the best model could be characterized by different values of p and q . In particular, the error tends to decrease when p increases. Like for the AR model, larger values of p could

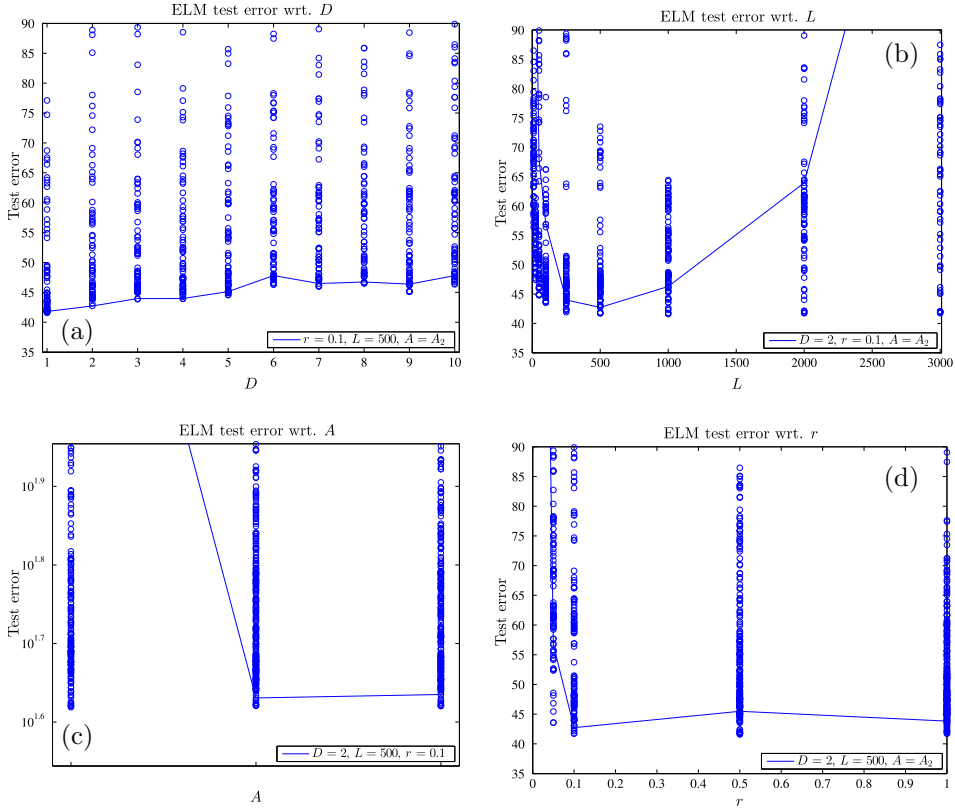


Fig. 13. Test error (averaged over five trials) wrt. the ELM hyperparameters.

be considered, but at large computational cost and numerical instability of the model parameters estimate.

In Fig. 12a–c, the performance of the ARIMA models with respect to the values of p , d , and q are represented. In each graph, the solid line represents the error as a function of the considered hyperparameter, when the other two are fixed at their optimal value. Although the error variance appears to be larger than in the ARMA model, the testing error becomes closer to the minimum for a large number of combinations of the parameters. For the p hyperparameter, it is clear from the graph that for value of p greater than 14, the performance of the predictor tends to improve (circles are more dense in the region close to the minimum). Larger values of p has not been considered in the experiment because of the increase in computational time and the numerical instability of the optimization procedure. The differencing parameter, d , that characterize ARIMA with respect to the other models here considered, shows a slight improvement in the prediction performance when it is assumed to be equal to one (the distribution of the error seems to be more sparse for higher values of d).

In Fig. 13, the test error achieved with all the challenged ELM models and for all the trials are reported with respect to the hyperparameter values used for the training. In order to understand the influence of the single hyperparameter on the performance of the ELM network, the graph of the error (averaged on the five trials) achieved using the hyperparameter values that achieved the best validation

error with respect to the value of the considered hyperparameter is also plotted. In particular, Fig. 13a shows the error wrt. the number of previous values in input. It should be noticed that the minimum is here achieved for the value $D = 1$, while the best ELM model has $D = 2$. This is due to the fact that the best model is selected using the validation dataset, while here the performance on the test dataset are reported. Again, this is a symptom of the variability of the dataset: with another randomization the best model could be different. However it should be noticed that the excursion in the error for the different values of D (less than ten) is well below the standard deviation of the error (57) reported in Table 2. The analysis of the other subpanels suggests that the optimal values found the other parameters, instead, are more robust. In particular, the Fig. 13b shows the error wrt. the size of the ELM network. It can be noted that although the minimum error for each value of L is close to the absolute minimum, the distribution for the higher value of L (1000, 2000, and 3000) seems to be more sparse. This means that the opportunities offered by the large number of units are not fully exploited by the distribution laws chosen for the neuron parameters (A and B). Hence, when the number of neurons is large, a great number of them is wasted.

In Fig. 14, the test error achieved with all the challenged SVR models are reported with respect to the hyperparameter values used for the training. The distribution of the errors wrt. the hyperparameter D (Fig. 14a) is similar to

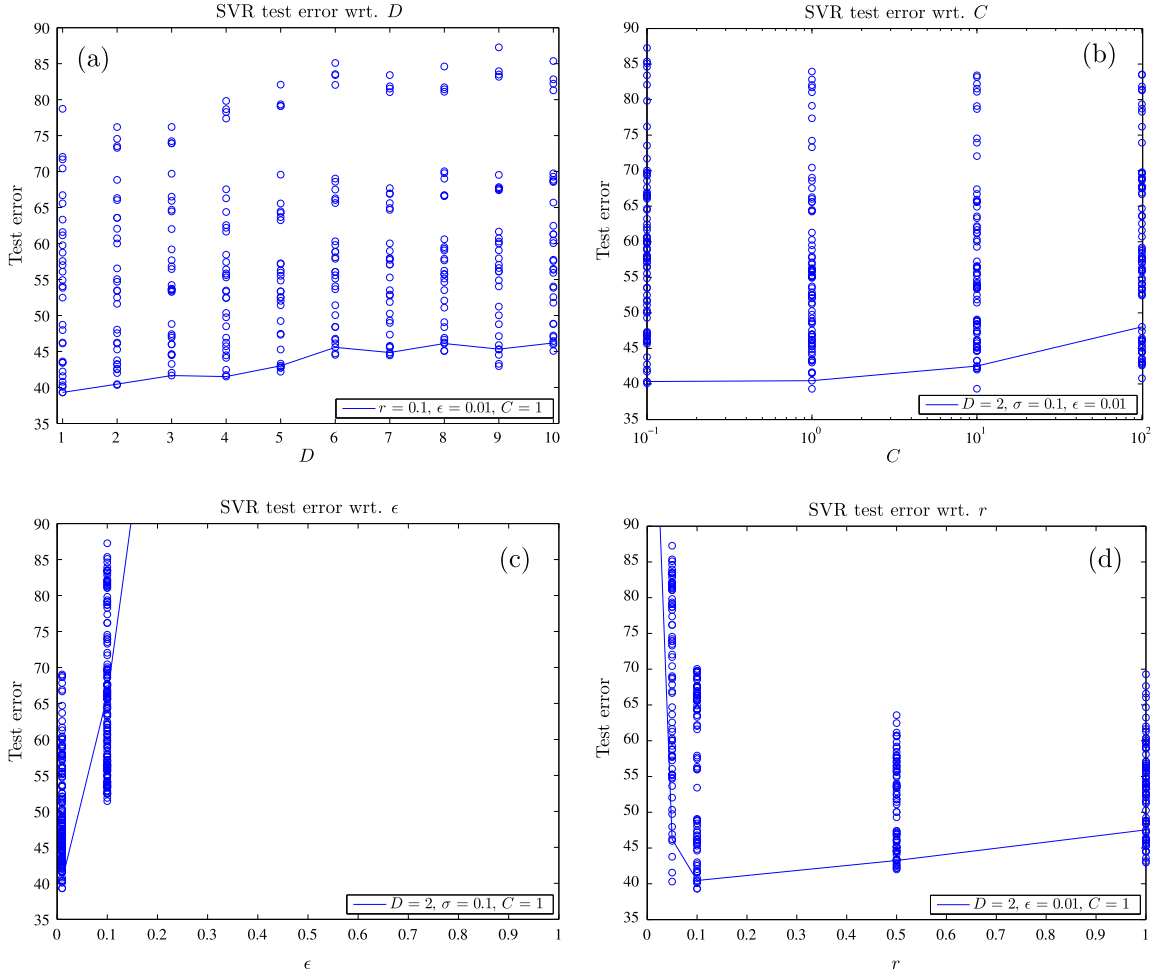


Fig. 14. Test error wrt. the SVR hyperparameters.

that of the ELM model (reported in Fig. 13b); hence, similar considerations holds also in this case. Also the distributions wrt. r (Figs. 13d and 14d) show that for values lower than 0.1 the error greatly increases, while mildly increases for larger values.

5.3. Comparison of the two experiments output

Although the two experiments have been conducted under different experimental condition and with two different datasets, to some extent they can be compared. This is due to the fact that they have been conducted on the same geographical area and using data measured in largely overlapping period of time.

The first common property is that the prediction models in both the experiments achieve an accuracy that is almost the half of the persistence accuracy. In the second experiment the ratio can seem also better, but it decreases when the results of the prediction without the time information are considered. The second consideration is that a simple predictor as the k -NN is able to achieve performance comparable to those of more sophisticated models. In fact, the difference in the mean of the testing error of all

the models is well below the standard deviation of the error (hence, such a difference cannot be considered really meaningful). It should be noted, however, that since the k -NN stores all the training data, it makes use of more actual parameters than the other models and hence it is impractical for real implementation.

5.4. Comparison with other approaches

The comparison of our experiments with the results in literature is a difficult task due to the differences in both the experimental conditions and the target application. The experimental condition can differ for the geographical position and the historical data period. The target application, instead, affects the comparison because different applications may induce the use of different error metrics to measure the performance of the predictors. Ultimately, this is the main obstacle to the comparison.

For instance, if for the target application the overall energy production is the goal, the absolute error like in (9) is the proper performance metrics, since its value is directly related to the economical cost of the prediction error. Differently, a normalized error like the mean

absolute percentage error (MAPE) used in [15] is suitable to make comparison among different datasets. Percentage error are also used, for instance, in [14,16,17].

Considering the maximum value of the solar radiation in the considered period as a normalization factor, the relative error achieved by both the ELM and the SVR predictors is about 5%. Since according to [15], a relative error below 10% means high prediction accuracy, while $10\% \leq \text{MAPE} \leq 20\%$ means good prediction, $20\% \leq \text{MAPE} \leq 50\%$ means reasonable prediction, $\text{MAPE} \geq 50\%$ means inaccurate forecasting, our predictors have a more than satisfactory performance.

6. Conclusion

In this paper two experiments for the solar radiation prediction have been described and discussed. In the first experiment, the data from a local measurement station have been used to build a model of the solar radiation. In the second one, the data from a remote public weather station have been used to predict the solar radiation on a ten kilometers distant site. In both the experiments the challenged models provided a comparable average prediction error, well below the persistence error and compatible with the results found in literature.

These results allow to consider as promising the use of data captured from remote stations for both short-term prediction and maintenance scheduling. In fact, once the ability to use data from remote public weather stations to predict the radiation condition in the plant site will be assessed, this value can be used to compare the theoretical power output of the solar panels through a suitable model and to identify through a cost analysis the optimal scheduling of the maintenance intervention.

Acknowledgment

The authors would like to thank Tiziana Poli of the Department of Architecture, Built Environment and Construction Engineering, Politecnico di Milano, Milan, Italy, for the dataset used for the local prediction experiment (Section 3).

References

- [1] M. Catelani, L. Ciani, L. Cristaldi, M. Faifer, M. Lazzaroni, P. Rinaldi, FMECA technique on photovoltaic module, in: IEEE International Instrumentation And Measurement Technology Conference (I2MTC 2011), 2011, pp. 1717–1722.
- [2] M. Catelani, L. Cristaldi, L. Ciani, M. Faifer, M. Lazzaroni, M. Rossi, Characterization of photovoltaic panels: the effects of dust, in: IEEE International Energy Conference and Exhibition (ENERGYCON 2012), 2012, pp. 49–54.
- [3] L. Cristaldi, M. Faifer, S. Ierace, M. Lazzaroni, M. Rossi, An approach based on electric signature analysis for photovoltaic maintenance, University of Sunderland, Sunderland, UK, 2012, pp. 1–8.
- [4] L. Cristaldi, M. Faifer, M. Rossi, F. Ponci, A simple photovoltaic panel model: characterization procedure and evaluation of the role of environmental measurements, IEEE Trans. Instrument. Measure. 61 (10) (2012) 2632–2641.
- [5] L. Cristaldi, M. Faifer, S. Ierace, M. Lazzaroni, M. Rossi, An approach based on electric signature analysis for photovoltaic maintenance, in: IEEE International Energy Conference and Exhibition (ENERGYCON 2012), 2012, pp. 1–8.
- [6] M. Catelani, L. Ciani, L. Cristaldi, M. Faifer, M. Lazzaroni, Electrical performances optimization of photovoltaic modules with FMECA approach, Measurement 46 (10) (2013) 3898–3909.
- [7] L. Cristaldi, M. Faifer, M. Rossi, L. Ciani, M. Lazzaroni, S. Toscani, Photovoltaic plant efficiency evaluation: a proposal, in: 12th IMEKO TC10 Workshop on Technical Diagnostics: New Perspective in Measurements, Tools and Techniques for Industrial Applications, Proceedings, 2013, pp. 260–265.
- [8] F. Attivissimo, F. Adamo, A. Carullo, A.M.L. Lanzolla, F. Spertino, A. Vallan, On the performance of the double-diode model in estimating the maximum power point for different photovoltaic technologies, Measurement 46 (9) (2013) 3549–3559.
- [9] R. Eke, A.S. Kavasoglu, N. Kavasoglu, Design and implementation of a low-cost multi-channel temperature measurement system for photovoltaic modules, Measurement 45 (6) (2012) 1499–1509.
- [10] S. Ferrari, M. Lazzaroni, V. Piuri, A. Salman, L. Cristaldi, M. Faifer, S. Toscani, A computational intelligence approach to solar panel modelling, in: IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2014, 2014, pp. 1261–1266.
- [11] L. Cristaldi, M. Faifer, M. Rossi, S. Toscani, M. Catelani, L. Ciani, M. Lazzaroni, Simplified method for evaluating the effects of dust and aging on photovoltaic panels, Measurement 54 (2014) 207–214.
- [12] L. Cristaldi, M. Faifer, N. Lazzaroni, A. Khalil, M. Catelani, L. Ciani, Failure modes analysis and diagnostic architecture for photovoltaic plants, in: Proceedings of the 13th IMEKO TC10 Workshop on Technical Diagnostics: Advanced Measurement Tools in Technical Diagnostics for Systems, 2014, pp. 206–211.
- [13] V. Kostylev, A. Pavlovski, Solar power forecasting performance – towards industry standards, in: 1st International Workshop on the Integration of Solar Power into Power Systems, Aarhus, Denmark, 2011.
- [14] A. Mellit, M. Bengham, S. Kalogirou, An adaptive wavelet-network model for forecasting daily total solar-radiation, Appl. Energy 83 (7) (2006) 705–722.
- [15] A.K. Yadav, S. Chandel, Solar radiation prediction using artificial neural network techniques: a review, Renew. Sustain. Energy Rev. 33 (1) (2014) 772–781.
- [16] R. Perdomo, E. Banguero, G. Gordillo, Statistical modeling for global solar radiation forecasting in Bogotá, in: 35th IEEE Photovoltaic Specialists Conference (PVSC), 2010, 2010, pp. 002374–002379.
- [17] T. Raji, A. Boyo, M. Hedayatshodeh, Analysis of global solar radiation data as time series data for some selected cities of western part of Nigeria, Int. J. Advan. Renew. Energy Res. 1 (1) (2012) 14–19.
- [18] C. Glasbey, Non-linear autoregressive time series with multivariate gaussian mixtures as marginal distributions, Appl. Statist. 50 (2) (2001) 143–154.
- [19] E. Meyer, E. van Dyk, Assessing the reliability and degradation of photovoltaic module performance parameters, IEEE Trans. Reliab. 53 (2004) 83–92.
- [20] S.J. Rhee, K. Ishii, Using cost based FMEA to enhance reliability and serviceability, Advan. Eng. Inform. 17 (2008) 179–188.
- [21] L. Cristaldi, M. Faifer, M. Lazzaroni, M. Rossi, S. Toscani, Photovoltaic plant maintenance: a method base on Economic evaluation of PV system losses, 2013, pp. 98–111.
- [22] S. Ferrari, M. Lazzaroni, V. Piuri, A. Salman, L. Cristaldi, M. Faifer, A data approximation based approach to photovoltaic systems maintenance, in: Proceedings of the 2013 IEEE Workshop on Environmental Energy and Structural Monitoring Systems EESMS 2013, 2013, pp. 7–12.
- [23] P. Mishra, J. Joshi, Reliability estimation for components of photovoltaic systems, Energy Convers. Manage. 37 (1996) 1371–1382.
- [24] N. Gautam, N. Kaushika, Reliability evaluation of solar photovoltaic arrays, Solar Energy 72 (2002) 129–141.
- [25] M. Catelani, L. Cristaldi, M. Lazzaroni, L. Peretto, P. Rinaldi, L'affidabilità nella moderna progettazione: un elemento competitivo che collega sicurezza e certificazione, I quaderni del GMEE, vol. 1, GMEE, 2008.
- [26] M. Lazzaroni, L. Cristaldi, L. Peretto, P. Rinaldi, M. Catelani, Reliability Engineering: Basic Concepts and Applications in ICT, Springer, 2011.
- [27] M. Catelani, L. Cristaldi, M. Lazzaroni, L. Peretto, P. Rinaldi, Le parole della fidatezza, Tutto misure 13 (1) (2011) 49–53.
- [28] M. Catelani, L. Cristaldi, M. Lazzaroni, L'affidabilità come requisito di progetto di componenti e sistemi: le strutture serie e parallelo, Tutto misure 13 (3) (2011) 213–216.
- [29] F. Bellocchio, S. Ferrari, M. Lazzaroni, L. Cristaldi, M. Rossi, T. Poli, R. Paolini, Illuminance prediction through SVM regression, in: IEEE

- Workshop on Environmental Energy and Structural Monitoring Systems (EESMS), 2011, 2011, pp. 1–5.
- [30] S. Ferrari, A. Fina, M. Lazzaroni, V. Piuri, L. Cristaldi, M. Faifer, T. Poli, Illuminance prediction through statistical models, in: 2012 IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS), 2012, pp. 90–96.
- [31] S. Ferrari, M. Lazzaroni, V. Piuri, A. Salman, L. Cristaldi, M. Rossi, T. Poli, Illuminance prediction through extreme learning machines, in: 2012 IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS), 2012, pp. 97–103.
- [32] S. Ferrari, M. Lazzaroni, V. Piuri, L. Cristaldi, M. Faifer, Statistical models approach for solar radiation prediction, in: IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2013, 2013, pp. 1734–1739.
- [33] S. Ferrari, M. Lazzaroni, V. Piuri, A. Salman, L. Cristaldi, M. Faifer, Computational intelligence models for solar radiation prediction, in: IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2013, 2013, pp. 757–762.
- [34] T. Poli, L.P. Gattoni, D. Zappalà, R. Gottardi, Daylight measurement in Milan, in: Proceedings of PLEA2006, Conference on Passive and Low Energy Architecture, 2006.
- [35] T. Poli, L.P. Gattoni, D. Zappalà, R. Gottardi, Daylight measurement in Milan, in: Clever Design, Affordable Comforta Challenge for Low Energy Architecture and Urban Planning, Raphael Compagnon & Peter Haefeli and Willi Weber, Geneva - CH, 2006, pp. 429–433.
- [36] ARPA Lombardia <<http://ita.arpalombardia.it/>>.
- [37] G.E.P. Box, G. Jenkins, Time Series Analysis, Forecasting and Control, Holden-Day, Incorporated, 1990.
- [38] V.N. Vapnik, Statistical Learning Theory, Wiley, 1998.
- [39] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000.
- [40] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (2004) 199–222.
- [41] L. Fausett, Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, Prentice Hall international ed., Prentice-Hall, 1994.
- [42] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Inc., New York, NY, USA, 1995.
- [43] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
- [44] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
- [45] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inform. Theory 13 (1) (1967) 21–27.
- [46] T. Joachims, Making large-scale SVM learning practical, in: B. Schölkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods – Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 169–184 (chapter 11).