Illuminance Prediction through Extreme Learning Machines

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, M. Rossi, and T. Poli
Politecnico di Milano
Milan, Italy
Email: {loredana.cristaldi, tiziana.poli}@polimi.it, marco4.rossi@mail.polimi.it

***Abstract*—Planning, managing, and operating power grids using mixed traditional and renewable energy sources requires a reliable forecasting of the contribution of the renewable sources, due to their variable nature. Besides, the short-term prediction of the climatic conditions finds application in other fields (e.g., Climate Sensitive Buildings). In particular, this work is related to the solar radiation forecasting, that affects the photovoltaic production. The variability of the weather phenomena and climate features make the prediction a difficult task. In fact, the amount of solar radiation that reaches a particular geographical location depends not only by its latitude, but also by the geographical characteristics of the region that can create local climate conditions. In order to capture such variability, the data collected in the past can be used. Several sources can provide the data needed for the prediction (satellite and ground images, numerical weather predictions, ground measurement stations) with different resolution in time and space. In this paper, a new learning paradigm, the Extreme Learning Machine, is used to train a neural network model for the prediction of the solar illuminance. The neural networks are challenged on a two-year ground solar illuminance dataset measured in Milan, and the results are compared with those of simple predictors and results in literature.**

I. Introduction

Renewable energies will play a role of increasing importance due to the evidence of the environmental impact of the use of traditional fossil materials and decreasing of their availability, also the general trend of the energy policies which is going toward a mixed sources policies. In the renewable energy sources scenario, a major role is played by the photovoltaic (PV) technology, which allows to obtain electric energy from solar radiation [1]. Besides, the appealing of this production technology lies in the possibility of exploiting also small otherwise-unused production sites, such as the roof of the buildings; part of the produced energy can be used for local needs, and the rest can be injected in the grid. The main weakness of this renewable energy source is its dependency by astronomical and weather phenomena that make its availability uncontrollable.

In fact, the solar radiation that reaches a given site is affected by different factors. The most important are: its geographic position, which involves the potential total amount of solar radiation that can be irradiated on the site, with daily and yearly seasonality; the weather, since the clouds shields part of the radiation that effectively reaches the ground; and the local climate, which describes the peculiar attitude to cloud formation and persistence.

According to [2], the grid operators needs solar radiation forecasts with different time and space granularity: short-term (intra-hour, hour ahead, and day ahead) forecasts, for grid management activities; medium-term (months ahead) forecasts, for planning and assets optimization; and long-term (years ahead) for planning activity such as resource assessment and site selection. Although the long-term availability can be quite easily estimated for large areas with a satisfiable accuracy, short-term localized prediction are challenging due to the high variability of the weather, which depends on many physical interconnected factors. This type of forecasting is precious for grid management. In fact, when a renewable energy source is connected to the grid, the grid operators have to compensate the power required from the grid (but not available from renewable sources) with that provided by traditional energy sources. Hence, an efficient management of the grid requires a reliable forecasting of the energy provided by renewable sources.

Since weather is the main cause of the solar radiation variability, solar radiation prediction algorithms in literature commonly make use of weather forecast data. Usually, weather forecasts are carried out using satellite images and data from the ground stations. The two main aspects for qualifying weather measurements used for weather forecast are the spatial and the time resolutions. For instance, satellite images cover

a large area, with a poor resolution (each pixel can cover several kilometers) and a very poor time resolution (hours of refresh time). On the contrary, ground stations can provide direct measurements with a high temporal resolution, but are representative of a small neighborhood of the measurement station. Hence, since direct or indirect measurements of the solar radiation are generally not available for each site of interest and at each time, approximation using data from the nearby sites should be carried out. The approximation algorithm have to consider not only the distance (in time and space) of the available data, but also the local characteristics, such as the ground morphology, which can have a direct effect on the local climate [3].

Depending on the application, different forecast granularity can be required, and different prediction paradigms can be used. This paper is focused on the one-hour-ahead forecast of the global horizontal illuminance using a two-year hourly sampling. The dataset has been acquired from October 2005 to October 2007 by the MeteoLab [3][4]. This dataset has been previously used in [5], where the forecasting has been obtained through a Support Vector Machine (SVM) model. In this paper, the problem is reframed as a time series prediction problem and a neural network, namely a single layer neural network with Gaussian neurons, trained using the Extreme Learning Machine (ELM) learning paradigm, is used to realize the forecast. Several learning scheme are challenged and their predictions are compared with a naïve predictor, the persistence model, a simple predictive model, namely the k-Nearest Neighbor (k-NN) model, and the SVM model from [5].

II. Time-series 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 and the observation are related to a phenomenon that varies with the time. A practical assumption is that the observations are taken in equally spaced instants.

A. Extreme Learning Machines

Neural networks are widely used paradigms to realize both a classifier and an approximator of a function described by means of a dataset composed of samples from the function itself [6][7]. The elements of this class of machine learning paradigms are very variegated, but are characterized by their architecture and their generalization ability: their behavior results as the composition of the activity of interconnected simple processing units (the neurons, or nodes), each computing the same parametric function (with different parameter values); a suitable algorithm (the learning algorithm) adapts the value of the network parameters to the given dataset (the training, or learning dataset), which represents a set of examples (possibly affected by noise).

The most common neural architecture is the feedforward neural network, where the neurons of the network are partitioned in several groups, called layers, and connected such that neurons of one layer are connected only to neurons of the same layer: the information flows from the first layer (called input layer) to the last (called output layer), passing through the internal layers (called hidden layers). It can be shown that Single-hidden Layer Feedforward Networks (SLFNs) enjoy the universal approximation property (i.e., for every continuous function, exists a neural network that approximates the considered function arbitrarily well). Radial Basis Function (RBF) networks are SLFNs where neurons implement a radial symmetry function. The output function of a RBF network for approximating $\mathbb{R}^D \to \mathbb{R}$ functions is represented by

$$f_{\text{RBF}}(x) = \sum_{i=1}^{L} \beta_i g\left(\frac{||x - a_i||}{b_i}\right) \tag{1}$$

where L is the number of neurons, g is the neuron function, $a_i \in \mathbb{R}^D$ and $b_i \in \mathbb{R}^+$ are respectively the centers and the width of the neuron, while $\beta_i \in \mathbb{R}$ are the weight of the connection of the i-th neuron with the output node.

The Extreme Learning Machine (ELM) is a SLFN with a fixed architecture and randomly assigned hidden nodes parameters [8][9][10][11]. In particular, with the model described in (1), the parameters $\{a_i\}$ and $\{b_i\}$ 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) will be:

$$\hat{y}_j = f_{\text{RBF}}(x_j) = \sum_{i=1}^{L} \beta_i g\left(\frac{||x_j - a_i||}{b_i}\right) \tag{2}$$

for $j = 1, \ldots, N$, where $\hat{y}_j$ is the network output for x_j, thus approximating y_j. In matricial notation, the N equations of (2) can be expressed:

$$G\beta = \hat{Y} \tag{3}$$

where G is a $N \times L$ matrix such that $G_{j,i} = g(x_j; a_i, b_i)$, $\beta = [\beta_1 \cdots \beta_L]^T$, 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 (3), and, under mild conditions, they can be computed as:

$$\hat{\beta} = (G^T G)^{-1} G^T \hat{Y} = G^\dagger \hat{Y} \tag{4}$$

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

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 wasting 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 (4) is an optimal solution in the least square sense, and has the smallest norm among the least square optimal solutions.

Many variants of the ELM learning schema has been proposed in literature. Among them, at least the following worth to be cited. In [9] and [10] incremental versions have

been developed, where the neurons are added one by one, while in [12] a pruning scheme is introduced to remove those neurons with low relevance.

The ELM network can be used in time series prediction using some previously observed values for composing the input vectors. For instance, when using a two-dimensional input space ($D = 2$), the training dataset will be composed by triples of the form (x_{t-2}, x_{t-1}, x_t), where $\hat{x}_t = f_{\text{RBF}}(x_{t-2}, x_{t-1})$ will be assumed as an approximation of x_t.

B. Persistence

In order to assess the performance of the model in the short-term prediction of a time series, the persistence predictor 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}, 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.

C. k-Nearest Neighbor

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 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 complex norm for computing the distance can be used. The k-NN can be used in time series prediction using some previously observed values for composing the input vectors. For instance, when using a two-dimensional feature space, the training dataset will be composed by triples of the form (x_{t-2}, x_{t-1}, x_t), where will be assumed that $x_t = f(x_{t-2}, x_{t-1}) = w_1 x_{t-2} + w_2 x_{t-1}$, for a-priori given weight w_1 and w_2.

III. Experiments

The dataset used in the experiments described in the present paper has been collected by the MeteoLab [3][4] between October 2005 and October 2007. MeteoLab measures:

- air temperature;
- relative humidity;
- global horizontal irradiance;
- diffuse horizontal irradiance;
- global horizontal illuminance.

The station samples the data every ten minutes, but the dataset used here considers only their hourly average.

The illuminance varies both on daily and seasonal basis. The surface reported in Figs. 1a–b shows this behavior. It has been obtained by averaging the illuminance samples measured

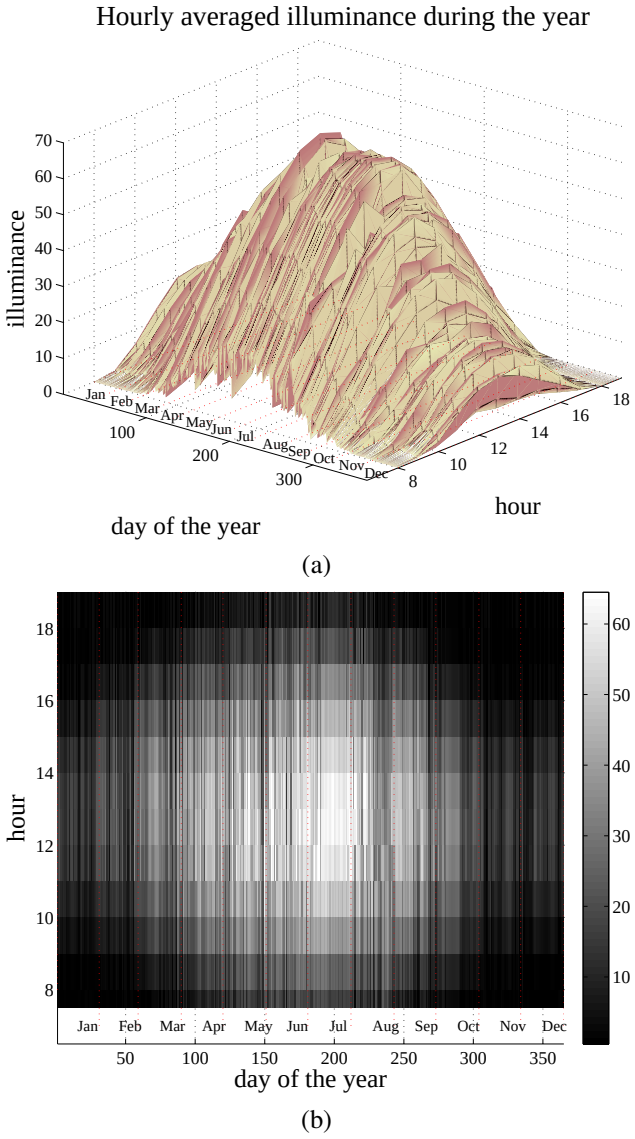


Fig. 1. The illuminance samples belonging to the same day of the year and to the same hour have been averaged and plotted as a surface. It is evident that there is a trend, but with an high variability which results in a non-smooth surface.

in the same hour of the same day of the year. Although there is a clear trend, the variability of the illuminance (which depends also by fast changing meteorological phenomena) makes the resulting surface very rough.

Figure 2, instead shows the relation between the illumination acquired at two successive hours. In particular, in Fig. 2a 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 of points, which is at least 40 long.

A. Dataset Pre-Processing

For this work, we focused only on the global horizontal illuminance (i.e., the fraction of the solar radiation that can

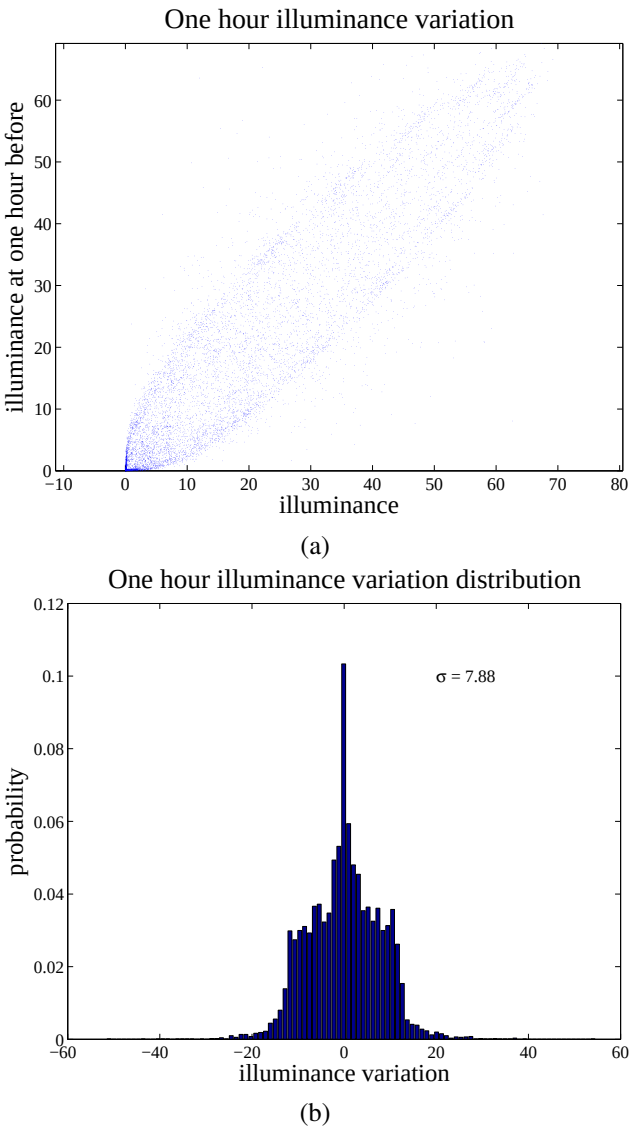


Fig. 2. The persistence predictor makes direct use of the illuminance value measured one hour before. In panel (a), the relationship between the two illuminance measurement made at distance of one hour one from the other is shown. Although the samples populate the region around the identity line, an evident dispersion is shown. In panel (b), the estimated probability density function of the variation (which standard deviation is 7.88).

be actually perceived by the human eyes). Since time series models requires that all the values are equally time spaced, 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 illuminance one hour before and ahead, and one day before and ahead are 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 18096 samples. Since the dataset covers a period of time of two years, 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 partitioned in the validation and training set. Hence, training, validation, and testing set are composed of, respectively, 9048, 4524, and 4524 samples.

B. Performance evaluation

For the evaluation of the performances, only the daylight hours data ($[8, 19]$) has been considered. Besides, since the illuminance cannot be negative, all the negative values predicted by the models are remapped to zero.

The prediction error has been measured by means of the average of the absolute error achieved on the testing set data:

$$\text{Err}(f) = E(|x_t - f(x_t)|) \tag{5}$$

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

Another performance statistics used in solar radiation prediction is the mean relative error:

$$\text{Rel}(f) = E\left(\frac{|x_t - f(x_t)|}{|x_t|}\right) \tag{6}$$

C. Prediction through k-NN models

The performance of a k-NN predictor depends on several hyperparameters. Since it 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:

- k: the number of neighbors;
- D: the number of dimension of the input space; it corresponds to the number of previous values used for the prediction;
- the weighting scheme: the law to assign the weights for the weighted averaging prediction;
- the norm of the input space.

The following values for the hyperparameters has been challenged:

$$k \in [1, 30] \tag{7}$$

$$D \in [1, 5] \tag{8}$$

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 one used for the ELM models, described in Section III-A.

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, a_i and b_i, the input space dimension, D, the number of the neurons, L, and the neuron function, g), have to be set to the proper value.

The dimensionality of the input training data, D has been chosen in $[1, 7]$ (8), while networks of several sizes, L, have been challenged:

$$L \in \{10,\ 25,\ 50,\ 100,\ 250,\ 500,\ 1000\} \tag{9}$$

Since it is the most used in literature, the only neuron function implemented has been the Gaussian function:

$$g(x;\ a,\ b) = \exp\left(-\frac{||x-a||^2}{b^2}\right) \tag{10}$$

while more efforts have been dedicated in exploring the strategies for assigning proper values to the $\{a_i\}$ and $\{b_i\}$ parameters. Since the Gaussian has a meaningful output only in a neighborhood of its center, the distribution of the centers, 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, sampling with replacement from the input training data;
- A_3, sampling without replacement from the input training data.

The width of the Gaussian, b, regulates the extent of its influence region (in regions further then $3b$ from a, the output is negligible). Since when the dimensionality of the input space increases the data becomes sparse (a problem often referred to as *curse of dimensionality*), a value of b that allows a Gaussian to cover a significant number of input examples in a given dimensionality, can be ineffective when the dimensionality of the space increases. Hence, 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 b the relative width, r, multiplied by the diagonal of the bounding box of the input training data. The value challenged for r are:

$$r \in \{0.01,\ 0.05,\ 0.1,\ 0.5,\ 1\} \tag{11}$$

Once the proper value of b has been computed for the considered dimensionality, the width of the neurons, $\{b_i\}$ are sampled from $B \sim N(b, b/3)$ (i.e., $\{b_i\}$ are distributed as a normal with mean b and standard deviation $b/3$). Since negative value are possible, but unacceptable, they are discarded and resampled. It worth noting that if all the neurons had the same width, the sampling with replacement distribution for $\{a_i\}$ has the only effect of reducing the number of neurons.

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.

IV. Results

The predictors described in Section II (i.e., the persistence, the k-NN, and the ELM models) have been coded in Matlab, and their performances evaluated using the prediction error, Err(f), described in (5). 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 and the ELM 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. The experiments have been carried out on a PC equipped with an Intel Core 2 Quad CPU at 2.5 GHz and 4 GB of RAM.

TABLE I
Test error achieved by the predictors.

Predictor	Err(f)
Persistence	6.09
k-NN	3.17
ELM	3.13
SVM	2.89
SVM [5]	2.34

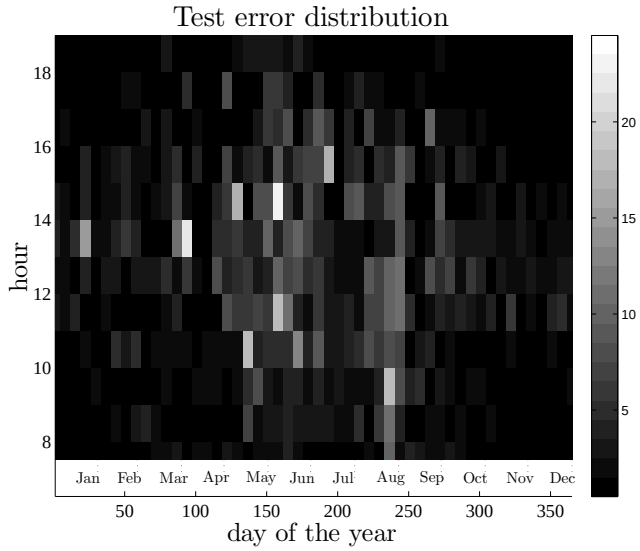
TABLE II
Test error achieved by the ELM predictor.

#trial	Err (std))	Err(f_{SVR})
1	3.17 (4.42)	
2	3.11 (4.32)	
3	3.09 (4.30)	3.13 (4.34)
4	3.14 (4.35)	
5	3.12 (4.34)	

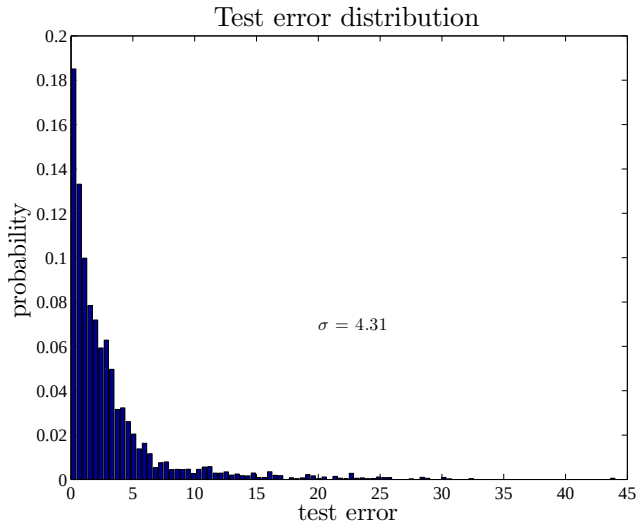
As reported in Table I, the persistence predictor has achieved an error Err(f_{P}) $= 6.09$, while the k-NN achieved and error Err($f_{k\text{-NN}}$) $= 3.17$, for $D = 4$, $k = 18$, and using the inverted distance weighting scheme. The best ELM model, which achieved an error of Err(f_{ELM}) $= 3.13$, resulted the one trained using the following combination of hyperparameters: $D = 5$ $r = 0.5$ ($b = 77.4$), $L = 100$, and using the A_3 distribution for choosing the centers position. The performance achieved in each of the five trials for this model has been reported in II, where also the standard deviation of the prediction error has been reported. It can be noted that the figure is quite stable.

In Fig. 3, the structure of the error for the best ELM model is represented with respect to the period of the year and the hour of the prediction and as an histogram. From Fig. 3a, it can be noted that the highest errors belong to the central region, i.e., in the period of the year and the day when the illuminance is higher. However, the distribution seems enough uniform. This means that the predictor cover all the region of interest. The histogram in Fig. 3b shows that a large majority of the prediction has been very close to the true value, although few large exception are present.

For the sake of comparison, the prediction error obtained using Support Vector Machines (SVM) as predictor [5] is also reported. It worth noting that in that work the randomization of the dataset has been different from the one used in the



(a)



(b)

Fig. 3. Test error distribution. In panel (a), the test error computed in all the trials is reported with respect to the day of the year and the hour. Since the test set does not include all the possible time combinations, the error have been reported averaging those of seven consecutive days. 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 4.31).

present paper. In particular, in the present work the data used for training and for assessment (validation and testing) belong to two different time intervals (one year for training, the following year for assessment), while in [5] the dataset has been randomly equally partitioned without any consideration for the time span. Besides, also the information used are different, since in [5] the input variables were: the day of the year, the hour, the illuminance of the previous hour, and the average illuminance of the previous day and the previous week. Here, instead, only the illuminance values have been used, in order to make easier the comparison with standard statistical tools for time series prediction. Hence, the results are only loosely comparable. In order to provide a proper comparison, some experiments have been run also with the SVM (using the LibCVM Toolkit [14]) on the datasets used for the present paper. The best performing model, as reported in Table I, achieved an error of $\text{Err}(f_{\text{SVM}}) = 2.89$; it is composed of $L = 5325$ support vectors, and has been trained using the following values for the hyperparameters: $D = 4$, $r = 0.1$, $\epsilon = 0.1$, and $C = 10$. This figure is more comparable to the error achieved with the ELM network (and, on the other hand, it can be seen as a measure of the influence of the dataset partitioning on the prediction ability).

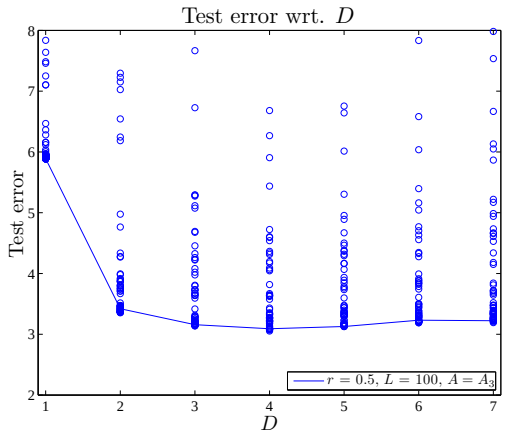
As supposed, the ELM model achieved an error well below the persistence. Since the performance obtained by the ELM network is only slightly lower than that achieved by the k-NN predictor, the ELM effectiveness can be questioned. However, it worth noting that the k-NN predictor stores 9048 training examples (and use them in the prediction), while the ELM network is composed of 100 neurons (i.e., ELM uses 1.11% units wrt. k-NN). A similar consideration applies when confronting ELM with SVM. Although the relative difference of $\text{Err}(f_{\text{SVM}}) = 2.89$ and $\text{Err}(f_{\text{ELM}})$ is quite consistent (0.24, about 8.86%), it should be put into proportion considering the span of the illuminance value, $[0, 69.2]$. Besides, the ELM network make use of 100 neurons, while the SVM is composed of 5325 units. Hence, it seems there is room for improvements.

In Fig. 4, the test error achieved with all the 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 achieved using the hyperparameter values that achieved the best validation error wit respect to the value of the considered hyperparameter is also plotted. In particular, the Fig. 4b 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 (500 and 1000) 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. This is more evident when the SVM results is considered: a large number of units allow to lower the error.

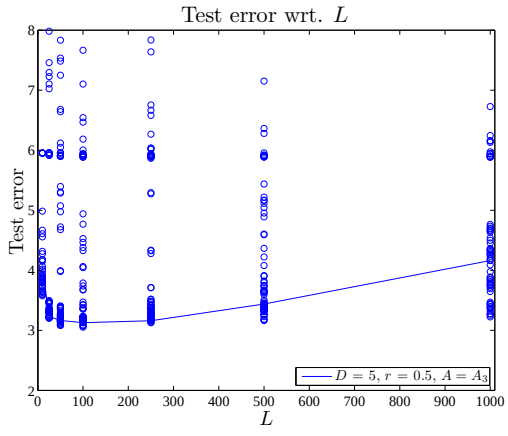
The optimal value for D, which is the number of previous data used as input, is always limited to small numbers (4, for k-NN and SVM, 5 for ELM). It is a reasonable result, since intuitively, the knowledge of the illuminance too far in the past is not very useful, since the weather can change more rapidly. On the other hand, the comparison with the results obtained in [5], suggests that other information may improve the performance: the period of the year and the hour of the prediction may be precious information for the predictor.

V. Conclusions

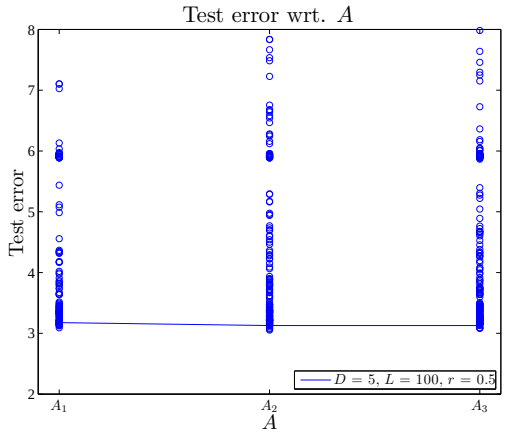
In this paper, the ELM neural network model has been challenged with a problem of time series prediction.



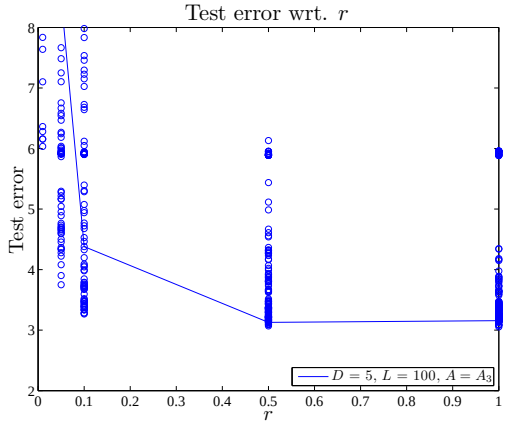
(a)



(b)



(c)



(d)

Fig. 4. Test error (averaged over five trials) wrt. the hyperparameters.

The results shows that it is able to achieve an error slightly larger than a predictor used in a previous work, the SVM, but using a fraction of the computational resources required by the SVM (an error 8.86% larger using 1.88% of the computational resources). Since larger ELM networks that have been challenged have achieved a larger error, the effectiveness of the distribution used to choose the neuron parameters have to be questioned.

Future works will explore different schemes for choosing the neuron parameters, also exploiting the use of multiscale approaches used in the literature for RBF [15]. Another research direction is the exploitation of the time and date information to improve the accuracy of the prediction.

REFERENCES

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

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

[3] T. Poli, L. P. Gattoni, D. Zappalà, and R. Gottardi, "Daylight measurement in Milan," in *Proc. of PLEA2006, Conf. on Passive and Low Energy Architecture*, 2006.

[4] ——, "Daylight measurement in Milan," in *Clever Design, Affordable Comforta Challenge for Low Energy Architecture and Urban Planning*. Geneve - CH: Raphael Compagnon & Peter Haefeli and Willi Weber, 6 2006, pp. 429–433.

[5] 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.

[6] L. Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, ser. Prentice Hall international editions. Prentice-Hall, 1994.

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

[8] 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.

[9] 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.

[10] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16–18, pp. 3056–3062, 2007.

[11] ——, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, no. 16–18, pp. 3460–3468, 2008.

[12] H.-J. Rong, Y. Ong, A.-H. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, no. 1–3, pp. 359—366, 2008.

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

[14] I. Tsang, J. Kwok, and P.-M. Cheung, "Core Vector Machines: Fast SVM training on very large data sets," *Journal of Machine Learning Research*, vol. 6, pp. 363–392, 2005.

[15] S. Ferrari, F. Bellocchio, V. Piuri, and N. A. Borghese, "A hierarchical RBF online learning algorithm for real-time 3-D scanner," *IEEE Trans. on Neural Networks*, vol. 21, no. 2, pp. 275–285, Feb. 2010.