DOI: http://dx.doi.org/10.1109/I2MTC.2013.6555517

Computational Intelligence Models for Solar Radiation Prediction

S. Ferrari, M. Lazzaroni, and V. Piuri
Università degli Studi di Milano
Milan, Italy
Email: {stefano.ferrari, massimo.lazzaroni, vincenzo.piuri}@unimi.it

A. Salman
Doğuş University
Istanbul, Turkey
Email: asalman@dogus.edu.tr

L. Cristaldi, and M. Faifer
Politecnico di Milano
Milan, Italy
Email: {loredana.cristaldi, marco.faifer}@polimi.it

***Abstract*—The modeling of solar radiation for forecasting its availability is a key tool for managing photovoltaic (PV) plants and, hence, is of primary importance for energy production in a smart grid scenario. However, the variability of the weather phenomena is an unavoidable obstacle in the prediction of the energy produced by the solar radiation conversion. The use of the data collected in the past can be useful to capture the daily and seasonal variability, while measurement of the recent past can be exploited to provide a short term prediction. It is well known that a good measurement of the solar radiation requires not only a high class radiometer but even a correct management of the instrument. In order to reduce the cost related to the management of the monitoring apparatus, a solution could be to evaluate the PV plant performance using data collected by public weather station installed near the plant.**

In this paper, two computational intelligence models are challenged; two different ground global horizontal radiation dataset have been used: the first one is based on the data collected by a public weather station located in a site different to that one of the plant, the second one, used to validate the results, is based on data collected by a local station.

I. Introduction

Nowadays, in order to activate actions related to the respect of Kyoto Protocol, some energetic scenarios are becoming strategic and are object of study. The rational use of the energetic resources, the study of the environmental impact of pollutant emissions and the exhaustion of non-renewable resources put the accent on sustainable energy production from renewable sources. Photovoltaic (PV) systems can be considered one of the most widespread solutions to the generation from renewable resources that are able to guarantee a low environmental impact [1].

Today a large variety of photovoltaic generators, from low power devices to large power plants, are in operation all over the world. The most common applications of PV systems are developed in industrial and domestic contexts. For this reason, the penetration of photovoltaic sources as distributed grid-connected power generation systems has increased dramatically in the last decades.

The solar radiation is one of the most available energy resources and the photovoltaic power conversion is an interesting exploitation of this energy. Despite the practically unlimited availability, the direct conversion in electric energy is still characterized by a relatively low efficiency and high cost. For this reason relevant efforts are performed in the research fields and in the manufacturing processes in order to achieve efficiency levels as high as possible. Like every complex system, the efficiency of a photovoltaic plant results from the combination of the efficiency of each component and the bottle neck is the low efficiency of the panel.

In order to guarantee the correct level of efficiency, the knowledge of solar radiation is mandatory. In fact, its knowledge allows to realize two tasks that are very important in a smart grid scenario. The first one, is represented by the capability of the model system to predict the energy production [2] and, the second one is represented by the capability of the model system to assess the dependability of the plant [3].

It is well known that a good measurement of the solar radiation requires not only a high class radiometer but even a correct management of the instrument. In fact, the radiometer has to be managed following a correct maintenance policy. In order to reduce the cost related to the management of the monitoring apparatus devoted to the acquisition of the solar radiation, a solution could be to evaluate the PV plant performance using data collected by public weather station installed near the plant but in a different location. The use of these data is attractive because they are often free and certified, if the station belongs to a network of public bodies.

In this paper a novel approach to condition monitoring technique has been proposed starting from the evaluation of data collected by public weather stations. In previous works [4][5], several models have been challenged in the task of predicting the global horizontal illuminance, while in the present work, instead the global horizontal radiation, will be considered. In particular, a 3-year hourly dataset will be used to model the time series of the global horizontal radiation. The prediction operated by the two computational intelligence models, namely the Support Vector Regression (SVR) and the Extreme Learning Machine (ELM) will be compared with a naïve predictor, the persistence model, and a simple predictive model, the k-Nearest Neighbor (k-NN) model.

II. The 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, the observation are related to a phenomenon that varies with the time, and the observations are taken in equally spaced instants.

Both the SVR and the ELM paradigms can model an mapping between an input and an output space, from only a finite set of input-output pairs (possibly affected by error), called training set. Time series can be modeled as a mapping between some previously observed values and the value to be predicted. For instance, when using a two-dimensional input space, the training dataset will be composed by triples of the form (x_{t-2}, x_{t-1}, x_t), where $\hat{x}_t = f(x_{t-2}, x_{t-1})$ will be assumed to approximate x_t. The two paradigms uses a linear combination of basis functions (usually Gaussians) to modeling the mapping:

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

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. 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 parameters (L, $\{\mu_i\}$, $\{\sigma_i\}$, $\{\beta_i\}$, b) from the training set.

A. Support Vector Regression

Support Vector Machines (SVM) is a powerful method for classification [6][7] and regression [8]. 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 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 loss function, which can be chosen such that the optimization problem results to be convex. The loss function is 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 the 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.

B. Extreme Learning Machines

Neural networks constitutes a very variegated class of models for classification and function approximation [9][10]. Among these, the Radial Basis Function (RBF) networks are very used, because of their simplicity and approximation power. In fact, they enjoy the universal approximation property (i.e., for every continuous function, exists an RBF network that approximates the considered function arbitrarily well). The Extreme Learning Machine (ELM) is a RBF with a fixed architecture and randomly assigned hidden nodes parameters [11][12]. 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 $\{(x_j,\, y_j) \,|\, x_j \in \mathbb{R}^D,\, y_j \in \mathbb{R},\, j = 1,\, \ldots,\, N\}$, the output of the ELM network (1) give rise to N equations that can be expressed in matricial notation as:

$$H\beta = \hat{Y} \tag{2}$$

where H is a $N \times L$ matrix such that $H_{j,i} = G(x_j; \mu_i,\, \sigma_i)$, $\beta = [\beta_1 \cdots \beta_L]^T$, and $\hat{Y} = [\hat{y}_1 \cdots \hat{y}_N]^T$. Given the training dataset and the hidden neurons parameters, the weights β are the only unknown of the linear system described in (2), and, under mild conditions, they can be computed as:

$$\hat{\beta} = (H^T G)^{-1} H^T \hat{Y} = H^{\dagger} \hat{Y} \tag{3}$$

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 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 (3) is an optimal solution in the least square sense, and has the smallest norm among the least square optimal solutions.

C. Persistence

The persistence 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}, i.e., $f_{\text{P}}(x_t) = x_{t-1}$. 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: any other model is supposed to perform better than the persistence model.

D. k-Nearest Neighbor Interpolator

The k-Nearest Neighbor (k-NN) model is a instance-based or lazy learning paradigm used both for function approximation and classification [13]. It is used to predict the value of a function, f, in unknown points, given a sampling of the function itself (training data), $\{(x_i,\, y_i) \,|\, y_i = f(x_i)\}$. 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 most simple scheme, often used in classification, estimates $f(x)$ as the most

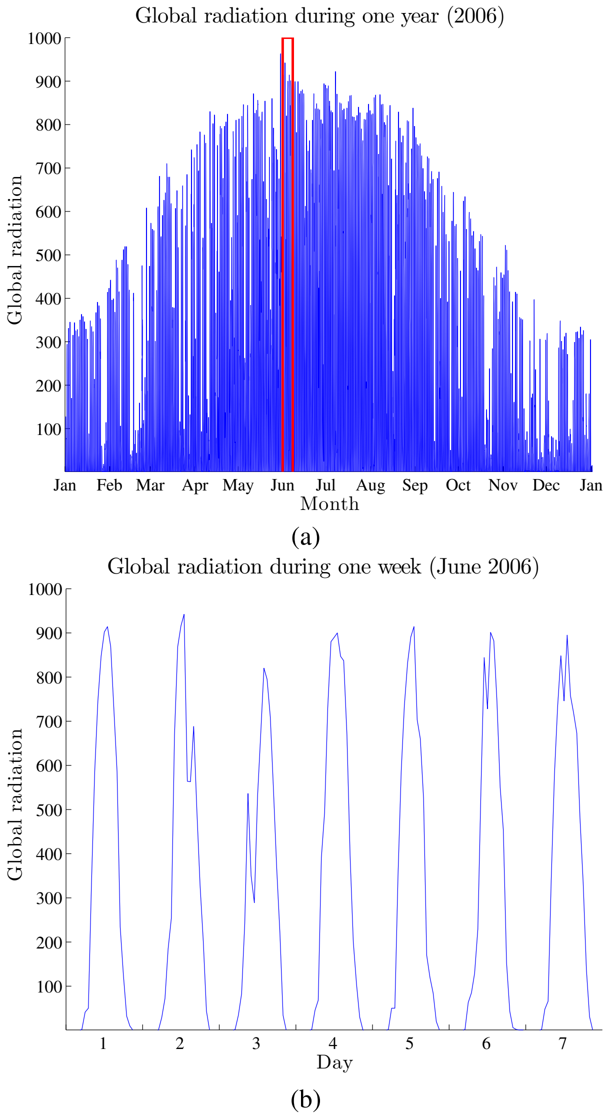


Fig. 1. One year (a) and one week (b) of the measured global horizontal radiation. Note the trend in the year and in the day, but also the strong variability in the intraday values.

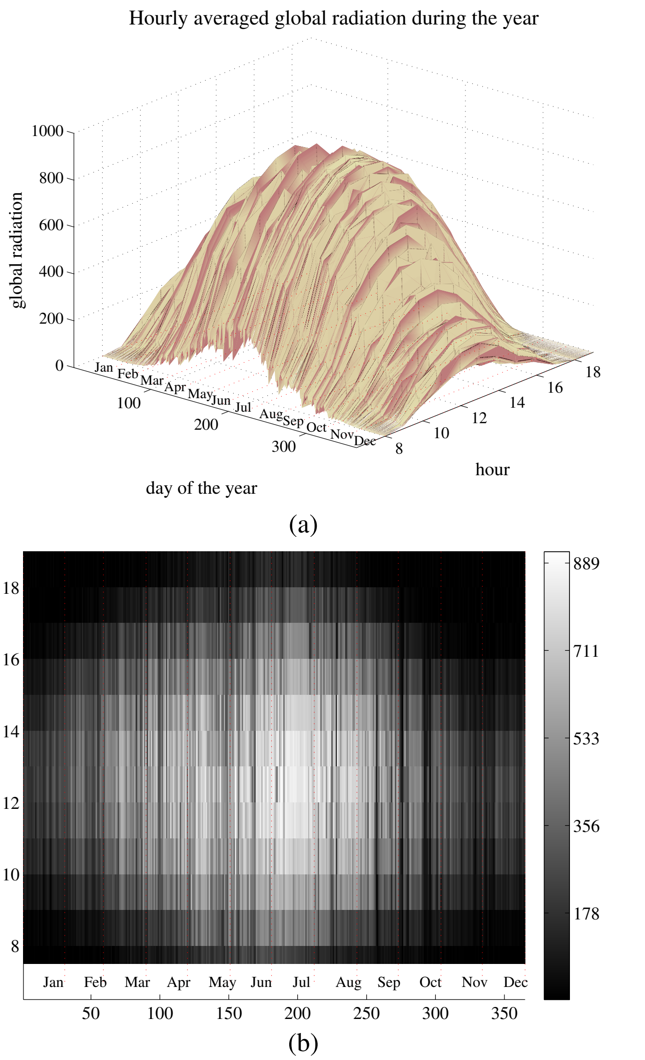


Fig. 2. The average global radiation for each day of the year and hour have been is plotted as a surface. The roughness of the surface is due to variability, although a clear trend of the phenomenon can be acknowledged.

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.

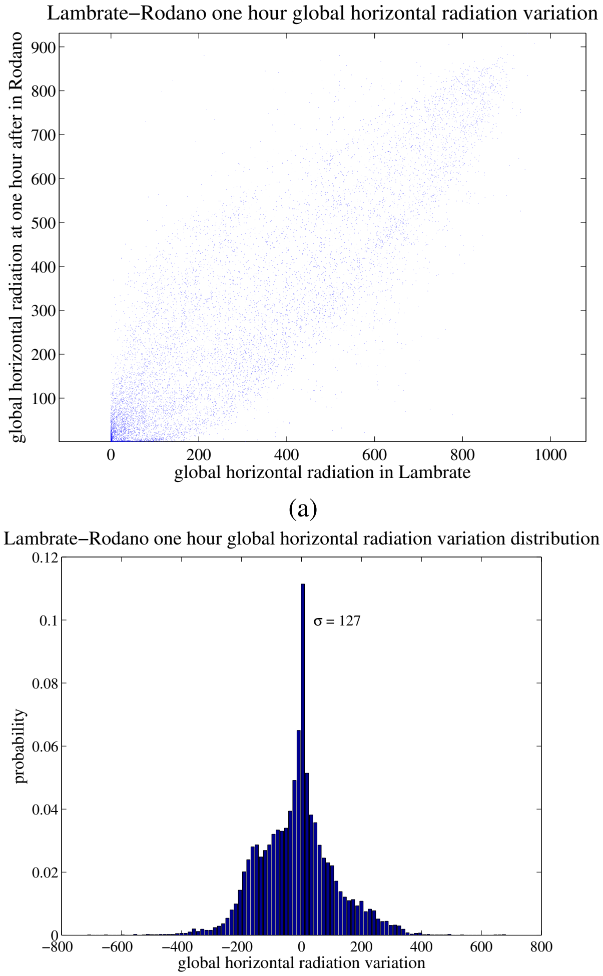
III. Experimental Activity

For the experiments described in the present paper, two datasets collected by ARPA Lombardia [14] between October 2005 and September 2008 has been used. The datasets contain the hourly measurement of the global radiance in two sites (Lambrate and Rodano, Italy) separated by about 10 km.

Subsets of the available samples are reported in Fig. 1, where Fig. 1a describes the global radiation measured in the year 2006, while in Fig. 1b only one week is reported (the first week of June). Regularities are apparent both in the yearly and in the daily scale, but also large deviations from the average behavior are possible, due to meteorological variability.

As shown by surface reported in Fig. 2, 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. A clear trend is apparent, but the variability of the global horizontal radiation (which depends also by fast changing meteorological phenomena) makes the surface very wrinkled.

Figure 3, instead shows the relation between the global horizontal radiation acquired at two consecutive hours at the two sites. In particular, in Fig. 3a 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: in fact, it can be estimated as the length of the vertical section of the cloud



(a)

(b)

Fig. 3. 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).

of points, whose thickness is at least 350. The histogram in Fig. 3b 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 is almost the same (especially in the winter). Hence, the consecutive samples acquired in those period of time are quite similar, while the other moments of the day show a larger variability.

A. Dataset Pre-Processing

Since our work require 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 22961 samples which have been used for composing the input vectors for the prediction as described in Sect. II. 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 $D \in \{1, \ldots, 10\}$, taken at time $\{t-D, \ldots, t-1\}$, 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.

B. Performance Evaluation

For the evaluation of the performances, only the daylight hours data (from 8 to 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:

$$\text{Err}(f) = E(|x_t - f(x_t)|) \tag{4}$$

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

C. Prediction through k-NN Models

Since the k-NN predictor does not requires other training process than just storing the training values, all the hyperparameters of a k-NN predictor operate in the prediction stage. In particular, the behavior of the k-NN predictor is ruled by the number of 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 hyperparameters has been challenged:

$$k \in [1, 30] \quad \text{and} \quad D \in [1, 10] \tag{5}$$

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.

D. 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 has been chosen in $[1, 10]$ (5), while networks of several sizes, L, have been challenged:

$$L \in \{10,\ 25,\ 50,\ 100,\ 250,\ 500,\ 1000,\ 2000,\ 3000\} \tag{6}$$

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 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. 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:

$$r \in \{0.01,\, 0.05,\, 0.1,\, 0.5,\, 1\} \qquad (7)$$

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 $\sigma/3$).

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.

E. 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 have to be assessed by cross validation.

The hyperparameters values that we challenged are:

- the input dimensionality, D: $[1, 10]$, as in (5);
- the accuracy, ε: $\{0.01,\, 0.1,\, 0.5,\, 1\}$;
- the regularization trade-off, C: $\{0.1,\, 1,\, 10,\, 100\}$;
- the width, σ: similarly to the ELM case, the proportionality factor r in (7) has been experimented for setting σ depending on D.

IV. Results and Discussion

The persistence, k-NN, and ELM predictors have been coded in Matlab, while for the SVR models we used the SVM*light* [15], and their performances evaluated using the prediction error, Err(f), described in (4). Since the persistence predictor configuration does not need any hyperparameters, the whole dataset described in Section III-A 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, 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 I, the persistence predictor has achieved an error $\mathrm{Err}(f_{\mathrm{P}}) = 95.4$. 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 that the 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.

TABLE I
Test error achieved by the predictors.

Predictor	Err(f) (std)	Err(f^*)
Persistence	95.4 (84.2)	—
k-NN	41.4 (57.0)	53.1
ELM	42.7 (57.0)	58.5
SVR	40.5 (59.3)	57.2

TABLE II
Test error achieved by the ELM predictor.

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

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

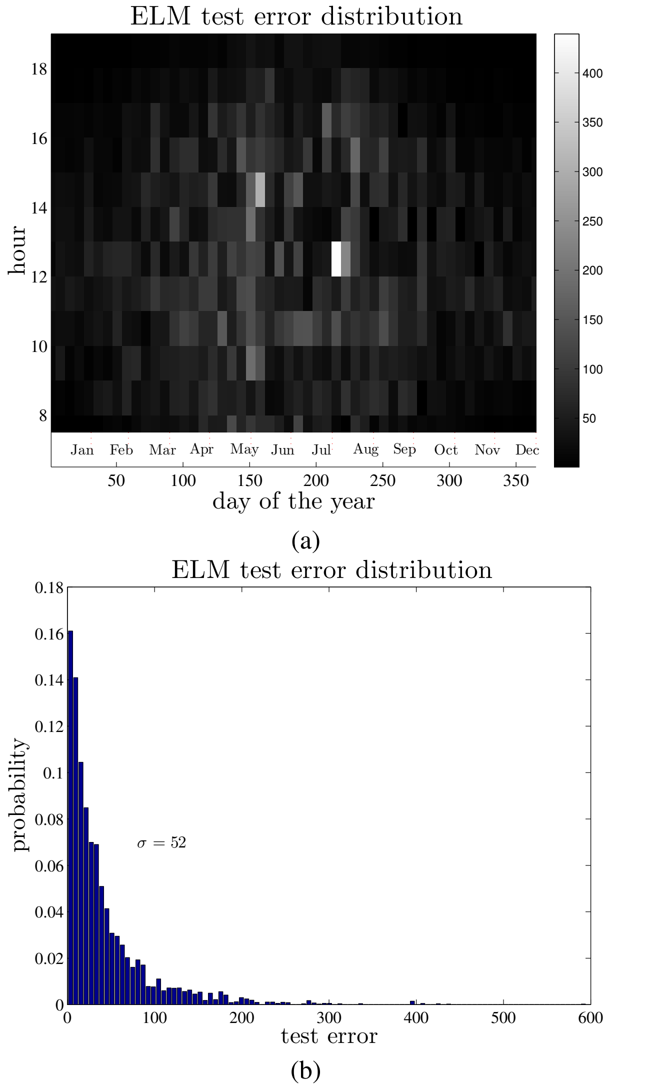
The best ELM model, which achieved an error of $\mathrm{Err}(f_{\mathrm{ELM}}) = 42.7$, resulted the one trained using the following combination of hyperparameters: $D = 2$ $r = 0.1$, $L = 500$, and using the A_2 distribution for choosing the centers position. The performance achieved in each of the five trials for this model is reported in II, with their average and standard deviation. This last value witnesses the stability of the learning.

The lowest error has been obtained by the best SVR model, which achieved an error of $\mathrm{Err}(f_{\mathrm{SVR}}) = 40.5$, using $L = 3853$ support vectors. The training has been realized with $D = 2$, $r = 0.1$, $\epsilon = 0.01$, 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 Fig. 4 and 5, respectively. Since the test set does not include all the possible time combinations, the error have been reported averaging those of seven consecutive days. It can be noticed that both the distribution are very similar (although the SVR distribution is slightly smoother); the error is reasonably low in the 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 have been reported in Table I, as $\mathrm{Err}(f^*)$. It can be noticed that the error significantly improves. Moreover, the dimension of the input space also increases ($D = 7$, for all the models).

Although all the models achieve a similar accuracy (especially if the error is compared with the maximum of the global radiation that is about 1000), if the computational cost is taken into consideration, the predictors show different properties: since the k-NN stores all the training data, it requires 5711 parameters, while the ELM 500, and SVR 3853.



(a)

(b)

Fig. 4. 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).

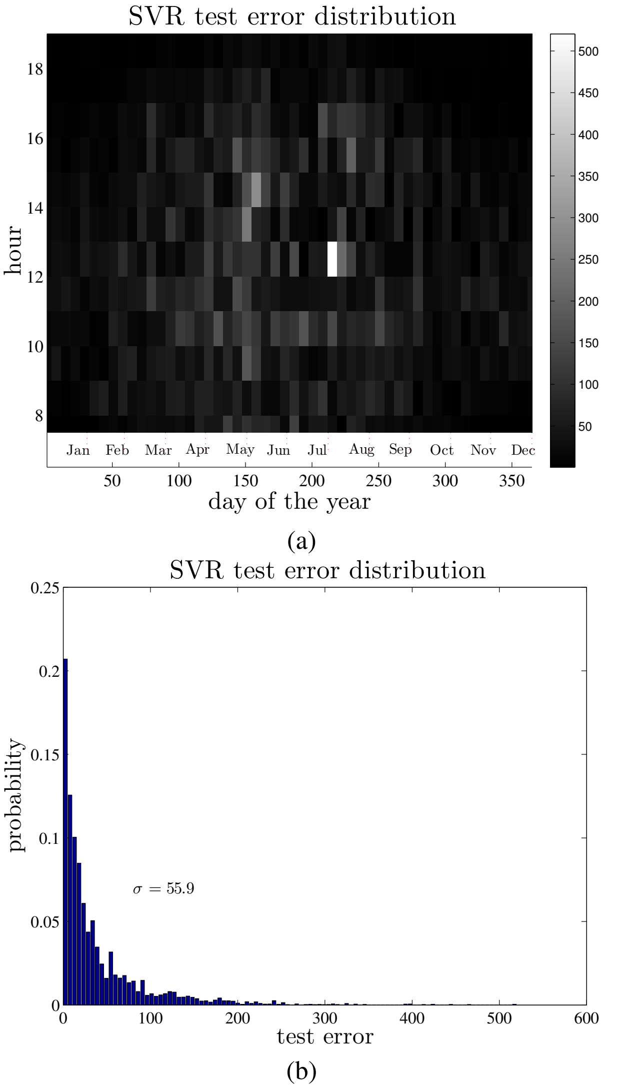
V. Conclusions

All the models challenged (k-NN, ELM, and SVR) have achieved a similar testing error, with also a similar distribution. SVR seems to offer the best compromise between the accuracy and the computational cost in term of space, although the computational time required for its training has been larger than the other models.

References

[1] M. Catelani, L. Ciani, L. Cristaldi, M. Faifer, M. Lazzaroni, and P. Rinaldi, "FMECA technique on photovoltaic module," in *IEEE Int. Instrumentation and Measurement Technology Conf. (I2MTC 2011)*, May 2011, pp. 1717–1722.

[2] V. Kostylev and A. Pavlovski, "Solar power forecasting performance — towards industry standards," in *1st Int. Workshop on the Integration of Solar Power into Power Systems*, Aarhus, Denmark, Oct. 2011.



(a)

(b)

Fig. 5. 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).

[3] L. Cristaldi, M. Faifer, M. Rossi, and F. Ponci, "A simple photovoltaic panel model: Characterization procedure and evaluation of the role of environmental measurements," *IEEE Trans. on Instrumentation and Measurement*, vol. 61, no. 10, pp. 2632–2641, 2012.

[4] F. Bellocchio, S. Ferrari, M. Lazzaroni, L. Cristaldi, M. Rossi, T. Poli, and R. Paolini, "Illuminance prediction through SVM regression," in *Environmental Energy and Structural Monitoring Systems (EESMS), 2011 IEEE Workshop on*, Sep. 2011, pp. 1–5.

[5] S. Ferrari, M. Lazzaroni, V. Piuri, A. Salman, L. Cristaldi, M. Rossi, and T. Poli, "Illuminance prediction through extreme learning machines," in *2012 IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS)*, 2012, pp. 97–103.

[6] V. N. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[7] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[8] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199–222, 2004.

[9] L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, ser. Prentice Hall international editions.

Prentice-Hall, 1994.

[10] C. M. Bishop, *Neural Networks for Pattern Recognition.* New York, NY, USA: Oxford University Press, Inc., 1995.

[11] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006.

[12] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *Neural Networks, IEEE Transactions on*, vol. 17, no. 4, pp. 879–892, Jul. 2006.

[13] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Trans. on*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[14] ARPA Lombardia. [Online]. Available: http://ita.arpalombardia.it/

[15] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999, ch. 11, pp. 169–184.