Noname manuscript No.
(will be inserted by the editor)

Spatial Autocorrelation and Entropy for Renewable Energy Forecasting

Michelangelo Ceci · Roberto Corizzo · Donato Malerba · Aleksandra Rashkovska

Abstract In renewable energy forecasting, data are typically collected by geographically distributed sensor networks, which poses several issues. *i)* Data represent physical properties that are subject to concept drift, i.e., their characteristics could change over time. To address the concept drift phenomenon, adaptive online learning methods should be considered. *ii)* The error distribution is typically non-Gaussian. Therefore, traditional quality performance criteria during training, like the mean-squared error, are less suitable. In the literature, entropy-based criteria have been proposed to deal with this problem. *iii)* Spatially-located sensors introduce some form of autocorrelation, that is, values collected by sensors show a correlation strictly due to their relative spatial proximity. Although all these issues have already been investigated in the literature, they have not been investigated in combination. In this paper, we propose a new method which learns artificial neural networks by addressing all these issues. The method performs online adaptive training and enriches the entropy measures with spatial information of the data, in order to take into account spatial autocorrelation. Experimental results on two photovoltaic power production datasets are clearly favorable for entropy-based measures that take into account spatial autocorrelation, also when compared with state-of-the art methods.

Acknowledgements The research described in this paper has been funded by the European project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (Grant ICT-2013-612944), the European project H2020 "TOREADOR - TrustwOrthy model-awaRE Analytics Data platform" (Grant 988797) and the Startup Project "Vi-POC: Virtual Power Operation Center" (Grant PAC02L1_00269), funded by the Italian Ministry of Education, Universities and Research. The computational work has been carried out on the resources provided by the projects ReCaS (PONa3_00052) and PRISMA (PON04a2_A). The authors also wish to thank Lynn Rudd for her help in reading the manuscript.

Michelangelo Ceci · Roberto Corizzo · Donato Malerba
Department of Computer Science, University of Bari Aldo Moro, via Orabona 4, 70125 - Bari, Italy

Consorzio Interuniversitario Nazionale per l'Informatica (CINI), Italy
E-mail: {michelangelo.ceci, roberto.corizzo, donato.malerba}@uniba.it ·
ORCID: { 0000-0002-6690-7583, 0000-0001-8366-6059, 0000-0001-8432-4608 }

Aleksandra Rashkovska
Department of Communication Systems, Jožef Stefan Institute, Jamova 39, 1000 - Ljubljana, Slovenia - E-mail: aleksandra.rashkovska@ijs.si - ORCID: 0000-0002-2014-8630

Keywords Entropy · Spatial autocorrelation · Artificial neural networks · Photovoltaic power · Forecasting

1 Introduction

Sensor networks enable monitoring and study of dynamic physical phenomena on previously impossible granularity levels (Jayasumana 2009). However, when gathering and analyzing data produced by sensor networks, it is necessary to take into account that sensors are geographically distributed, they naturally produce a continuous stream of data, they act in time-changing environments and that the number of sensors can be very large and can change over time. Seeking spatial and temporal-aware information in a sensor network leads to additional computational challenges but, at the same time, creates new opportunities for storage, processing and analysis (Aggarwal 2013). Recent advances in data analytics and data mining provide techniques that can appropriately address the complex dynamics of sensor networks, i.e. processing large data volumes and accounting for spatio-temporal information (Nanni et al 2008) (Appice et al 2014).

In this paper, we consider data streams coming from sensors monitoring renewable energy plants, mainly for the purpose of forecasting energy production. Forecasting the produced energy with high accuracy represents one of the key issues in smart grid systems (Usaola et al 2004) (European Photovoltaic Industry Association 2014). The reason is that the forecasts of both the consumption and the production enable dynamic pricing models, as well as proactive control of the macrogrid network. However, mining data streams generated from sensors that measure physical properties and, in particular, mining renewable energy data, introduces additional challenges discussed in the following.

First, renewable energy production highly depends on a particular physical phenomena, e.g., wind power production from wind speed and photovolatic (PV) power production from solar irradiance. Moreover, some physical properties (like wind speed and solar irradiation) show the concept drift phenomenon, i.e., their characteristics change over time (Bessa et al 2009). For this reason, it is of fundamental importance to consider the issue of concept drift when applying data mining methods in sensor networks. In fact, adaptive models typically produce better predictions in presence of concept drift, although they need a continuous model update. As a consequence, many machine learning algorithms have been applied in renewable energy forecasting for learning adaptive models (Bessa et al 2009) (Bacher et al 2009) (Sharma et al 2011).

Second, when forecasting renewable energy production, the transformation of the renewable resource into power changes the statistical properties of the prediction errors (Lange 2005). Studies confirm that the shape of the error distribution from forecasting renewable energy production is typically non-Gaussian (Bludszuweit et al 2008) (Fabbri et al 2005) (He et al 2014) Dowell and Pinson (2016) Gneiting et al (2006). However, most of the existing ap-

proaches adopt the mean-square error (MSE) as a quality criterion. Minimizing the MSE during the learning phase is optimal only if the probability distribution function (pdf) of the prediction errors is Gaussian (Bishop 1995). Studies for wind power prediction have shown that for non-Gaussian pdf, entropy-based measures (Principe and Xu 1999b) (Principe and Xu 1999a) (Erdogmus and Principe 2002) (Morejon and Principe 2004) are more suitable for training (Bessa et al 2008) (Bessa et al 2009).

Lastly, geographically distributed power plants bring spatial dependencies (autocorrelation) in measured values with the effect of violating the classical i.i.d. (independently and identically distributed) assumption of examples. Spatial dependencies are motivated by the Tobler's first law of geography (1970) according to which "everything is related to everything else, but near things are more related than distant things". The consideration of these spatial dependencies, although requiring additional effort in the learning task, generally leads to better models (Stojanova et al 2012). Several approaches consider the spatial dimensions in stream data (Gaber et al 2005), but they do not consider the problem of non-Gaussian error distribution. In addition, they do not take the spatial autocorrelation into account and, thus, do not take advantage from the inherent spatial dependencies in the data (Borcard et al 2004).

In the method we propose, we exploit entropy-based measures for online adaptive training of Artificial Neural Networks (ANNs), when forecasting one-day ahead renewable energy production; more specifically, forecasting one-day ahead PV energy production at hourly granularity. To the best of our knowledge, there is no method for online training of ANNs which optimizes entropy and, at the same time, takes into account spatial autocorrelation of values observed by sensors. Moreover, methods for online training of ANNs, which optimize entropy for renewable power prediction, have been tested so far on wind power prediction (Bessa et al 2008) (Bessa et al 2009), but not on PV power.

The remainder of the paper is organized as follows. In the next section, we discuss the background of the work presented. Next, in Section 3, we describe the method and the entropy measures which take into account spatial autocorrelation. The experimental design and key experimental questions are outlined in Section 4. Section 4 also presents and discusses the results of the empirical evaluation. Finally, we draw the main conclusions and give directions for further work in Section 5.

2 Related Work

In this work, we focus on forecasting energy produced by renewable energy plants. This task has been deeply investigated during the last years (Usaola et al 2004) (European Photovoltaic Industry Association 2014). Existing work refers to a single renewable power generation system (Rashkovska et al 2015), or refers to multiple renewable power generation systems distributed over an extended geographic area (Bacher et al 2009) (Pelland et al 2013) (He et al

2014). In the literature, several data mining solutions have been applied in the context of renewable energy power forecasting and they are generally classified as physical and statistical (and data mining) approaches. The physical approach is heavily based on Numerical Weather Prediction (NWP) (Mathiesen and Kleissl 2011) (Zhang et al 2015b), with the addition of physical considerations (e.g., orography) (Bofinger and Heilscher 2006), or sensor data (Pelland et al 2013) (Bessa et al 2015). On the other hand, the statistical approach is based on models that relate historical values with forecasted variables. They are mainly based on time series (Chakraborty et al 2012), but there are also approaches that learn adaptive models from data e.g., autoregressive (AR) models (Bacher et al 2009), Vector autoregression (VAR) models (Bessa et al 2015) (Dowell and Pinson 2016) (Cavalcante et al 2017) (Gneiting et al 2006), artificial neural networks (ANNs) (Kalogirou 2000) (Rashkovska et al 2015), or Support Vector Machines (Sharma et al 2011). Adaptive models are considered to produce better predictions regarding concept drift, but recently, combinations of statistical (ANN and SVM) and physical approaches have also been investigated (Buhan and Cadirci 2015). Other studies perform the forecasting task relying on the analysis of sky and cloud images (Marquez and Coimbra 2013) (Chu et al 2013) (Yang et al 2014). The comparison and the assessment of the aforementioned classes of approaches for renewable energy forecasting can be found in several comprehensive studies (Jebaraj and Iniyan 2006) (Kleissl 2013) (Inman et al 2013) (Lauret et al 2015) (Pedro and Coimbra 2012) (Hong et al 2016), whereas (Zhang et al 2015a) focuses on the assessment of metrics for solar power forecasting.

However, most of the existing approaches ignore the spatial information of the plants, even when it is easily accessible: they mainly generate forecasting models that do not consider sites' proximity (or even work on single plants). In this work, we show that this information loss may result in lower predictive capabilities of the models. We propose to learn forecasting models from data related to multiple plants by leveraging the spatial autocorrelation that characterizes geophysical phenomena, such as irradiance and cloud coverage.

Several works in the literature consider multiple plants taking into account spatial autocorrelation (Bacher et al 2009) (Lorenz et al 2008) (Pelland et al 2013) (Bessa et al 2015) (Cavalcante et al 2017) (Tastu et al 2014) (Dowell and Pinson 2016) (Gneiting et al 2006) (Ceci et al 2017). Specifically, some works exploit the information of sites in the vicinity. For instance, in (Gneiting et al 2006), geographically dispersed meteorological observations in the vicinity of a wind farm are used as off-site predictors. In (Dowell and Pinson 2016), spatio-temporal dependencies are captured using a sparse parametrization of VAR models, which retains coefficients linking sites that exhibit spatial co-dependence and discards those that do not. Similarly, in (Bessa et al 2015), the authors propose a spatio-temporal model, based on the VAR framework fitted with Recursive Least Squares and Gradient Boosting. In (Cavalcante et al 2017), a set of different sparse structures for the VAR model are explored using the least absolute shrinkage and selection operator (LASSO) framework. In (Tastu et al 2014), the authors propose a conditional parametric model for

tracking spatio-temporal dependencies, based on the assumption that the local forecasting error made at time t at the target location depends on the errors previously observed at a set of neighboring sites.

On the contrary of these methods, our method considers non-linear dependencies existing between the feature space (weather conditions) and the target space (observed production), which is not the case with auto-regressive algorithms that usually train a model exclusively based on the target space. By exploiting one-day-ahead weather forecasts as independent variables of the model, it is possible to provide valuable information when weather conditions are changing over time. This potentially leads to an increased predictive accuracy of the model. Methodologically, our method incorporates spatial weighting factors in entropy-based optimization criteria, depending on the pairwise distance between plants. This allows to explicitly exploit the spatial dependencies between plants during the training phase. In (Ceci et al 2017), the idea is to use feature construction methods and apply off-the-shelf learning methods (based on the MSE training criteria) for forecasting. In this paper, we include the spatial autocorrelation into entropy-based measures for training models, which has not been done previously. As discussed in the previous section, the use of entropy-based measures has been motivated by the non-Gaussian distribution of errors when forecasting wind or photovoltaic power production.

Concerning entropy-based measures for training models, Information Theoretic Learning (ITL), introduced by Principe (Principe and Xu 1999b) (Principe and Xu 1999a), deals with entropy (as a measure of information content) while learning models. In renewable energy forecasting, ITL has been used for the first time on wind parks in Portugal (Bessa et al 2008) (Bessa et al 2009). Renyi's entropy (Rényi 1976), integrated with a Parzen windows estimation of the error distribution (Parzen 1962), has been used in the formulation of three ITL criteria (minimum entropy, maximum correntropy and the combination of both) for training neural networks for wind power prediction. When comparing the MSE criterion with the entropy-based criteria, the results showed that adopting entropy, instead of MSE, as a performance criterion leads to better predictions (in terms of higher frequency of errors close to zero and insensitivity to outliers). However, as discussed before, in the literature there is no approach which takes into account spatial autocorrelation while learning neural networks with entropy-based criteria. Moreover, while the use of entropy-based measures has already been investigated in the field of wind power forecasting, there is no study that has employed such measures in the field of PV power forecasting.

Another related research field to this topic is that of data mining methods which take spatial autocorrelation into account. Initial studies in this field are based on the SAR model (spatial autoregressive), defined as:

$$\hat{e}_i = \rho \sum_{j=1}^{N} w_{ij} e_j + \epsilon_i \; i = 1, \; \ldots, \; N, \tag{1}$$

where N is the number of training observations, $e_j = Y_j - \overline{Y}$ is the prediction error for the average, w_{ij} represents the spatial proximity between i and j, ρ expresses the spatial dependence, and ϵ_i is the error that follows a normal distribution.

For the specific task of learning predictive models, (Zhao and Li 2011) have proposed a decision tree learning algorithm that replaces a traditional entropy-based measure with the "spatial entropy" (Li and Claramunt 2006). This measure evaluates the dispersion of the entropy over the neighborhoods. Decision trees are also used in (Rinzivillo and Turini 2007) where, however, the spatial entropy is computed for each example as the weighted information gain of overlapping examples.

For the regression task, a well-known way to take spatial autocorrelation into account is GWR (Geographically Weighted Regression) (Fotheringham et al 2003). In GWR, a linear regression model is associated to each point $(u,\ v)$. In this way, the weighting of an example is not a constant, but depends on $(u,\ v)$. Formally:

$$y(u,\ v) = \alpha_0(u,\ v) + \sum_k \alpha_k(u,\ v)x_k(u,\ v) + \epsilon_{(u,v)}, \tag{2}$$

where $\alpha_k(u,\ v)$ is estimated from measurements close to $(u,\ v)$.

The idea of local models which use autocorrelation is also used in Kriging (Bogorny et al 2006), where an optimal linear interpolation method is used to estimate the response values $y(u,\ v)$ at each site $(u,\ v)$. Linear interpolation takes into account: a structural component, which represents a constant trend (average), a random component (spatially correlated), and noise. Finally, in (Ceci and Appice 2006), the authors propose a spatial associative classifier that simultaneously learns spatial association rules and a classifier (which exploits association rules). For regression, (Malerba et al 2005) presents a regression method that captures both global and local spatial effects of the predictive attributes in the learning phase.

3 Method

In this section, we present our approach for online learning of ANNs to forecast one-day ahead renewable energy production. However, before providing technical details, we check the preconditions for the application of the MSE criteria, that is, the Gaussian distribution of the prediction errors. Afterwards, we present the entropy measures that consider spatial autocorrelation and, finally, we present the online learning algorithm.

3.1 Entropy-based measures in renewable power prediction: preconditions

In (Bessa et al 2009), the authors motivated the use of entropy-based measures for training ANNs in the context of wind power forecasting with the

non-Gaussian distribution of the wind speed error. In particular, the authors considered wind speed predictions for one wind park (gathered from the NWP MM5 metereological mesoscale wind speed/direction model) for the year 2005 against real wind speed values measured by the metering station at the wind park. Then, they performed a Kolmogorov-Smirnov test, which rejected the null hypothesis of Gaussian distribution of the wind speed prediction error.

For photovoltaic data, we followed the same approach. In particular, we considered a real world photovoltaic dataset and computed two prediction errors: one obtained by considering the irradiance observed by the sensors against the irradiance predicted by the NWP PVGIS[1] model and another obtained considering the irradiance observed by the sensors against the average historical irradiance observed by the sensors at the same hour of the same month of the same year. Moreover, we computed the power production prediction error, obtained using ANNs (we used the algorithm RPROP+: `http://www.heatonresearch.com/encog/`).The prediction error histograms compared to normal distributions are shown in Fig. 1. We then performed the Kolmogorov-Smirnov normality tests and all the tests rejected the null hypothesis of Gaussian distribution of the irradiance and the power prediction error, with very high significance (p-value < 0.0001). As it can be seen from the figures, the distributions involved in our work are skewed, flatter than the Gaussian distribution and characterized by very high frequency of errors close to zero. This motivated the investigation of entropy-based measures in the context of photovoltaic data, as a potential way to improve the predictions of the power produced by the plants.

3.2 Entropy with Spatial Autocorrelation

The basic approach in a training procedure (when learning from examples) is to find a mapper/model between the output and input variables by optimizing the parameters of some learning algorithm based on some performance criterion. For example, when training ANNs, in each iteration, we adjust their weights based on a performance criterion that is some kind of estimation of the prediction error. Fig. 2 illustrates the generic training procedure of an ANN.

The performance criteria considered in this paper, as in (Bessa et al 2009), are the three ITL criteria, defined as follows:

– **Criterion 1**: Minimum error entropy (MEE) - the fundamental ITL criterion that minimizes the entropy, which is equivalent to maximizing the information potential V (Principe and Xu 1999b) (Principe and Xu 1999a) (Bessa et al 2009):

$$MEE(\epsilon) \Leftrightarrow maximize V = maximize \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G(\epsilon_i - \epsilon_j,\ 2\sigma^2), \qquad (3)$$

[1] `http://re.jrc.ec.europa.eu/pvgis/`

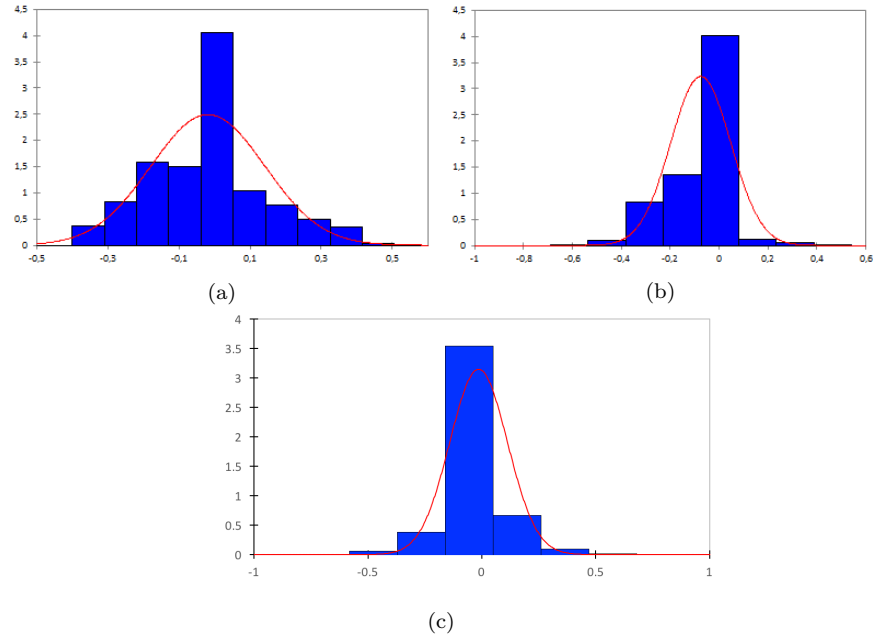

Fig. 1: Prediction error histograms and normal distribution (curve): (a) irradiance sensors vs. PVGIS, (b) irradiance sensors vs. average historical irradiance sensor data, (c) actual power vs. predicted power using ANNs.

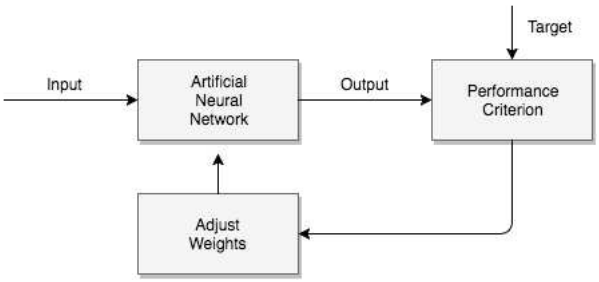

Fig. 2: Basic training procedure of an Artificial Neural Network

where ϵ_i is the error of sample i, ϵ_j is the error of sample j, G is a Gaussian kernel function, N is the number of data points and σ is the kernel size.

– **Criterion 2**: Maximum correntropy (MCC) – an approximation criterion in terms of an entropy concept based on a similarity measure, called correntropy (Barbounis and Theocharis 2007):

$$MCC(\epsilon) \Leftrightarrow maximize \frac{1}{N} \sum_{i=1}^{N} G(\epsilon_i,\ \sigma^2). \tag{4}$$

– **Criterion 3**: Minimum error entropy with fiducial points (MEEF) – a compromise between minimizing entropy and maximizing correntropy:

$$MEEF(\epsilon) \Leftrightarrow maximize\left[\gamma\frac{1}{N}\sum_{i=1}^{N}G(\epsilon_i,\ \sigma^2)+(1-\gamma)\frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}G(\epsilon_j-\epsilon_i,\ 2\sigma^2)\right], \tag{5}$$

where γ is a weighting constant between 0 and 1.

Criterion 1 is based on Renyi's entropy definition and on a representation of the error pdf by the Parzen window method (Parzen 1962). When combining Renyi's entropy definition with an estimate of a pdf by the Parzen window method, it has been shown that the practical evaluation of the entropy can be done using the information potential of the dataset, by simply calculating the Gaussian function values of the vector distances between pairs of samples (Principe and Xu 1999b,a). This, according to (Erdogmus et al 2002) and (Bessa et al 2009), implicates that the information potential V_{k+1} at time $k+1$ of the error can be iteratively estimated, according to Equation 6. This formula takes into account a Parzen-Window Density Estimation over the time dimension. In this way, when we have a new observation, the error of the neural network is computed and added to a time window with L errors of previous predictions. Formally, the formula can be written as:

$$V_{k+1} = (1-\lambda)\cdot V_k + \frac{\lambda}{L}\sum_{i=k-L+1}^{k} G(\epsilon_i - \epsilon_{k+1}, 2\sigma^2), \tag{6}$$

where L is the size of a the Parzen window P, λ is the forgetting factor with values between 0 and 1, and G is a Gaussian kernel function with a variance $2\sigma^2$. This approach guarantees that a single window of L most recent errors is taken into account for the minimization of the MEE. The consideration of the Parzen window, in principle, provides additional (historical) information to be used during minimization and, consequently, makes predictions more robust to overfitting. Minimization is based on the classical gradient descent approach.

In our method, we follow the same principle when exploiting spatial information. In particular, we modify the MEE criterion by adopting a kernel function that, in pairwise evaluation of examples, smooths the contribution of examples to be considered in the model by weighting their vicinity.

$$MEE(\epsilon) \Leftrightarrow maximize\ \frac{1}{|P|}\sum_{p\in P}\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{|N(p)|}\cdot\sum_{q\in N(p)}\left(1-\frac{dist(p,q)}{\max Dist(P)}\right)G(\epsilon_i^q-\epsilon_j^q,\ 2\sigma^2). \tag{7}$$

In this formula, N is the number of time points, P is the set of plants, $N(p) \subset P$ is the set of the neighbourhood plants[2] of the plant p, ϵ_i^p is the error for the

[2] In our formulation, the neighborhood $N(p)$ includes the considered current plant p, i.e. $p \in N(p)$.

plant p at the i-th time point, $dist$ is the distance between two plants, and $maxDist(P)$ is:

$$maxDist(P) = \max_{p,q \in P, p \neq q} dist(p,\ q). \tag{8}$$

Consequently, the update of the information potential is:

$$V_{k+1}^{MEE} = (1-\lambda)V_k^{MEE} + \tfrac{1}{|P|}\sum_{p\in P}\frac{\lambda}{L}\sum_{i=k-L+1}^{k}\left[\frac{1}{|N(p)|}\sum_{q\in N(p)}\left(1-\frac{dist(p,q)}{\max Dist(P)}\right)G(\epsilon_i^q - e_{k+1}^q, 2\sigma^2)\right]. \tag{9}$$

It is noteworthy that Equation (9) is similar to Equation (6), with a correction of the kernel, which allows us to give priority to the maximization of the information potential of spatial areas with a high density of plants and, consequently, reduce the importance of the maximization of the information potential for low density spatial areas. The rationale is that, in this way, we are able to make smoother predictions in high-density regions. The purpose is to make the method more robust to overfitting and, consequently, increase the predictive capabilities of the system.

Concerning Criterion 2, we can modify the optimization defined in formula (4), by considering the spatial component:

$$MCC(\epsilon) \Leftrightarrow maximize \tfrac{1}{|P|}\sum_{p\in P}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|N(p)|}\sum_{q\in N(p)}\left(1-\frac{dist(p,q)}{\max Dist(P)}\right)G(\epsilon_i^q,\ \sigma^2). \tag{10}$$

This formula, like formula (7), provides a correction of the kernel, which allows us to give priority to the maximization of the information potential of spatial areas with a high density of plants and, consequently, reduce the importance of the maximization of the information potential for low density spatial areas. Moreover, we can easily make formula (10) incremental and exploit the Parzen window to provide additional (historical) information to be used during optimization:

$$V_{k+1}^{MCC} = (1-\lambda)V_k^{MCC} + \tfrac{1}{|P|}\sum_{p\in P}\frac{\lambda}{L}\sum_{i=k-L+1}^{k}\left[\frac{1}{|N(p)|}\sum_{q\in N(p)}\left(1-\frac{dist(p,q)}{\max Dist(P)}\right)G(\epsilon_{k+1}^q, \sigma^2)\right]. \tag{11}$$

Similarly to criteria 1 and 2, also criterion 3 can be modified, in order to take into account spatial information:

$$MEEF(\epsilon) \Leftrightarrow maximize \frac{1}{|P|}\sum_{p\in P}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|N(p)|}\sum_{q\in N(p)}\left(1-\frac{dist(p,q)}{\max Dist(P)}\right)\cdot$$
$$\cdot\left[\gamma G(\epsilon_i^q, \sigma^2) + (1-\gamma)\frac{1}{N}\sum_{j=1}^{N}G(\epsilon_j^q - \epsilon_i^q, 2\sigma^2)\right]. \tag{12}$$

We can also make formula (12) incremental and exploit the Parzen window:

$$V_{k+1}^{MEEF} = (1-\lambda) \cdot V_k^{MEEF} + \frac{1}{|P|} \sum_{p \in P} \frac{\lambda}{L} \sum_{i=k-L+1}^{k} \left[\frac{1}{|N(p)|} \cdot \right.$$
$$\left. \cdot \sum_{q \in N(p)} \left(1 - \frac{dist(p,q)}{\max Dist(P)}\right) \left(\gamma \cdot G(\epsilon_{k+1}^q, \sigma^2) + (1-\gamma) \cdot G(\epsilon_i^q - \epsilon_{k+1}^q, 2\sigma^2)\right)\right]. \quad (13)$$

In all the formulae that consider spatial autocorrelation (formulae (9), (11) and (13)), we use the same definition of spatial proximity: $\left(1 - \frac{dist(p,q)}{\max Dist(P)}\right)$. The benefit of such a function is two-fold: (1) it avoids the problem of choosing a distance threshold to identify which plants are included in the local neighbourhoods, which usually implies discarding the contribution of plants that are more distant than the specified threshold; (2) for a specified plant, each neighbour's contribution is weighted proportionally to its closeness with respect to the specified plant under analysis, avoiding the naïve solution that gives an equal weight to each neighbour.

From the point of view of the algorithm, in order to properly address the spatial component in formulae (9), (11) and (13), $|P|$ Parzen windows (of size L) are employed, one for each plant p in the network. This choice adds a cost in space which is minor, considering that we only have to group examples per site. However, the management of the neighbourhoods in the algorithm adds a cost in time, which is linear with the number of plants in the network. This problem can be mitigated by pre-computing the values of the G function to be used in the inner loop. In the case of correntropy, the computation of V_{k+1}^{MCC} requires additional time, introduced by the Parzen window: this represents a real difference in terms of time complexity with respect to the original computation of the maximum correntropy.

From a more theoretical viewpoint, it is important to verify that the cost function satisfies the differentiability requirement for gradient descent learning. In this respect, it has been demonstrated that entropy and correntropy based cost functions are differentiable and they can be easily wrapped into the backpropagation framework to train any nonlinear system (in particular neural networks) with gradient descent learning (Principe 2010). Since the cost functions proposed in this paper adapt the original formulations of entropy and correntropy with a spatial smoothing factor that considers the pairwise distance between plants, it follows that the differentiability property is still guaranteed for the cost functions proposed, since the weighted sum of differentiable functions is still a differentiable function. The same applies for the MEEF criterion.

3.3 On-line learning algorithm

The training methodology follows a self-adaptive online training approach. For training purposes, the neural network is initialized with random weights

and the training data are passed through the network (feed-forward phase). For each training example, the difference between the predicted value and the expected value is stored in the specific Parzen window of the plant p, from which the example originated. Then, the training phase takes place (at the end of each day), according to the standard backpropagation scheme, and the error in the network is calculated and updated, according to one of the chosen criteria: MEE (3), MCC (4), $MEEF$ (5), MEE_{SA} (7), MCC_{SA} (10), $MEEF_{SA}$ (12).

For each learning session (at the end of each day), the optimization process iterates until the error starts to grow for two consecutive epochs, or until a maximum number of epochs is reached (800 in our experiments). Comparing our approach with that implemented in (Bessa et al 2009), instead of using a validation set chosen independently from the training set, we evaluate the behavior of the error only on the training set. In fact, the use of a fixed validation set (e.g. a fixed month or period in the year, as done in (Bessa et al 2009)), leads to the problem of overfitting the model to the climatic conditions of that specific period. Moreover, the validation set has to be independent of the training set, but this leads to the problem of removing useful data for training.

The topology of the ANN is composed of three layers: an input layer with $I_n = |I|$ neurons, defined by the cardinality $|I|$ of input features (such as temperature, irradiance, windspeed, humidity, ...); an output layer with $O_n = 1$ (single power prediction at a specified hour for the next day); and a hidden layer with $H_n = \frac{2}{3}.(I_n + O_n)$ neurons, as suggested in (Sheela and Deepa 2013). For training ANNs, we use the *Encog* implementation of the Resilient Propagation (RPROP+) algorithm (Heaton 2015). We use RPROP+ because it has been already successfully applied in renewable energy prediction (Bessa et al 2009) (Ceci et al 2017). We have modified the RPROP+ implementation to include, besides the MSE cost function, also MEE (3), MCC (4), $MEEF$ (5), MEE_{SA} (7), MCC_{SA} (10) and $MEEF_{SA}$ (12).

3.4 Automated parameters' tuning

As discussed before, our method requires to set up the value for the following input parameters:

- L: Parzen window size
- λ: Forgetting factor
- σ: Kernel size.

The value of L could be set depending on the specific execution time constraints and available data. However, the optimal values for λ and σ are often difficult to obtain. In order to take into account this aspect, we have introduced a function which, given two sets L_V and K_S of possible values for λ and σ (respectively), automatically performs a grid search over the different configurations, exploiting historical data during the training phase.

For this purpose, the training set is partitioned into learning set and validation set. Given a single day in the validation set, a model is learned on previous training days and evaluated on the validation day. This process is repeated ten times for each pair $(\lambda, \sigma) \in L_V \times K_S$ and the average forecasting error is collected. The effect is equivalent to a cross-validation scheme for parameter tuning, but coherent with the data stream setting, which considers only data observed in the past as training data. The function returns the list of pairs (λ, σ), involved in the grid search, ordered in ascending order by the average forecasting error obtained (e.g., Root Mean Squared Error (RMSE)), as well as the best configuration (λ_B, σ_B) found.

In conclusion, this procedure is a valuable addition to the method and provides the ability to identify the best parameter configuration for a specific dataset with no manual effort. It is particularly helpful in real application scenarios, when no prior knowledge about the data characteristics of a specific solar farm is available.

4 Experiments

In this section, we first provide a description of the datasets and clarify the objective of the learning task. Then, we describe the experimental setting, and report and discuss the results.

4.1 Data description

In our experiments, we used two datasets: a real world PV dataset, named PV Italy, collected by an Italian company (which operates in the renewable energy sector), and a simulated dataset concerning PV production in the USA, provided by the National Renewable Energy Innovation (NREL)[3], and henceforth referred to as PV NREL.

The input features considered are the geographic coordinates of the plants (lat, lon) and the weather variables *temperature, irradiance, pressure, windspeed, wind bearing, humidity, dew point* and *cloud cover*. They are queried from Forecast.io[4] for both datasets, whereas irradiance is queried from PVGIS[5] only in the case of PV Italy. Data are collected for each plant, for each hour and for each day.

The independent (target) variable $y_{i,j,h}$ is the power production of the i-th plant at time h (hour) of the day j. The task is to predict the power production for each hour of the next day (24 values are predicted in the same prediction step).

[3] http://www.nrel.gov/

[4] http://forecast.io/

[5] http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php

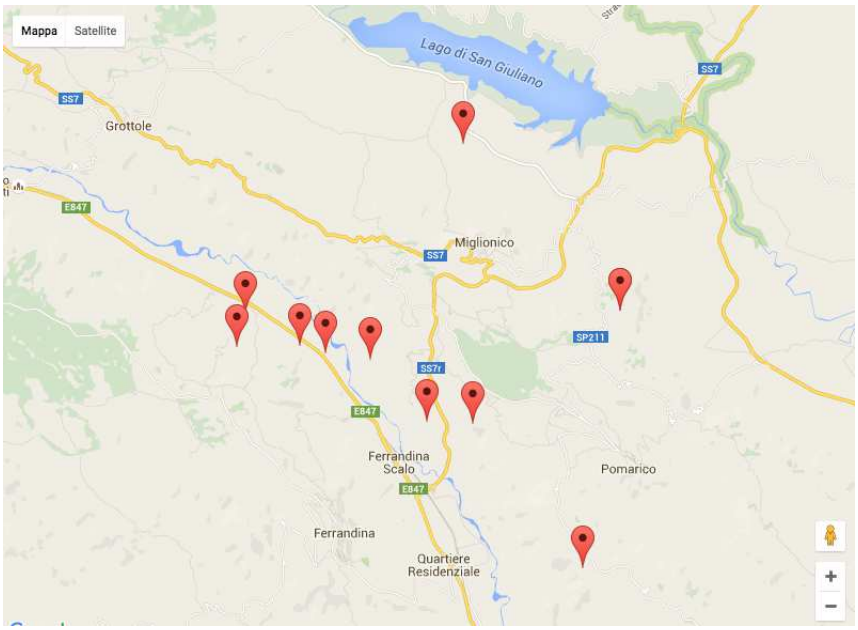

Fig. 3: Geographical distribution of the plants in the PV Italy dataset

4.2 Datasets

PV Italy data are generated every 15 minutes by sensors on 17 plants in Italy. (Fig. 3). The time period spans from January 1st, 2012 to May 4th, 2014. The original dataset contains many missing values for the features considered, due to sensor failures or communication problems, and many outliers (values outside the range defined, according to the 4-sigma "rule-of-thumb": $[\overline{x}-4\cdot\sigma_x; \mathrm{X}+4\cdot\sigma_x]$). To solve these problems, we followed the same data preprocessing steps described in (Ceci et al 2017), according to which we substituted missing values and outliers with the average of the same feature observed for the same month of the same year at the same hour.

PV NREL dataset originally consists of simulated PV data for 6000 plants for the year 2006. We performed adaptive cluster sampling (Thompson 1990) over the original dataset, by first selecting 16 states with the highest Global Horizontal Irradiation (GHI): Alabama, Arizona, Arkansas, California, Colorado, Florida, Georgia, Kansas, Louisiana, Mississipi, Nevada, New Mexico, Oklahoma, South Carolina, Texas and Utah. Then, from each state, we selected 3 PV plants, resulting in PV data from 48 plants (see Fig. 4). The data was not affected by outliers or missing values.

The description of the datasets is summarized in Table 1. All the datasets and the system are available for replication and future research purposes as a permanent repository on Zenodo: `http://doi.org/10.5281/zenodo.1242854` (Ceci et al 2018).

Table 1: Brief description of the datasets

Dataset	Plants	Days	Rows
PV Italy	17	857 (731 training, 126 test)	264689
PV NREL	48	365 (240 training, 125 test)	331968

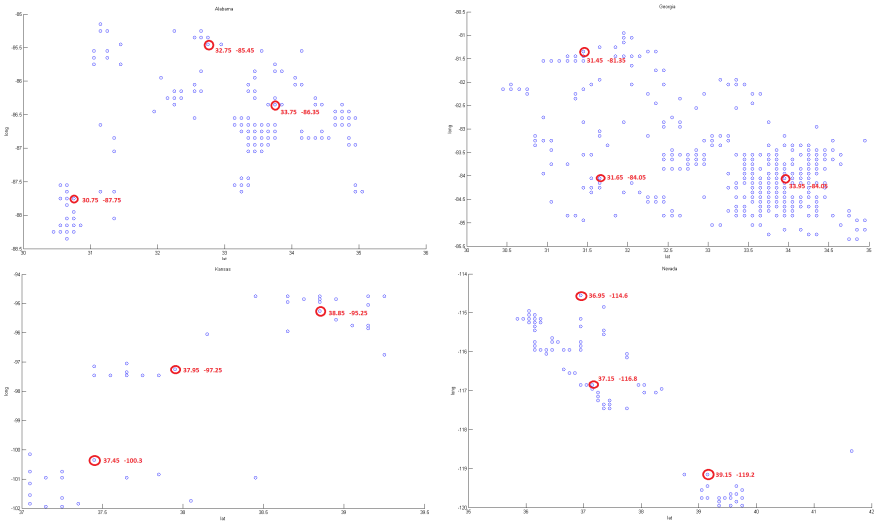

Fig. 4: Example of plant selection by cluster sampling for the PV NREL dataset.

4.3 Experimental settings

For the evaluation, we performed a random selection of days, to split data between training set (85% of days) and testing set (15% of days). For each dataset, the experiments were run five times, with different random splits into training and test days. The learning strategy was iterative - for each testing day, the model was learned on a fixed size window consisting of the data of the previous days and tested on the considered day (example(s) unseen by the trained model). After testing, the testing day becomes part of the training set. This testing-retraining procedure was repeated for each testing day and the error contributed to the reported result. For each run, the final error result was obtained by averaging the daily errors made by the neural network (which, in turn, were obtained by averaging the hourly errors).

For all the experiments, we investigated different values for the *Kernel size* σ^2, the *forgetting factor* λ and the *Parzen window size* L. The aim is to evaluate the sensitivity of the method with respect to the different values of these parameters. In particular, the following values were tested according to a grid analysis (i.e., all the combinations of possible values have been analyzed):

- *Kernel size* σ^2: 0.4, 0.5, 0.6;
- *Forgetting factor* λ: 0.04, 0.08, 0.12;
- *Parzen window size* L: 1000, 2000, 4000.

Moreover, we investigated two additional settings: *NoSpatial* – the latitude and longitude of the plant are not taken as input atributes, and *LatLon* – the latitude and longitude of the plant are taken as input attributes (in addition to spatial information implicitly included in the adopted criteria).

For the evaluation of the results, we consider Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Normalized Percentage Absolute Error (NPAE) as indicators of the predictive performance. The results for a particular training criterion are denoted with the name of the criterion, i.e., MCC, MEE and $MEEF$, while for the same criteria, when considering also spatial autocorrelation in the entropy measure, the results are accompanied additionally with the abbreviation SA, i.e MCC_{SA}, MEE_{SA} and $MEEF_{SA}$. For each criterion, we also make a performance comparison (in terms of percentage of improvement) with respect to the results obtained by the ARIMA (AutoRegressive Integrated Moving Average) model. ARIMA models are designed for time series analyses and for forecasting tasks (Box et al 2015). For this reason, we consider the ARIMA model as a good baseline. For evaluation, we used the Spark-TS library[6] to implement the ARIMA predictive task, under the same experimental conditions used for the ANNs. A utility function has been applied to perform a model search that automatically selects the best ARIMA model, based on AIC (Akaike Information Criterion) values. The process is similar to the one described in (Hyndman et al 2007).

We also compared the results obtained by our method with respect to the elastic net regularized linear regression algorithm (Zou and Hastie 2005) and the isotonic regression algorithm (Barlow and Brunk 1972). In particular, the former overcomes the limitations of the LASSO (Least Absolute Shrinkage and Selection Operator) method (which penalty function has shown to present different drawbacks (Zou and Hastie 2005)) by combining the L1 and L2 penalties of the LASSO and ridge methods, whereas the latter is capable of fitting a non-decreasing free-form line to a set of points, without making assumptions about the linearity of the target function. For both regression algorithms, we adopted the implementations available in the Spark MLlib [7]. Since the algorithm for elastic net regularized linear regression requires, as input, a regularization parameter, we performed a grid search considering the values $\{0.15, 0.3, 0.45\}$ and reported the best result obtained. No parameter is required for the Spark implementation of the isotonic regression algorithm.

Finally, we also compared our algorithms with the well-known SVR (Support Vector Regression) algorithm, also using different kernels (linear, polynomial, sigmoid). In this case, we used the implementations available in the Weka toolkit[8].

[6] `https://github.com/sryza/sparktimeseries`

[7] `https://spark.apache.org/docs/latest/mllib-guide.html`

[8] `http://www.cs.waikato.ac.nz/ml/weka/`

In order to statistically evaluate our approach, we compared the RMSE obtained with different configurations and different algorithms by considering two statistical tests: *i)* a Friedman test combined with Nemenyi post-hoc tests and *ii)* a Signed Rank Wilcoxon test.

In the former, we followed the suggestions reported in (Demšar 2006) and we plotted the graphs which summarize the results. This test is used to compare:

- the different entropy-based criteria,
- the effect of different values of kernel size,
- the effect of different values of λ,
- the effect of the size of the Parzen windows,
- all the methods considered in this study.

The latter is used to compare the spatial autocorrelation variant with the corresponding variant which does not consider spatial autocorrelation.

4.4 Results and discussion

The results for the PV Italy and PV NREL datasets, for the two settings and for all training criteria, are reported in Tables 2 and 3. The improvement of the best performing results for each *Kernel size* and *forgetting factor* λ are highlighted in bold. The results refer to executions with *Parzen window size* $L = 4000$, which is the best value for this parameter according to Fig. 6 (which is commented later). Additional results with other values for the *Parzen window size* can be found at the following link: https://zenodo.org/record/1242854#.WvArJ9OFMyk. At first inspection, the results show that the best overall performances for each dataset are always obtained by considering spatial autocorrelation: MEE_{SA} is the best criterion for PV Italy (with $Impr. = 17.71\%$) and MCC_{SA} is the best criterion for PV NREL (with $Impr. = 40.70\%$).

To provide statistical support to this conclusion, we performed the Wilcoxon Signed Rank Test in order to compare each criterion with its counterpart which uses spatial autocorrelation. The results are reported in Table 4. The (*) symbol marks the comparisons in which the newly proposed entropy-based measures, that consider spatial autocorrelation, outperform the baseline entropy-based measures, i.e., entropy measures without considering spatial autocorrelation. The results show that the entropy measures that consider spatial autocorrelation always outperform the corresponding baseline entropy measures. Moreover, for the MCC and the $MEEF$ criteria, the difference in performance is statistically significant. The important conclusion we can draw from these results is that spatial autocorrelation provides important information that allow the algorithm to improve its prediction capabilities. This is obtained by smoothing the contribution of examples to be considered in the model by weighting their vicinity. Moreover, this conclusion is valid independently of the parameter configuration adopted.

λ	0.04											
Kernel Size	0.4				0.5				0.6			
	RMSE	**MAE**	**NPAE**	**Impr.%**	**RMSE**	**MAE**	**NPAE**	**Impr.%**	**RMSE**	**MAE**	**NPAE**	**Impr.%**
MCC No Spatial	0.139	0.104	10.356	14.12	0.138	0.103	10.295	14.50	0.137	0.102	10.157	15.31
MCC Lat Lon	0.138	0.103	10.341	14.19	0.140	0.104	10.432	13.45	0.140	0.105	10.483	13.41
MCC_{SA} No Spatial	0.138	0.103	10.311	14.47	0.137	0.102	10.215	**15.01**	0.138	0.103	10.307	14.63
MCC_{SA} Lat Lon	0.139	0.104	10.372	14.00	0.139	0.104	10.409	13.80	0.139	0.104	10.415	13.78
MEE No Spatial	0.137	0.103	10.281	14.95	0.138	0.103	10.269	14.70	0.136	0.101	10.147	**15.47**
MEE Lat Lon	0.141	0.105	10.547	12.81	0.143	0.107	10.722	11.64	0.139	0.104	10.358	13.84
MEE_{SA} No Spatial	0.138	0.103	10.305	14.52	0.138	0.103	10.291	14.58	0.138	0.102	10.240	14.66
MEE_{SA} Lat Lon	0.140	0.105	10.451	13.23	0.138	0.103	10.281	14.44	0.141	0.105	10.523	12.71
$MEEF$ No Spatial	0.137	0.102	10.184	**15.30**	0.138	0.102	10.160	14.57	0.139	0.103	10.269	13.87
$MEEF$ Lat Lon	0.141	0.105	10.530	12.83	0.139	0.103	10.252	14.08	0.140	0.104	10.358	13.20
$MEEF_{SA}$ No Spatial	0.137	0.102	10.207	15.29	0.138	0.103	10.307	14.36	0.137	0.104	10.372	15.24
$MEEF_{SA}$ Lat Lon	0.139	0.104	10.408	13.65	0.141	0.107	10.716	12.75	0.139	0.106	10.582	13.56
λ	0.08											
Kernel Size	0.4				0.5				0.6			
	RMSE	**MAE**	**NPAE**	**Impr.%**	**RMSE**	**MAE**	**NPAE**	**Impr.%**	**RMSE**	**MAE**	**NPAE**	**Impr.%**
MCC No Spatial	0.137	0.102	10.197	**15.26**	0.137	0.102	10.183	15.01	0.138	0.104	10.426	14.29
MCC Lat Lon	0.140	0.105	10.532	13.06	0.140	0.105	10.484	12.94	0.142	0.108	10.760	12.03
MCC_{SA} No Spatial	0.138	0.104	10.380	14.22	0.137	0.102	10.196	15.16	0.137	0.103	10.315	14.81
MCC_{SA} Lat Lon	0.140	0.105	10.548	12.92	0.139	0.103	10.350	13.81	0.140	0.106	10.583	13.11
MEE No Spatial	0.137	0.102	10.211	14.85	0.137	0.102	10.175	14.96	0.137	0.102	10.234	15.09
MEE Lat Lon	0.140	0.105	10.476	13.35	0.140	0.105	10.466	13.45	0.142	0.108	10.839	11.79
MEE_{SA} No Spatial	0.138	0.103	10.335	14.35	0.137	0.102	10.183	15.21	0.138	0.104	10.412	14.31
MEE_{SA} Lat Lon	0.140	0.105	10.506	13.43	0.138	0.103	10.304	14.19	0.142	0.107	10.728	12.13
$MEEF$ No Spatial	0.139	0.103	10.275	14.00	0.138	0.102	10.151	14.71	0.136	0.100	10.001	15.66
$MEEF$ Lat Lon	0.139	0.103	10.347	13.67	0.140	0.104	10.363	13.37	0.138	0.103	10.284	14.25
$MEEF_{SA}$ No Spatial	0.138	0.103	10.284	14.53	0.136	0.101	10.109	**16.01**	0.135	0.101	10.089	**16.51**
$MEEF_{SA}$ Lat Lon	0.140	0.105	10.486	12.98	0.139	0.105	10.459	13.69	0.138	0.105	10.501	14.75
λ	0.12											
Kernel Size	0.4				0.5				0.6			
	RMSE	**MAE**	**NPAE**	**Impr.%**	**RMSE**	**MAE**	**NPAE**	**Impr.%**	**RMSE**	**MAE**	**NPAE**	**Impr.%**
MCC No Spatial	0.137	0.102	10.196	**15.06**	0.139	0.104	10.380	13.67	0.138	0.103	10.283	14.39
MCC Lat Lon	0.142	0.106	10.618	12.20	0.140	0.105	10.464	13.20	0.139	0.104	10.405	13.63
MCC_{SA} No Spatial	0.138	0.103	10.303	14.61	0.137	0.102	10.224	14.92	0.139	0.104	10.357	14.05
MCC_{SA} Lat Lon	0.140	0.105	10.530	13.14	0.140	0.105	10.509	13.27	0.140	0.105	10.528	12.96
MEE No Spatial	0.137	0.102	10.227	14.99	0.136	0.101	10.141	15.54	0.136	0.101	10.116	15.63
MEE Lat Lon	0.139	0.104	10.402	13.63	0.141	0.106	10.559	12.56	0.140	0.104	10.448	13.43
MEE_{SA} No Spatial	0.140	0.104	10.404	13.51	0.138	0.103	10.252	14.49	0.133	0.100	10.001	**17.71**
MEE_{SA} Lat Lon	0.141	0.106	10.555	12.84	0.140	0.105	10.545	13.00	0.138	0.105	10.461	14.29
$MEEF$ No Spatial	0.139	0.102	10.241	13.99	0.139	0.103	10.277	13.81	0.137	0.102	10.187	15.12
$MEEF$ Lat Lon	0.142	0.107	10.692	11.76	0.139	0.104	10.364	13.64	0.141	0.106	10.555	12.84
$MEEF_{SA}$ No Spatial	0.138	0.103	10.312	14.26	0.136	0.101	10.068	**15.62**	0.136	0.102	10.194	15.42
$MEEF_{SA}$ Lat Lon	0.139	0.104	10.362	14.13	0.141	0.108	10.778	12.48	0.137	0.105	10.502	14.89

Table 2: Average RMSE, MAE and NPAE (5 runs) for the PV Italy dataset, with the best performing Parzen window configuration (Parzen window size=4000). The percentage of improvement (Impr.%) considered is w.r.t. the ARIMA baseline model in terms of RMSE.

As clarified before, for a wider comparison, we performed Nemenyi tests on the results for all datasets, considering different training criteria and parameter settings:

– Training criteria considering spatial autocorrelation (Fig. 5): The test shows that the MCC_{SA} criterion outperforms $MEEF_{SA}$, which in turn outperforms MEE_{SA}. This is in line with results obtained in (Bessa et al 2009), where the MCC and $MEEF$ criteria achieved better performances than

λ	0.04											
Kernel Size	0.4				0.5				0.6			
	RMSE	MAE	NPAE	Impr.%	RMSE	MAE	NPAE	Impr.%	RMSE	MAE	NPAE	Impr.%
MCC No Spatial	0.176	0.139	13.949	31.93	0.174	0.138	13.847	32.55	0.172	0.135	13.541	33.45
MCC Lat Lon	0.160	0.123	12.321	38.06	0.160	0.124	12.448	37.94	0.154	0.118	11.799	**40.17**
MCC_{SA} No Spatial	0.174	0.138	13.799	32.55	0.173	0.137	13.653	32.99	0.176	0.140	14.016	31.70
MCC_{SA} Lat Lon	0.156	0.121	12.096	39.40	0.153	0.117	11.744	**40.70**	0.157	0.120	12.020	39.28
MEE No Spatial	0.174	0.138	13.819	32.51	0.174	0.138	13.769	32.59	0.176	0.140	14.007	31.84
MEE Lat Lon	0.160	0.123	12.317	38.14	0.158	0.122	12.248	38.70	0.156	0.121	12.075	39.40
MEE_{SA} No Spatial	0.177	0.142	14.151	31.41	0.174	0.138	13.818	32.49	0.173	0.137	13.696	33.00
MEE_{SA} Lat Lon	0.153	0.118	11.777	**40.69**	0.155	0.119	11.903	40.10	0.156	0.120	12.003	39.69
$MEEF$ No Spatial	0.176	0.139	13.890	31.90	0.174	0.137	13.735	32.59	0.177	0.141	14.120	31.49
$MEEF$ Lat Lon	0.160	0.125	12.453	37.87	0.155	0.120	11.985	39.78	0.155	0.119	11.906	40.09
$MEEF_{SA}$ No Spatial	0.174	0.137	13.701	32.60	0.175	0.139	13.890	32.07	0.176	0.138	13.774	31.66
$MEEF_{SA}$ Lat Lon	0.159	0.123	12.329	38.55	0.153	0.118	11.790	40.70	0.161	0.122	12.177	37.62
λ	0.08											
Kernel Size	0.4				0.5				0.6			
	RMSE	MAE	NPAE	Impr.%	RMSE	MAE	NPAE	Impr.%	RMSE	MAE	NPAE	Impr.%
MCC No Spatial	0.176	0.139	13.936	31.96	0.175	0.139	13.888	32.20	0.174	0.138	13.751	32.51
MCC Lat Lon	0.158	0.122	12.180	38.97	0.154	0.118	11.773	**40.46**	0.156	0.119	11.918	39.62
MCC_{SA} No Spatial	0.173	0.138	13.757	32.81	0.174	0.138	13.780	32.62	0.174	0.138	13.847	32.56
MCC_{SA} Lat Lon	0.156	0.120	11.980	39.64	0.155	0.119	11.945	39.87	0.158	0.122	12.183	38.89
MEE No Spatial	0.176	0.139	13.934	31.78	0.175	0.138	13.804	32.35	0.177	0.141	14.111	31.56
MEE Lat Lon	0.157	0.121	12.097	39.19	0.157	0.121	12.080	39.27	0.153	0.117	11.721	**40.67**
MEE_{SA} No Spatial	0.176	0.139	13.928	31.77	0.173	0.137	13.734	32.82	0.175	0.139	13.874	32.19
MEE_{SA} Lat Lon	0.157	0.121	12.091	**39.35**	0.158	0.122	12.191	38.81	0.158	0.122	12.184	38.80
$MEEF$ No Spatial	0.175	0.139	13.931	32.06	0.174	0.138	13.824	32.47	0.173	0.136	13.635	33.10
$MEEF$ Lat Lon	0.158	0.122	12.192	38.66	0.161	0.125	12.519	37.68	0.158	0.122	12.175	38.99
$MEEF_{SA}$ No Spatial	0.178	0.141	14.089	31.18	0.173	0.137	13.728	32.89	0.176	0.140	13.979	31.92
$MEEF_{SA}$ Lat Lon	0.157	0.122	12.154	39.19	0.160	0.123	12.344	37.93	0.158	0.123	12.252	38.68
λ	0.12											
Kernel Size	0.4				0.5				0.6			
	RMSE	MAE	NPAE	Impr.%	RMSE	MAE	NPAE	Impr.%	RMSE	MAE	NPAE	Impr.%
MCC No Spatial	0.172	0.135	13.550	33.38	0.176	0.139	13.910	31.89	0.175	0.139	13.910	32.04
MCC Lat Lon	0.159	0.123	12.341	38.36	0.159	0.123	12.319	38.25	0.155	0.120	11.966	**39.78**
MCC_{SA} No Spatial	0.177	0.141	14.084	31.35	0.174	0.138	13.826	32.42	0.175	0.140	13.965	32.10
MCC_{SA} Lat Lon	0.154	0.118	11.779	**40.45**	0.157	0.120	12.039	39.34	0.158	0.122	12.196	38.69
MEE No Spatial	0.175	0.138	13.829	32.30	0.174	0.138	13.769	32.57	0.174	0.138	13.789	32.61
MEE Lat Lon	0.156	0.119	11.916	39.71	0.159	0.122	12.236	38.49	0.158	0.121	12.141	38.98
MEE_{SA} No Spatial	0.175	0.139	13.855	32.21	0.177	0.140	14.019	31.54	0.177	0.140	14.048	31.62
MEE_{SA} Lat Lon	0.161	0.125	12.516	37.81	0.157	0.122	12.159	39.05	0.159	0.122	12.242	38.32
$MEEF$ No Spatial	0.174	0.138	13.761	32.71	0.175	0.139	13.946	32.04	0.175	0.140	13.959	32.05
$MEEF$ Lat Lon	0.158	0.122	12.179	38.79	0.154	0.118	11.768	**40.30**	0.156	0.120	12.022	39.50
$MEEF_{SA}$ No Spatial	0.174	0.138	13.756	32.78	0.177	0.140	14.003	31.52	0.175	0.138	13.794	32.37
$MEEF_{SA}$ Lat Lon	0.159	0.123	12.251	38.58	0.157	0.121	12.055	39.20	0.158	0.122	12.195	38.63

Table 3: Average RMSE, MAE and NPAE (5 runs) for PV NREL. For description and configurations see caption of Table 2

MCC_{SA} vs MCC	MEE_{SA} vs MEE	$MEEF_{SA}$ vs $MEEF$
0.018(*)	0.181(*)	**0.004**(*)

Table 4: Wilcoxon Signed Rank Tests (all datasets). Legend: (*) indicates that the spatial autocorrelation variant outperforms its non-spatial autocorrelation counterpart. Bold: improvement is statistically significant at $\alpha = 0.05$.

MEE, although not exploiting spatial autocorrelation. This is also in line with the results presented in (Liu et al 2007), where the MCC criteria has shown better performances than MEE in the regression setting. The theoretical motivation of this behavior is that correntropy is insensitive to the peak in the noise PDF tail, effectively handling the bulk of residuals around the origin (Liu et al 2007).

- *Kernel Size* σ^2 (Fig. 5): The selection of larger values for the *Kernel size* is favorable. The test shows that the results with *Kernel size* $= 0.6$ slightly outperform the results with *Kernel Size* $= 0.5$, which in turn outperform the results with *Kernel Size* $= 0.4$.
- Forgetting factor λ (Fig. 6): The results with $\lambda = 0.04$ and $\lambda = 0.08$ outperform the results with $= 0.12$. There is no statistically significant difference between the results with $\lambda = 0.04$ and $\lambda = 0.08$. This result shows how much we should weight the recent examples.
- *Parzen window size* L (Fig. 6): The results show that performance is increasingly better when using a wider Parzen window. In particular, results with *Parzen window size* $= 4000$ are better than the ones obtained with *Parzen window size* $= 2000$ and *Parzen window size* $= 1000$. This confirms that the consideration of a Parzen window is always beneficial.

This analysis is confirmed by the results obtained by the automated parameters' tuning procedure, which identifies the following best configurations for *Parzen window size* $= 4000$:

- PV Italy: $\lambda_B = 0.08, \sigma_B = 0.6$
- PV NREL: $\lambda_B = 0.04, \sigma_B = 0.5$.

In our analysis, we also compare our algorithms MCC_{SA}, MEE_{SA} and $MEEF_{SA}$ with competitor algorithms. RMSE values are reported in Table 5, where the elastic net regularized linear regression algorithm is identified as $LinReg$ and the isotonic regression algorithm is identified as $IsoReg$. It can be noticed that, among the competitors, SVR_{Poly} is the best performing method for both datasets: average RMSE=0.141 for PV Italy and 0.166 for PV NREL.

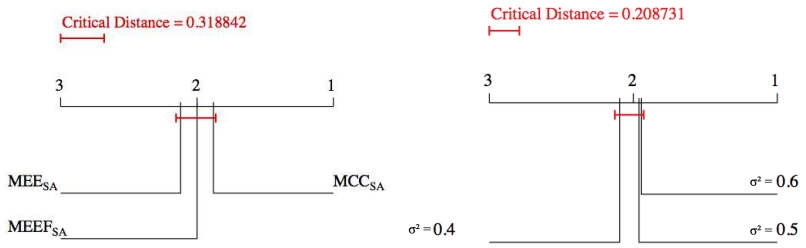

Fig. 5: Nemenyi test for different training criteria (left) and different values for the *Kernel size* σ^2 (right), considering all datasets. The best criteria and *Kernel size* values are positioned on the right.

Average RMSE performance of all algorithms

Method	$PVItaly$	$PVNREL$
LinReg	0.165	0.230
IsoReg	0.204	0.265
$SVR_{Sig.}$	0.175	0.208
$SVR_{Lin.}$	0.170	0.179
SVR_{Poly}	0.141	0.166
$ARIMA$	0.168	0.251
MCC_{SA}	0.139 (±0.002)	**0.165** (±0.010)
MEE_{SA}	0.139 (±0.002)	0.166 (±0.010)
$MEEF_{SA}$	**0.138** (±0.002)	0.167 (±0.009)

Table 5: Average RMSE for all the algorithms with the best performing Parzen window configuration (Parzen window size=4000). Results of MCC_{SA}, MEE_{SA} and $MEEF_{SA}$ are averaged over all the configurations and their standard deviation is reported in parenthesis. Bold: best performing algorithm for each dataset.

However, if we see the RMSE results obtained by the proposed algorithms, it is clear that they outperform all the competitors of a great margin.

In Fig. 7, we compare all the methods and variants using a Nemenyi test. From this visual representation of the ranks, we can see that the overall best performing model for all datasets is ANN with MCC_{SA} followed by $MEEF_{SA}$ and MEE_{SA}. If we consider the performances of other competitor algorithms, SVR_{Poly} exhibits worse performances than the criteria which use spatial autocorrelation, and similar performances with respect to the criteria without spatial autocorrelation (MCC, MEE, $MEEF$). Other algorithms ($IsoReg$, $ARIMA$, $LinReg$, $SVR_{Lin.}$ and $SVR_{Sig.}$) show statistically significantly worse performances. We also observe that all the criteria which consider spatial autocorrelation are ranked better than their counterparts (this confirms the statistical test reported in Fig. 4).

In order to inspect the forecasting error obtained with the proposed method at a finer level of granularity, we have also represented the results obtained with the best performing criterion (MCC_{SA}), in different time frames of the

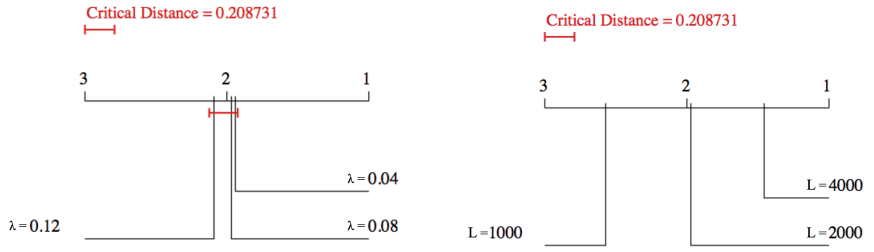

Fig. 6: Nemenyi test for different values of the forgetting factor λ (left) and the *Parzen window size* L (right), considering all datasets. The best configurations are positioned on the right.

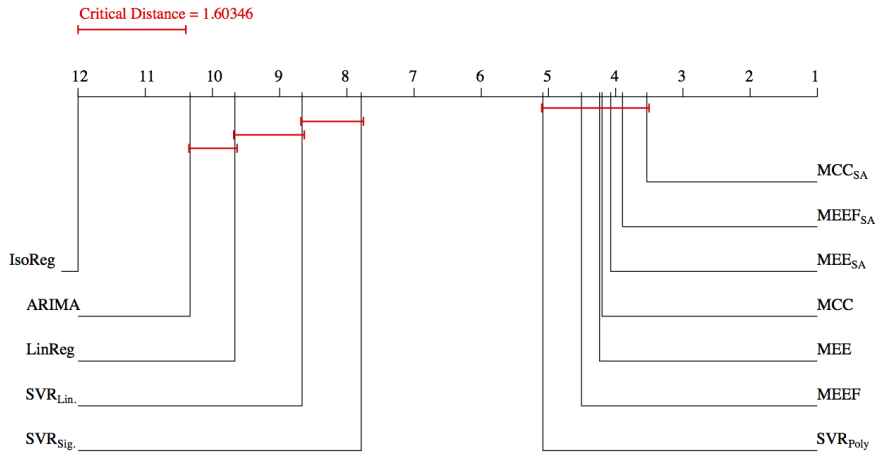

Fig. 7: Nemenyi test considering all models and all window sizes. The best configurations are positioned on the right.

day (Fig. 8, 10) and with increasing time horizons (cumulative RMSE), up to one-day-ahead (Fig. 9, 11). This brings additional insights on the impact of our learning criteria for different lead-times. In particular, it can be observed that the most challenging hours of the day, in which RMSE is higher than 0.2, are included in the range between 08:00 and 13:00 for the PV Italy dataset, and in the range between 11:00 and 16:00 for the PV NREL dataset. This is normal considering the fact that in central hours of the day the energy production is higher and, obviously, the prediction is (relatively) more error-prone.

A better analysis, reported in Figures 12 and 13, shows the accuracy of the models in terms of the coefficient of determination (R^2) over different time frames of the day. These results are obtained as (one minus) the squared ratio between the RMSE of the best performing training criterion (MCC_{SA}) and the RMSE of a baseline method, which predicts one-day-ahead energy production as the average production observed in the training data (average). The graphs show that our method is more helpful in the central hours of the day, when the production may vary significantly and can be significantly different from the simple average. This behaviour is completely different from what was observed in Cavalcante et al (2017) for wind power forecasting. The reason is that in wind power forecasting there is always some contiguity from one hour to the next hours, whereas for PV energy production, nightly hours introduce a strong discontinuity in the produced energy.

Figures 14 and 15 help to better understand this phenomenon. In these figures, we report the error reduction in percentage, observed in different time frames of the day. The results are obtained considering the relative change in percentage between the RMSE obtained with MCC_{SA} and the RMSE obtained by predictions based on historical average. They confirm that the

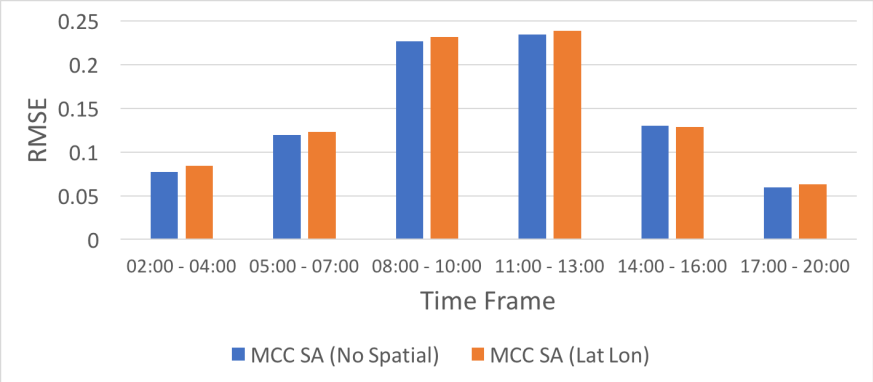

Fig. 8: Average forecasting error (RMSE) in different time frames of the day, at a three-hour granularity for the PV Italy dataset. Results are obtained with the MCC_{SA} training criterion.

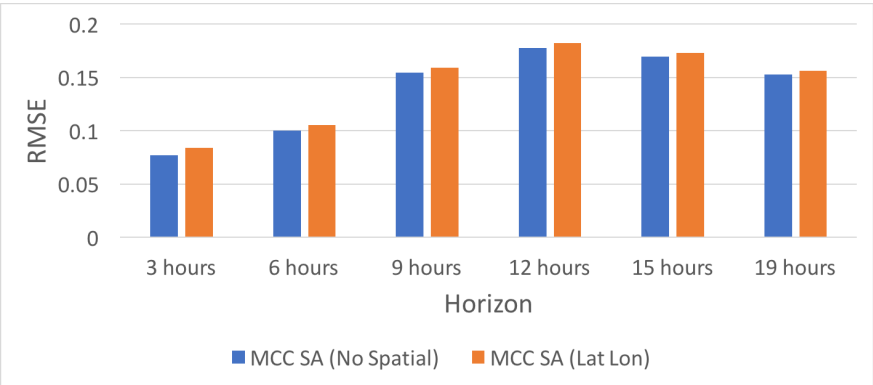

Fig. 9: Average forecasting error (RMSE) with increasing time horizons (cumulative), up to one-day ahead for the PV Italy dataset. Results are obtained with the MCC_{SA} training criterion.

biggest improvement is obtained during the central hours of the day. The reason is that the power produced increases during the hours featured by high irradiance. Therefore, the power production during these hours is challenging to predict using a baseline model, which has been trained over historical data and assumes that 1) the weather conditions remain close to the average the following day and 2) there is no spatial autocorrelation. In contrast, the baseline model is more likely to be accurate during the initial and final hours of the day, when spatial autocorrelation is less important due to reduced irradiance. The results depicted in Figures 8, 9, 10, 11, 12, 13, 14 and 15 are averaged over different configurations of λ and σ and using a Parzen window of size 4000.

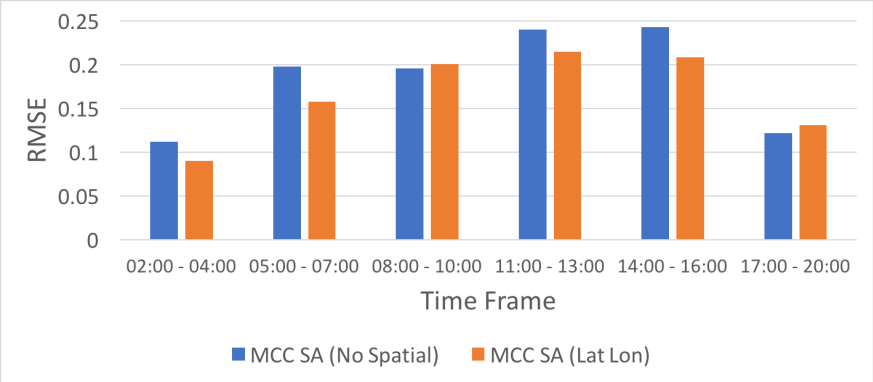

Fig. 10: Average forecasting error (RMSE) in different time frames of the day, at a three-hour granularity for the PV NREL dataset. Results are obtained with the MCC_{SA} training criterion.

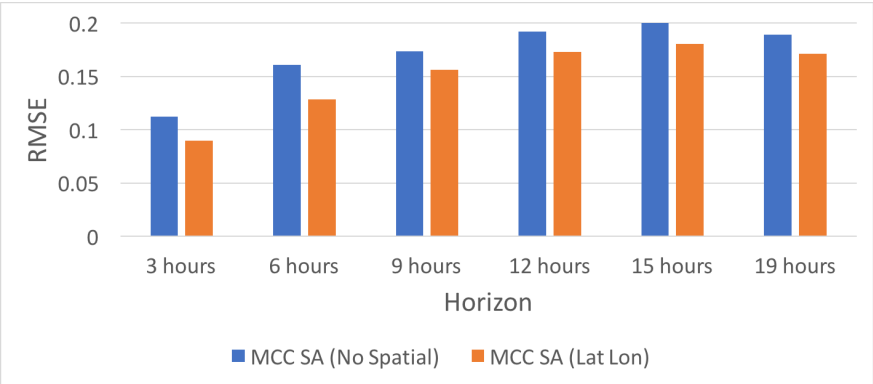

Fig. 11: Average forecasting error (RMSE) with increasing time horizons (cumulative), up to one-day ahead for the PV NREL dataset. Results are obtained with the MCC_{SA} training criterion.

Finally, we present a scalability analysis obtained with increasing values of the Parzen window size, considering the best performing training criterion. The results in terms of RMSE are reported in Figures 16 and 17, whereas. the results in terms of execution time are reported in Fig. 18. It can be observed that the best value of RMSE has been obtained with a Parzen window size of 8000 for both datasets: the RMSE obtained with a Parzen window size wider than 8000 examples, tends to become stable or increase. Moreover, the execution time required to perform training with wider Parzen window sizes, increases significantly. Therefore, we can conclude that the best trade-off between the accuracy of the model and the training time required, can be obtained with Parzen window sizes $\leq$ 8000 examples.

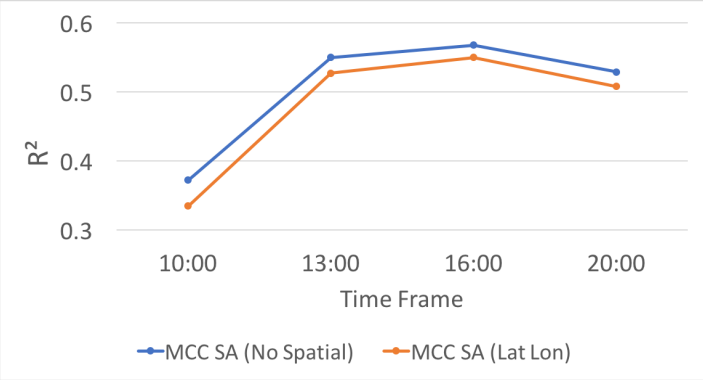

Fig. 12: Coefficient of determination (R squared) over different time frames of the day, for the PV Italy dataset. Results are obtained as the proportion between the RMSE of the best performing training criterion (MCC_{SA}) and the RMSE of a baseline average predictor.

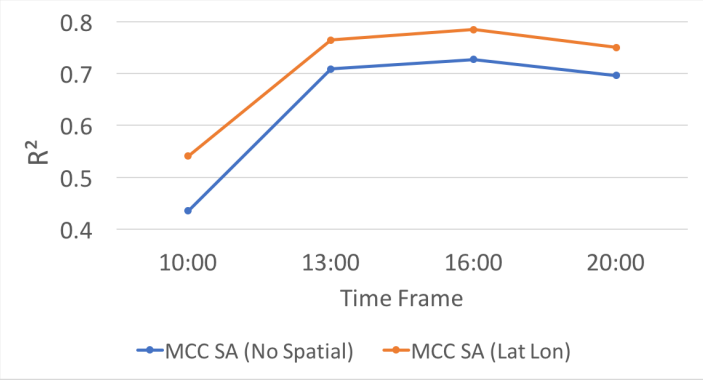

Fig. 13: Coefficient of determination (R squared) over different time frames of the day for the PV NREL dataset. Results are obtained as the proportion between the RMSE of the best performing training criterion (MCC_{SA}) and the RMSE of a baseline average predictor.

5 Conclusions

This paper targets several issues in sensor network data mining, in particular, in mining renewable energy data. To deal with the concept drift of physical properties, such as wind speed and solar irradiation, it works in an online adaptive learning setting. Next, motivated by the non-Gaussian error distribution when forecasting renewable energy production, it investigates several entropy-based criteria for online adaptive training of ANNs: MCC, MEE and $MEEF$. Moreover, it also compares the afore-mentioned baseline entropy-based criteria with their variants that consider also the spatial information of the data. Such

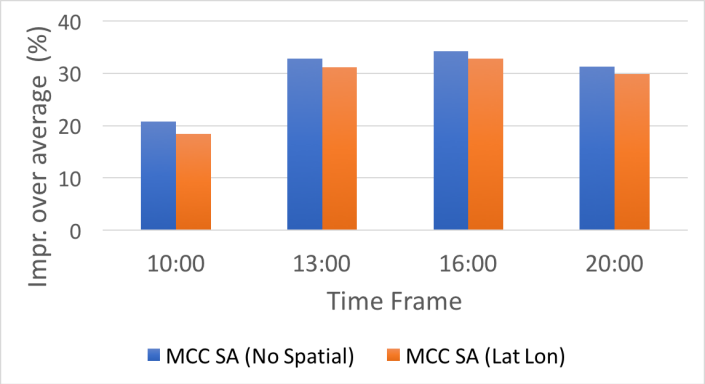

Fig. 14: Percentage of error reduction in different times frames of the day for the PV Italy dataset. Results are obtained considering the RMSE of the best performing training criterion (MCC_{SA}) and the RMSE of a baseline average predictor.

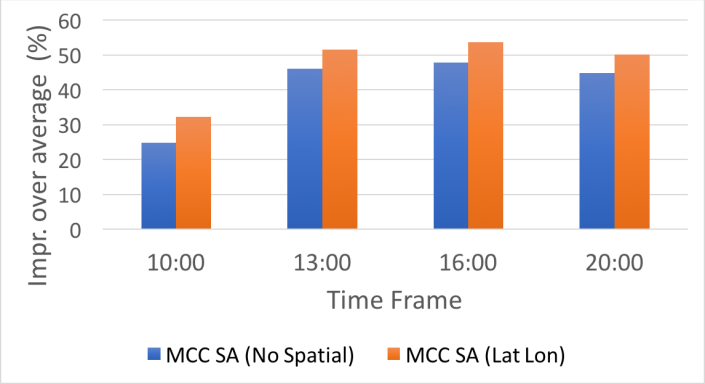

Fig. 15: Percentage of error reduction in different times frames of the day, for the PV Italy dataset. Results are obtained considering the RMSE of the best performing training criterion (MCC_{SA}) and the RMSE of a baseline average predictor.

variants, that consider spatial autocorrelation, to the best of our knowledge, are introduced for the first time in the present study.

The empirical evaluation was performed on two photovoltaic datasets which differ from each other in their size (number of examples), the number of plants, the characteristics of the geographical distribution of the plants, etc. Results show that different training criteria are shown to perform better on different datasets. However, in general, they also show that using entropy-based criteria that consider spatial autocorrelation leads to improvement over those that

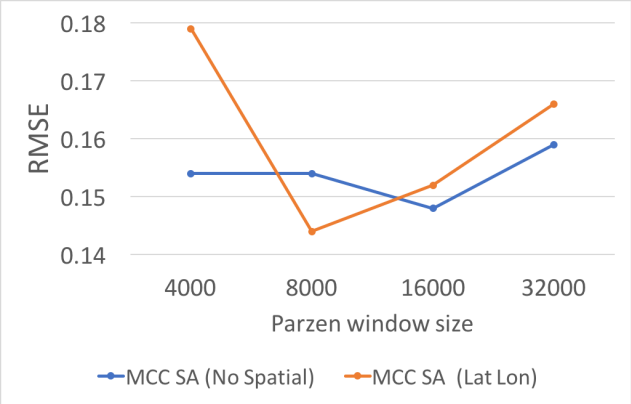

Fig. 16: Average forecasting error (RMSE) with increasing values of the Parzen window size for the PV Italy dataset. Results are obtained with the best performing training criterion (MCC_{SA}).

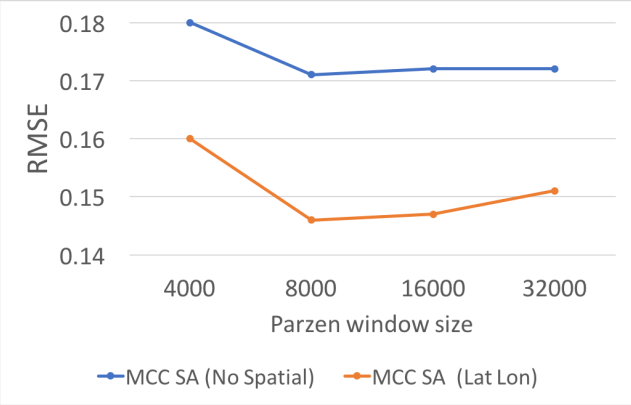

Fig. 17: Average forecasting error (RMSE) with increasing values of the Parzen window size, for the PV NREL dataset. Results are obtained with the best performing training criterion (MCC_{SA}). (MCC_{SA}).

do not consider spatial autocorrelation. The actual improvement will always depend on the spatial and other characteristics of the plant.

References

Aggarwal CC (2013) An introduction to sensor data analytics. In: Aggarwal CC (ed) Managing and Mining Sensor Data, Springer-Verlag, New York, NY, pp 1–8

Appice A, Ciampi A, Fumarola F, Malerba D (2014) Data Mining Techniques in Sensor Networks. SpringerBriefs in Computer Science, Springer-Verlag, London

European Photovoltaic Industry Association E (2014) Global Market Outlook for Photovoltaics 2014-2018

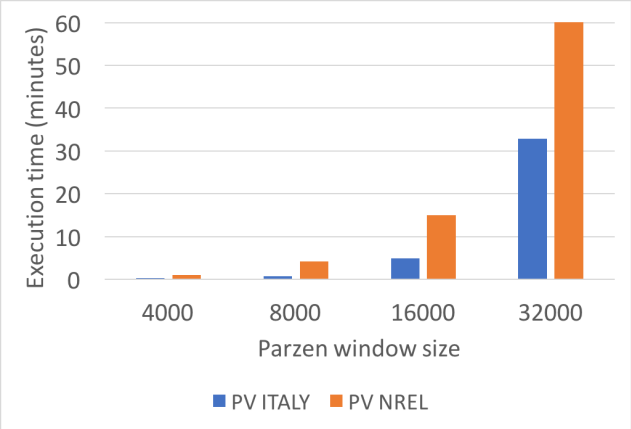

Fig. 18: Execution time with increasing values of the Parzen window size for all datasets. Results are obtained with the best performing training criterion (MCC_{SA}).

Bacher P, Madsen H, Nielsen HA (2009) Online short-term solar power forecasting. Solar Energy 83(10):1772–1783

Barbounis T, Theocharis JB (2007) Locally recurrent neural networks for wind speed prediction using spatial correlation. Information Sciences 177(24):5775–5797

Barlow R, Brunk H (1972) The isotonic regression problem and its dual. Journal of the American Statistical Association 67(337):140–147

Bessa R, Miranda V, Gama J (2008) Wind power forecasting with entropy-based criteria algorithms. In: Proceedings of the 10th International Conference on Probabilistic Methods Applied to Power Systems, IEEE, PMAPS '08, pp 1–7

Bessa RJ, , Miranda V, Gama J (2009) Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting. Power Systems, IEEE Transactions on 24(4):1657–1666

Bessa RJ, Trindade A, Miranda V (2015) Spatial-temporal solar power forecasting for smart grids. IEEE Transactions on Industrial Informatics 11(1):232–241

Bishop CM (1995) Neural Networks for Pattern Recognition. Oxford Univ. Press, Oxford, London

Bludszuweit H, Dominguez-Navarro JA, Llombart A (2008) Statistical analysis of wind power forecast error. Power Systems, IEEE Transactions on 23(3):983–991

Bofinger S, Heilscher G (2006) Solar electricity forecast - approaches and first results. In: 20th Europ. PV conf.

Bogorny V, Valiati J, Camargo S, Engel P, Kuijpers B, Alvares LO (2006) Mining maximal generalized frequent geographic patterns with knowledge constraints. In: Sixth International Conference on Data Mining (ICDM'06), IEEE, pp 813–817

Borcard D, Legendre P, Avois-Jacquet C, Tuomisto H (2004) Dissecting the spatial structure of ecological data at multiple scales. Ecology 85(7):1826–1832

Box GE, Jenkins GM, Reinsel GC, Ljung GM (2015) Time series analysis: forecasting and control. John Wiley & Sons

Buhan S, Cadirci I (2015) Multistage wind-electric power forecast by using a combination of advanced statistical methods. IEEE Trans Industrial Informatics 11(5):1231–1242

Cavalcante L, Bessa RJ, Reis M, Browell J (2017) Lasso vector autoregression structures for very short-term wind power forecasting. Wind Energy 20(4):657–675

Ceci M, Appice A (2006) Spatial associative classification: propositional vs structural approach. Journal of Intelligent Information Systems 27(3):191–213

Ceci M, Corizzo R, Fumarola F, Malerba D, Rashkovska A (2017) Predictive modeling of PV energy production: How to set up the learning task for a better prediction? IEEE Trans Industrial Informatics 13(3):956–966, DOI 10.1109/TII.2016.2604758

Ceci M, Corizzo R, Malerba D, Rashkovska A (2018) Spatial Autocorrelation and Entropy for Renewable Energy Forecasting. DOI 10.5281/zenodo.1242854, URL `https://doi.org/10.5281/zenodo.1242854`

Chakraborty P, Marwah M, Arlitt MF, Ramakrishnan N (2012) Fine-grained photovoltaic output prediction using a bayesian ensemble. In: AAAI

Chu Y, Pedro H, Coimbra C (2013) Hybrid intra-hour dni forecasts with sky image processing enhanced by stochastic learning. Solar Energy 98(PC):592–603, DOI 10.1016/j.solener.2013.10.020

Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Dowell J, Pinson P (2016) Very-short-term probabilistic wind power forecasts by sparse vector autoregression. IEEE Transactions on Smart Grid 7(2):763–770

Erdogmus D, Principe JC (2002) Generalized information potential criterion for adaptive system training. Neural Networks, IEEE Transactions on 13(5):1035–1044

Erdogmus D, Principe JC, Kim SP, Sanchez JC (2002) A recursive renyi's entropy estimator. In: Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on, IEEE, pp 209–217

Fabbri A, Gomezsanroman T, Rivierabbad J, Mendezquezada VH (2005) Assessment of the cost associated with wind generation prediction errors in a liberalized electricity market. Power Systems, IEEE Transactions on 20(3):1440–1446

Fotheringham AS, Brunsdon C, Charlton M (2003) Geographically weighted regression: the analysis of spatially varying relationships. John Wiley & Sons

Gaber MM, Zaslavsky A, Krishnaswamy S (2005) Mining data streams: A review. SIGMOD Rec 34(2):18–26

Gneiting T, Larson K, Westrick K, Genton MG, Aldrich E (2006) Calibrated probabilistic forecasting at the stateline wind energy center: The regime-switching space–time method. Journal of the American Statistical Association 101(475):968–979

He M, Yang L, Zhang J, Vittal V (2014) A spatio-temporal analysis approach for short-term forecast of wind farm generation. IEEE Transactions on Power Systems 29(4):1611–1622

Heaton J (2015) Encog: Library of interchangeable machine learning models for java and c#. J Mach Learn Res 16(1):1243–1247

Hong T, Pinson P, Fan S, Zareipour H, Troccoli A, Hyndman R (2016) Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. International Journal of Forecasting 32(3):896–913, DOI 10.1016/j.ijforecast.2016.02.001

Hyndman RJ, Khandakar Y, et al (2007) Automatic time series for forecasting: the forecast package for r. Tech. rep., Monash University, Department of Econometrics and Business Statistics

Inman R, Pedro H, Coimbra C (2013) Solar forecasting methods for renewable energy integration. Progress in Energy and Combustion Science 39(6):535–576, DOI 10.1016/j.pecs.2013.06.002

Jayasumana AP (2009) Sensor networks - technologies, protocols and algorithms. In: Industrial Electronics, IEEE International Symposium on, IEEE, ISIE 2009

Jebaraj S, Iniyan S (2006) A review of energy models. Renewable and Sustainable Energy Reviews 10(4):281–311, DOI 10.1016/j.rser.2004.09.004

Kalogirou S (2000) Artificial neural networks in renewable energy systems applications: A review. Renewable and Sustainable Energy Reviews 5(4):373–401, DOI 10.1016/S1364-0321(01)00006-5

Kleissl J (2013) Solar Energy Forecasting and Resource Assessment. DOI 10.1016/C2011-0-07022-9

Lange M (2005) On the uncertainty of wind power predictionsanalysis of the forecast accuracy and statistical distribution of errors. Journal of Solar Energy Engineering 127(2):177–194

Lauret P, Voyant C, Soubdhan T, David M, Poggi P (2015) A benchmarking of machine learning techniques for solar radiation forecasting in an insular context. Solar Energy 112:446–457, DOI 10.1016/j.solener.2014.12.014

Li X, Claramunt C (2006) A spatial entropy-based decision tree for classification of geographical information. Transactions in GIS 10(3):451–467

Liu W, Pokharel PP, Príncipe JC (2007) Correntropy: Properties and applications in non-gaussian signal processing. IEEE Transactions on Signal Processing 55(11):5286–5298

Lorenz E, Hurka J, Karampela G, Heinemann D, Beyer HG, Schneider M (2008) Qualified forecast of ensemble power production by spatially dispersed grid-connected pv systems. In: Proceedings of the 23rd European Photovoltaic Solar Energy Conference and Exhibition, pp 3285–3291

Malerba D, Ceci M, Appice A (2005) Mining model trees from spatial data. In: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, pp 169–180

Marquez R, Coimbra C (2013) Intra-hour dni forecasting based on cloud tracking image analysis. Solar Energy 91:327–336, DOI 10.1016/j.solener.2012.09.018

Mathiesen P, Kleissl J (2011) Evaluation of numerical weather prediction for intra-day solar forecasting in the continental united states. Solar Energy 85(5):967–977, DOI 10.1016/j.solener.2011.02.013

Morejon RA, Principe JC (2004) Advanced search algorithms for information-theoretic learning with kernel-based estimators. Neural Networks, IEEE Transactions on 15(4):874–884

Nanni M, Kuijpers B, Korner C, May M, Pedreschi D (2008) Spatiotemporal data mining. In: Giannotti F, Pedreschi D (eds) Mobility, Data Mining and Privacy: Geographic Knowledge Discovery, Springer-Verlag, pp 267–296

Parzen E (1962) On estimation of a probability density function and mode. The annals of mathematical statistics 33(3):1065–1076

Pedro H, Coimbra C (2012) Assessment of forecasting techniques for solar power production with no exogenous inputs. Solar Energy 86(7):2017–2028, DOI 10.1016/j.solener.2012.04.004

Pelland S, Galanis G, Kallos G (2013) Solar and photovoltaic forecasting through post-processing of the global environmental multiscale numerical weather prediction model. Prog Photovolt Res Appl 21(3):284–296

Principe JC (2010) Information theoretic learning: Renyi's entropy and kernel perspectives. Springer Science & Business Media, chap 5, pp 181–218

Principe JC, Xu D (1999a) Information-theoretic learning using renyi's quadratic entropy. In: Cardoso JF, Jutten C, Loubaton P (eds) Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation, Aussois, pp 407–412

Principe JC, Xu D (1999b) An introduction to information theoretic learning. In: Neural Networks, International Joint Conference on, IEEE, IJCNN '99, vol 3, pp 1783–1787

Rashkovska A, Novljan J, Smolnikar M, Mohorčič M, Fortuna C (2015) Online short-term forecasting of photovoltaic energy production. In: Innovative Smart Grid Technologies Conference (ISGT), 2015 IEEE Power & Energy Society, IEEE, ISGT 2015

Rényi A (1976) Selected papers of alfred renyi, vol. 2akademia kiado

Rinzivillo S, Turini F (2007) Knowledge discovery from spatial transactions. Journal of Intelligent Information Systems 28(1):1–22

Sharma N, Sharma P, Irwin DE, Shenoy PJ (2011) Predicting solar generation from weather forecasts using machine learning. In: SmartGridComm, IEEE, pp 528–533

Sheela KG, Deepa S (2013) Review on methods to fix number of hidden neurons in neural networks. Mathematical Problems in Engineering 2013

Stojanova D, Ceci M, Appice A, Dzeroski S (2012) Network regression with predictive clustering trees. Data Min Knowl Discov 25(2):378–413

Tastu J, Pinson P, Trombe PJ, Madsen H (2014) Probabilistic forecasts of wind power generation accounting for geographically dispersed information. IEEE Transactions on Smart Grid 5(1):480–489

Thompson SK (1990) Adaptive cluster sampling. Journal of the American Statistical Association 85(412):1050–1059

Usaola J, Ravelo O, Gonzlez G, Soto F, Dvila MC, Daz-Guerra B (2004) Benefits for wind energy in electricity markets from using short term wind power prediction tools; a simulation study. Wind Engineering 28(1):119–127

Yang H, Kurtz B, Nguyen D, Urquhart B, Chow C, Ghonima M, Kleissl J (2014) Solar irradiance forecasting using a ground-based sky imager developed at uc san diego. Solar Energy 103:502–524, DOI 10.1016/j.solener.2014.02.044

Zhang J, Florita A, Hodge BM, Lu S, Hamann H, Banunarayanan V, Brockway A (2015a) A suite of metrics for assessing the performance of solar power forecasting. Solar Energy 111:157–175, DOI 10.1016/j.solener.2014.10.016

Zhang J, Hodge BM, Lu S, Hamann H, Lehman B, Simmons J, Campos E, Banunarayanan V, Black J, Tedesco J (2015b) Baseline and target values for regional and point pv power forecasts: Toward improved solar forecasting. Solar Energy 122:804–819, DOI 10.1016/j.solener.2015.09.047

Zhao M, Li X (2011) An application of spatial decision tree for classification of air pollution index. In: Geoinformatics, 2011 19th International Conference on, IEEE, pp 1–6

Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67(2):301–320