Very short-term solar forecasting using multi-agent system based on extreme learning machines and data clustering

Carlos Severiano
Graduate Program in Electrical Engineering,
Federal University of Minas Gerais,
Av. Antônio Carlos 6627,
31270-901 Belo Horizonte, MG, Brazil
Email: carlossjr@gmail.com

Frederico Gadelha Guimarães
Department of Electrical Engineering,
Federal University of Minas Gerais,
UFMG, Belo Horizonte, Brazil
Email: fredericoguimaraes@ufmg.br

Miri Weiss Cohen
Department of Software Engineering,
ORT Braude College of Engineering,
Karmiel, Israel
Email: miri@braude.ac.il

***Abstract*—This paper proposes a new multi-agent system to solve very short-term solar forecasting problems. The system organizes the training data into clusters using Part and Select Algorithm. These clusters are used to generate different forecasting models, where each one is performed by a different agent. Finally, another agent is responsible for deciding which model will be applied at each forecasting situation. Results present improvements in forecasting accuracy and training performance if compared to other forecasting methods. A discussion of how to use this architecture for the implementation of a more comprehensive model is also addressed.**

I. Introduction

During the past two decades, renewable energy such as solar photovoltaics (PV) or wind has become relevant as a source of electricity generation in power grids worldwide. In the last fifteen years, PV energy reached a compound annual growth rate of circa 40% [1]. The installed capacity in OECD countries has already reached 60 GW. In some European countries, PV production reaches 30% of overall power production during clear summer days. This scenario encourages the incorporation of PV energy into electrical power grids that operate with other energy sources, such as fossil.

One of the challenges related to this incorporation and operation of solar PV is dealing with the variable aspect of this source of energy, which can alternate during the day due to weather conditions, such as cloud motion, and also has no production during night time. These characteristics highlight the importance of applying a forecasting method to a PV system [2]. It is useful to distinguish forecasting techniques according to the temporal scales we are interested in. Very short-term forecasting deals with time horizons from minutes to hours, while short-term forecasting deals with time horizons from hours to a few days ahead. This work focuses on very short-term forecasting (for time horizons such as 15-min and 30-min), investigating data collected from a station of a PV system located in a tropical region, specifically in Singapore.

In general, electrical load forecasting models suffer from similar needs of techniques with good capability of dealing with complex relations between input and output, which encourages the adoption of Artificial Intelligence (AI) models [3]. Specifically in solar forecasting, Mellit and Pavan [4] and Pedro and Coimbra [5] applied Artificial Neural Networks (ANN) to solve this problem.

As discussed in [3], seasonality is a recurrent problem in load forecasting models. Different seasons present different patterns of weather variables, which directly affect the solar irradiance curves along the days. Some works deal with seasonality by designing multi-agent systems responsible for organizing data and generating distinct and more accurate forecasting models. For instance, Yap and Yap [6] and Hernandez et al. [7] consider load or demand profiles associated with specific days of the year to divide the forecasting problem into smaller problems.

Another factor that can influence the solar forecasting is the spatial location of grid stations. Even in limited areas, distributed stations can be affected by different weather conditions. This implies that a good forecasting system for a distributed electricity grid should be based on the development of customized models for the different conditions to which its components are subject. The models should be designed to solve local problems, but must behave as a unit to provide useful information.

Solar forecasting provides extremely useful information for tasks such as management of electricity grids and solar energy trading. In a very short term forecasting context, it means that fast turnaround of results is also an important feature to the model, where the time for decision making is reduced and the amount of data to be processed is usually higher. Hence, in addition to good accuracy, good performance is a desirable goal for a very short term forecasting model.

Coordination and control of these new emerging grid components remains a great challenge [8]. Advanced networking, as well as Information and Communication Technologies, have been motivating the integration of the conventional power grid in smarter ways [9], inspiring the use of distributed Multi-Agent Systems (MAS). Autonomous control of these systems allows placing additional stations without reengineering the whole system, and using it in the peer-to-peer model eliminates the requirement of a complex central controller and associated telecommunication facilities [10]. Logenthiran et al. [11] underscored that MAS is one of the fastest growing domains in agent oriented technology which deals with autonomous decision modeling. Moreover, it has been showing to be crucial in Smart Grid (SG) operations [12]. MAS has spread to diverse SG applications in the field of power systems restoration, security and protection, control, monitoring, energy storage and maintenance scheduling, and electric power market simulation [13].

In this paper, the proposed model combines Extreme Learning Machines (ELM) [14, 15] with a clustering technique named Part and Select Algorithm (PSA) [16] to compose a MAS responsible for performing the solar forecasting of a distributed energy grid. The main motivation for this model is to deal with the seasonality in an indirect way, by grouping days with similar irradiance patterns so that each group can generate a different specialized ELM model. Additionally, we aim at elaborating a multi-agent model with better performance for very short term forecasting. The technique is compared to some popular forecasting techniques as well as to a simple ELM model in order to asses its accuracy and performance.

This article is organized as follows. Section II presents some classical techniques applied to forecasting, which are evaluated in this work. Section III describes the data used for forecasting as well as the preprocessing steps taken. Section IV describes the proposed multi-agent model. Section V shows the experiments and results. Section VI concludes the paper discussing the results and suggesting some future work.

II. Classical Forecasting Methods

The forecasting problem discussed in this work has been widely studied by several approaches. This section presents the methods chosen for a comparison with the new proposed model.

A. Persistence

The Clear Sky model represents the values for solar irradiance if sky conditions would be neglected. Under this model, solar irradiance forecasting could be calculated with good accuracy using a deterministic model. However, in actual situations the sky conditions also provide a stochastic component that strongly affects the forecasting result.

Persistence is a very common forecasting method. Usually serves as a benchmark against other proposed methods. Persistence assumes that the forecast value has equal conditions to the current one. In solar irradiance forecasting, equal conditions are related to cloud cover conditions, which are assumed to stay the same for the next time step. The equation applied for persistence uses solar irradiance observations and values from the Clear Sky model, described as follows:

$$\hat{I}_{t+1} = \begin{cases} I_{cs,t+1} & \text{if } I_{cs,t} = 0 \\ \frac{I_{cs,t+1}}{I_{cs,t}} \times I_t & \text{otherwise} \end{cases} \tag{1}$$

Whereas persistence is considered a simple method, good results can be obtained from its application, specially in regions where weather patterns change very little or the features on the weather maps move very slowly. As could be observed in the experiments of this work, this method can have a performance comparable to more advanced techniques in some specific conditions, which implies that it can be considered more than a naive forecasting alternative.

B. Autoregressive integrated moving average

Autoregressive Integrated Moving Average (ARIMA) models are popular time series models in forecasting tools for non-stationary time series [17]. The model combines an autoregressive component to a moving average component, elaborated from a generalization of the ARMA model. Its representation $ARIMA(p, d, q)$ refers, respectively, to autoregression, integration and moving average degrees, i.e., p is the autoregressive order, d is the differencing degree, q is the moving average model order.

The ARIMA model for the time series x_t can be written as:

$$y_t = (1 - L)^d x_t \tag{2}$$

$$\phi(L)y_t = \theta(L)\varepsilon_t \tag{3}$$

where y_t is the predicted value at time t, x_t are the time series values, and:

$$\phi(L) = \left(1 - \sum_{i=1}^{p} \phi_i L^i\right) \tag{4}$$

$$\theta(L) = \left(1 - \sum_{j=1}^{q} \theta_j L^j\right) \tag{5}$$

The parameters ϕ_i and θ_j are associated with the autoregressive part and the moving average part, respectively, while ε_t represent the error terms. In addition, L^i is the backshift or lag operator, i.e., given x_t, we have $L^k x_t = x_{t-k}$.

Evaluating the correlation between the values of the time series can improve the elaboration of an ARIMA model with proper parameters. The autocorrelation plot for 30-min lag observations of the evaluated stochastic component depicted in Figure 1 presents positive values for a high number of lags, which suggests an ARIMA model with some degree of differencing. The partial autocorrelation plot presents a cutoff at the first lags, which points to a lower order of autoregressive and moving average components.

The seasonal ARIMA model can be written as $SARIMA(p, d, q)(P, D, Q)_s$, with additional parameters such that P is the seasonal autoregressive terms, D is the number of seasonal differences, Q is the seasonal moving average order. The model is given by:

$$y_t = (1 - L)^d (1 - L^s)^D x_t \tag{6}$$

$$\phi(L)\Phi(L^s)y_t = \theta(L)\Theta(L^s)\varepsilon_t \tag{7}$$

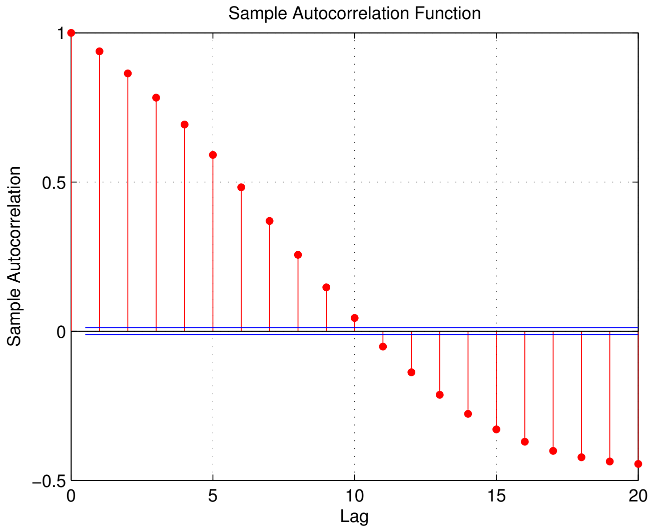


Fig. 1. Autocorrelation plot for the stochastic component.

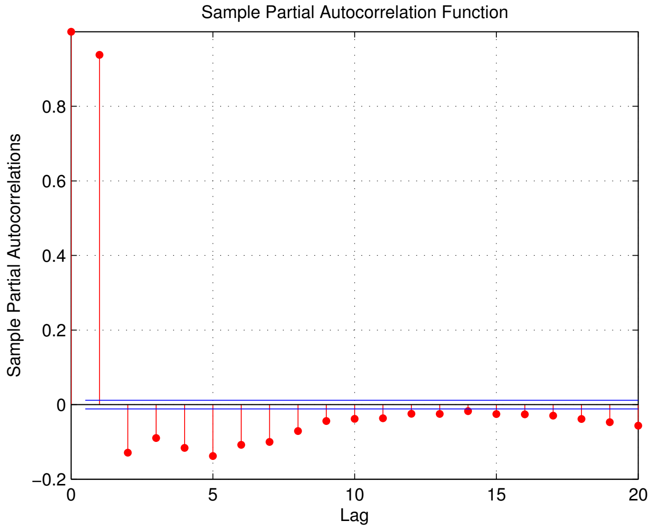


Fig. 2. Partial correlation plot for the stochastic component.

where

$$\Phi(L^s) = \left(1 - \sum_{i=1}^{P} \Phi_i L^{si}\right) \tag{8}$$

$$\Theta(L^s) = \left(1 - \sum_{j=1}^{Q} \Theta_j L^{sj}\right) \tag{9}$$

Based on the 30-min lags used in this work, an entire day can represent a seasonality period for this model. Since there are 48 daily observations, this value was evaluated as the seasonality parameter. However the best results were obtained with no seasonal component and an entire day of lags as input. The chosen parameters for this work were $p = 1$, $d = 1$ and $q = 1$.

C. k-Nearest-Neighbors

k-Nearest-Neighbors (kNN) [18] is a technique widely used in pattern recognition and classification. Basically, the method compares patterns, finding the most similar ones, which are the nearest neighbors. Applied to this forecasting problem, the method evaluates input patterns from the training dataset. For example, consider a pattern from the training dataset with the format $< P_t, O_t >$ where:

$$\begin{aligned} P_t &= (I_{st,t}, I_{st,t-1}, I_{st,t-2}, \ldots, I_{st,t-n}) \\ O_t &= I_{st,t+1} \end{aligned} \tag{10}$$

Given a current input pattern Q_t, the method compares its values to all the P_t patterns in the training dataset using the following equation:

$$Dist(s) = \sqrt{\sum_{i=1}^{n} (P_{s,i} - Q_{t,i})^2} \tag{11}$$

The forecast result is the value O_s, which is the output for the pattern P_s with the lowest value for $Dist$. If more than one pattern have the minimal distance, an average from their output values is taken.

III. Data Processing

A. Source of data

The data used in this work was obtained in a solar energy station located in Singapore. The station is part of a ground-based meteorological network deployed by SERIS - Solar Energy Research Institute of Singapore - along the island as part of a research project [19]. In order to obtain the ground measurement readings at a good spatial resolution, 25 stations of the project were distributed across Singapore using a 5x5 km grid as reference. Figure 3 shows this distribution.

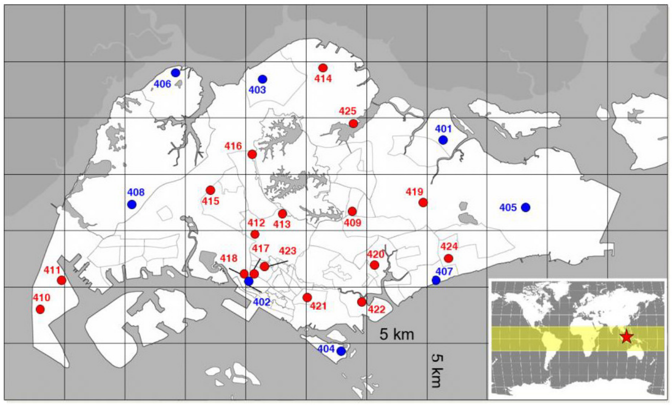


Fig. 3. Singapore map of SERIS ground-based stations.

All the monitoring systems record the solar irradiance values as one minute averages, with a sampling rate of 1 Hz. Similarly, the 30-min values used in this work correspond to averages of one minute groups. It is expected that, due to material specifications, loggers utilized in this work present errors in the order of ±0.2%.

Since the focus of this work is presenting the training and forecasting characteristics of this model only a central station was chosen. It is referenced in the map of Figure 3 as number **402**. The entire multi-agent system, proposed to cover the whole network, is composed by the same set of agents described here, applied to each station.

B. Preprocessing data

As previously discussed, the observed solar irradiance values can be seen as time series with trend and seasonal components. In this case, better results in forecasting can be obtained if these components are eliminated or mitigated [20], specially when working with neural network models. Detrending and seasonal adjustment are common steps in order to improve time series analysis.

In this work, the applied detrending method was smoothing the time series using the moving average filter. It means that, considering a time series y, its smoothed $\tilde{y}$ values are calculated with:

$$\begin{aligned}
\tilde{y}(1) &= y(1) \qquad (12)\\
\tilde{y}(2) &= (y(1) + y(2) + y(3))/3\\
\tilde{y}(3) &= (y(1) + y(2) + y(3) + y(4) + y(5))/5\\
\tilde{y}(4) &= (y(2) + y(3) + y(4) + y(5) + y(6))/5\\
&\vdots\\
\tilde{y}(n) &= (y(n-2) + y(n-1) + y(n) + y(n+1) + y(n+2))/5
\end{aligned}$$

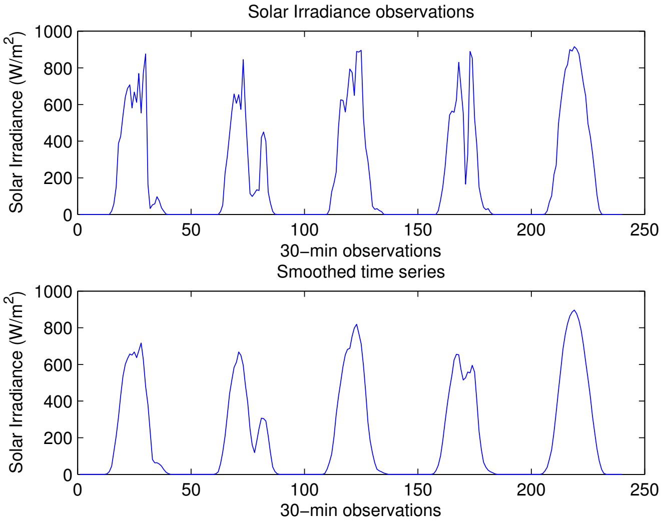


Fig. 4. Solar irradiance observations from April 01, 2013 to April 05, 2013 and their smoothed values.

Figure 4 shows an example of smoothed values obtained from the application of a moving average filter with a span of 5 to a period of observations.

Thus, the detrending process consists of removing the smoothed values I_t from the original time series I, leaving only the residual detrended data Idt.

$$I = I_{dt} + I_t \qquad (13)$$

Since preliminary results pointed that the detrending process improved the average forecasting accuracy, all the evaluated methods in this work operate with the detrended data I_{dt}.

Seasonal adjustment is a more complicated task for this kind of time series, since sky conditions are affected by different factors along the year, which provide different daily solar irradiance curves. This leads to a scenario where inferring a proper additive or multiplicative model for seasonality can be a complex task. Seasonal adjustment is not performed during the preprocessing step in a traditional way, but left for each forecasting method, if applied.

IV. Multi-agent forecasting system

The proposed Multi-agent system is basically composed by agents responsible for different steps of the forecasting process. The forecasting itself is performed by a combination of Extreme Learning Machines and the Part and Select Algorithm. The following subsections describe the agents and also the methods that were combined in order to create the forecasting methods.

A. Extreme Learning Machines

Extreme learning machines (ELM) [14] have an architecture similar to a single hidden layer feedforward network, but the distinct feature lies in its learning algorithm. The algorithm is based on empirical risk minimization theory and requires only a single iteration. Considering n samples $(x_i, y_i) \in X \times Y$, $i = 1, \ldots, N$, where $x_i = [x_{i1}, x_{i2}, ..., x_{in}]$ with $X \subset \mathbb{R}^n$ and $y_i = [y_{i1}, y_{i2}, ..., y_{im}]$ with $Y \subset \mathbb{R}^m$. The training process basically consists of two steps:

1) Find an intermediate mapping

$$H = f(XZ) \qquad (14)$$

The intermediate mapping H is a result of an activation function f applied to a projection made with the input X and a set of weights Z, where a weight $z_{i,j}$ connects an input node i to a node j that belongs to the hidden layer. The values of Z are chosen randomly. The objective is to convert the original input to a linearly solvable problem.

2) Find a solution for the linear problem

$$\hat{Y} = HW \qquad (15)$$

where $\hat{Y}$ is the labelled output and W is a weight matrix that connects the hidden layer to the output layer. To solve this problem, the values of W are obtained using the training data Y and the Moore-Penrose pseudoinverse of the mapping $H\dagger$ as follows:

$$W = YH\dagger \qquad (16)$$

Since the W matrix is obtained, the ELM model is trained and can be applied to test samples following similar steps. Given an input data set X_{test}:

$$H_{test} = h(X_{test}Z) \tag{17}$$
$$Y_{test} = H_{test}W \tag{18}$$

In this work, ELM only takes as input an excerpt from a time series of solar irradiance concerning to recent past observations. For example, given a current time t and a set of n past observations, the model takes as input the solar irradiance observations from $t-n$ to t in order to forecast the value y for time $t+1$.

B. Part and Select Algorithm

Part and Select Algorithm [16] selects m well-spread points from a set of n candidate solutions. Basically, the algorithm has two steps:

1) Similar members are grouped in the same subset.
2) A diverse subset is formed by selecting one member from each generated subset.

Aiming to divide a dataset into m subsets, PSA performs $m-1$ divisions of one single set into two subsets. At each step, the set with the greatest dissimilarity among its members is the one that is divided. This is repeated until the desired stopping criterion is met. In this work, a predefined number of subset is applied as stopping criterion. The dissimilarity of a set A, denoted by $\varnothing A$, is calculated as follows:

Let $A = x_1 = [x_{11}, \ldots, x_{1k}], \ldots, x_n = [x_{n1}, \ldots, x_{nk}] \in R^k$, and consider

$$\begin{aligned} a_j &= \min_{i=1,\ldots,n} x_{ij} \\ b_j &= \max_{i=1,\ldots,n} x_{ij} \\ \Delta_j &= b_j - a_j, j = 1, \ldots, k \\ \varnothing A &= \max_{j=1,\ldots,k} \Delta_j \end{aligned} \tag{19}$$

As discussed in [16], $\varnothing A$ can also be seen as the diameter of the set A in the Chebyshev metric.

Once the dataset is divided into m subsets, another subset A' of size m is obtained. A' contains the most representative element of each subset. The criterion used in PSA is choosing the closest element to the hyper-rectangle circumscribing A_i. The distance is calculated using Euclidean metric.

The subset containing the most representative elements can provide a very useful information for this work. It facilitates the visualization and identification of different patterns of time series, which can assist in making decisions about the more appropriate method for each situation. Therefore, it is one of the motivations for using PSA as clustering method.

C. Multi-agent Architecture

The proposed multi-agent forecasting system involves delegating the various forecasting tasks to different modules. The modules are described in detail as follows. Figure 5 depicts the model.

Data Preprocessing Agent: The Data Preprocessing Agent is responsible for receiving and processing the data collected from the stations. The following tasks are performed by this agent:

1) *Detrending*: The agent receives as input the raw training data extracted from the monitoring system. In the first step, the detrending process discussed in section IV is executed to the training data.
2) *Normalization*: The stochastic component extracted from the detrending process is normalized as suggested in [14]. The normalization range for this work was -1 to $+1$.

Training Agent: The training agent is responsible for all the learning process of the forecasting model. The tasks below are performed:

1) *Data clustering*: The agent uses the normalized stochastic component generated by the Data Preprocessing agent. The data clustering is performed using the PSA method. According to the PSA method, each cluster has its most representative element, which is elected in this step. Each element contains all the observations of an entire day. The model works with a fixed number of clusters. In this work, this number was defined empirically using as a criterion the balance between forecasting accuracy and training performance.
2) *ELM Training*: Each cluster created by PSA serves as input to train an ELM model. During the experiments, the ELMs were trained sequentially.

Forecasting Agent: Forecasting agent is responsible for returning the predicted value using the trained ELM models. Two steps are taken to perform the forecasting:

1) *ELM model selection*: In order to decide the most suitable ELM model for forecasting, among the ones created by the Training Agent, the forecasting compares the distance between the input data and the most representative elements of all the data clusters. This comparison is done by calculating the Euclidean distance between two vectors, which are the time series used as input data and the same interval in the representative element. For example, given an input data composed by observations from 8am and 11am, divided in 30 minute intervals, generating a vector with 7 elements. Since each representative element contains an entire day of observations, the same interval from 8am and 11am is extracted from the element generating a vector with same to finally apply the Euclidean distance. The ELM whose representative element returns the lower distance is chosen.
2) *Forecasting*: Chosen the most suitable ELM model, the forecasting is performed using the input data. In this work the returned value corresponds to the solar irradiance one step ahead $(t+1)$.

V. Experiments

The experiments were conducted to evaluate the performance of all the described forecasting methods in terms of

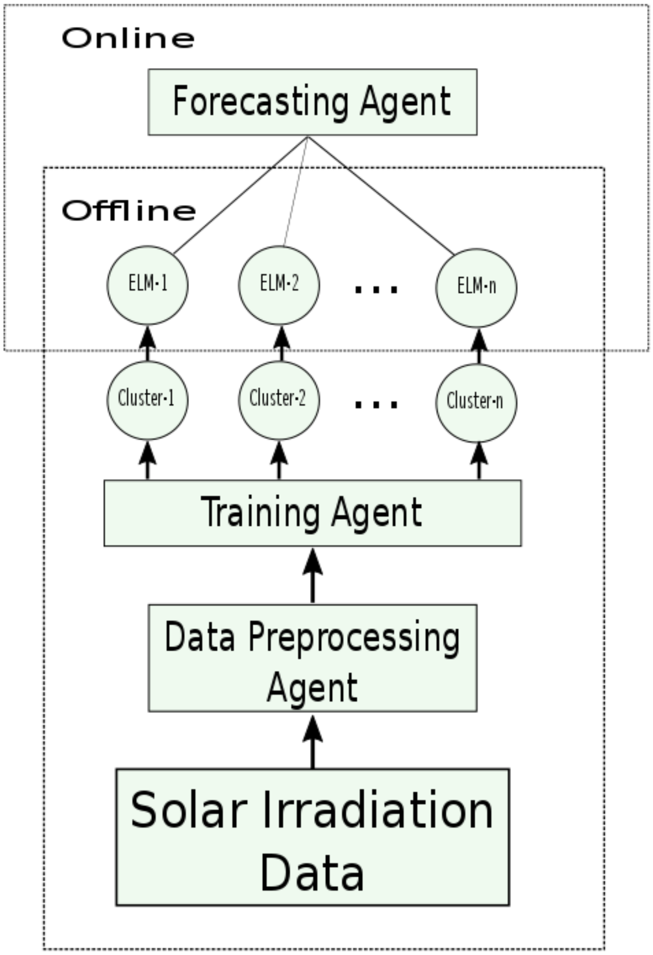


Fig. 5. *Multi-agent ELM model.*

processing and accuracy. As mentioned earlier in section III, the input data used was the detrended component I_{dt} of solar irradiance observations. However, the accuracy evaluation was performed using the actual data, obtained by the equation $I = I_{dt} + I_t$.

The dataset used for training and testing was irradiance observations registered on intervals of 30 minutes. As detailed below, 12 months from 2013 to 2014 were applied for training and the subsequent 12 months from 2014 to 2015 were used for testing, which means that a dataset composed by 2 years of observations was split in half.

1) **Training data:**

   from April, 2013 to March, 2014

2) **Testing data:**

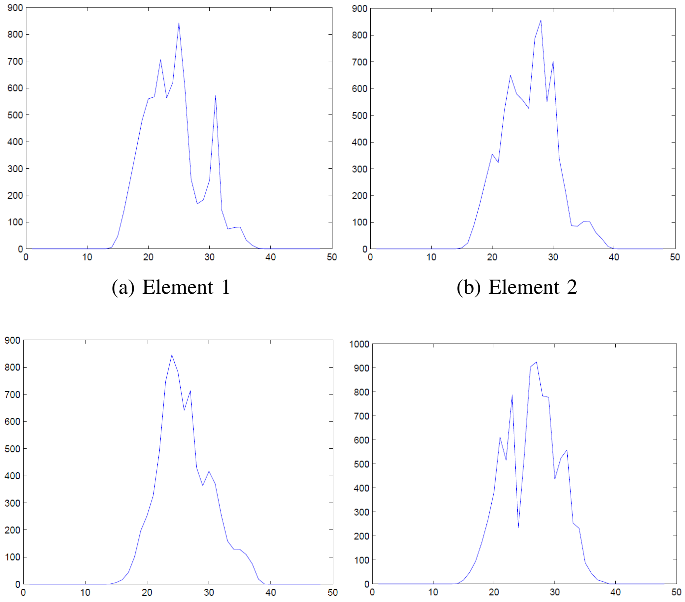
   from April, 2014 to March, 2015

The purpose of taking an entire year for testing was to evaluate forecasting performance over different profiles of solar irradiance.

The input data for the methods was a time series representing 6 sequential observations of solar irradiance. In other words, given a current observation t, the input data is collected from $t-5$ to t in order to predict the future value $t+1$. This input format is used in all the evaluated ELM models in this work. Also, the number of nodes in the hidden layer is the same for all the models. Each model is organized with 12 nodes.

Both the simple ELM and Multi-agent ELM (MA-ELM) models have random components in their foundations. Since this characteristic can affect the final results, each model was run 30 times and the results reported correspond to an average of all the independent runs. For these models, their standard deviations are also presented.

For the multi-agent model, the training dataset was divided into 4 subsets, as a result of the PSA method. Each element of the dataset contains the irradiance observed during one day, which means for a dataset with 30-min intervals 48 observations per element. Figure 6 shows the most representative elements of each subset.



(a) Element 1

(b) Element 2

(c) Element 3

(d) Element 4

Fig. 6. Example of representative elements for each PSA cluster

The statistical metric Root Mean Square Error (RMSE) and its normalized version (nRMSE) were applied to evaluate the accuracy of the methods:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(I_t - \hat{I}_t)^2} \qquad (20)$$

$$nRMSE = \sqrt{\frac{\sum_{t=1}^{n}(I_t - \hat{I}_t)^2}{\sum_{t=1}^{n}\hat{I}_t^2}} \qquad (21)$$

Tables I and II show the results of forecasting in terms of accuracy for the ARIMA model with seasonal component (ARIMA), kNN, ELM and MA-ELM.

During the experiments, training and test times in seconds were also observed. Tables III and IV present the results.

TABLE I
Forecasting error - RMSE.

RMSE					
	Persistence	**ARIMA**	**kNN**	**ELM**	**MA-ELM**
2014-04	84.7667	92.2204	80.6711	79.7529 ± 0.2564	81.8279 ± 0.5122
2014-05	80.1549	93.6016	79.1877	76.4931 ± 0.6512	77.2989 ± 0.4338
2014-06	66.0692	70.4749	64.2329	60.6537 ± 0.2723	61.7802 ± 0.2681
2014-07	74.7301	78.2315	72.4574	69.3097 ± 0.2850	70.9656 ± 0.3942
2014-08	81.9827	83.0472	78.5713	75.6961 ± 0.2094	76.1392 ± 0.3089
2014-09	64.5804	81.2112	67.1694	62.4151 ± 0.3093	63.1119 ± 0.4765
2014-10	73.1025	85.8366	70.2621	68.8848 ± 0.3145	72.0660 ± 1.0020
2014-11	88.9649	86.4535	78.1195	76.1150 ± 0.2844	79.1113 ± 0.7843
2014-12	78.2640	88.9237	76.4641	74.9684 ± 0.3761	79.3020 ± 0.6221
2015-01	80.3508	78.6993	76.0221	76.2232 ± 0.3073	77.0896 ± 0.3459
2015-02	71.8613	74.4020	69.3758	68.3714 ± 0.2315	69.8711 ± 0.3088
2015-03	82.9459	83.0699	80.4487	78.1325 ± 0.3205	78.5508 ± 0.3404
Average	**77.3144**	**83.0143**	**74.4152**	**72.2513 ± 0.3182**	**73.9262 ± 0.4831**

TABLE II
Forecasting error - nRMSE.

nRMSE (%)					
	Persistence	**ARIMA**	**kNN**	**ELM**	**MA-ELM**
2014-04	26.0970	28.3981	24.8361	24.5534 ± 0.0789	25.1922 ± 0.1577
2014-05	25.2920	29.5008	24.9868	24.1365 ± 0.2055	24.3908 ± 0.1369
2014-06	20.4130	21.9464	19.8457	18.7398 ± 0.0841	19.0879 ± 0.0828
2014-07	24.2044	25.6209	23.4683	22.4488 ± 0.0923	22.9851 ± 0.1277
2014-08	28.0606	28.5605	26.8930	25.9088 ± 0.0717	26.0605 ± 0.1057
2014-09	17.9420	22.8274	18.6613	17.3404 ± 0.0859	17.5340 ± 0.1324
2014-10	22.2266	26.0694	21.3630	20.9442 ± 0.0956	21.9114 ± 0.3046
2014-11	31.2467	30.0230	27.4375	26.7335 ± 0.0999	27.7859 ± 0.2755
2014-12	28.2985	31.9184	27.6477	27.1069 ± 0.1360	28.6738 ± 0.2250
2015-01	23.9825	23.2046	22.6905	22.7506 ± 0.0917	23.0091 ± 0.1032
2015-02	20.2438	21.1003	19.5436	19.2607 ± 0.0652	19.6831 ± 0.0870
2015-03	24.5532	24.7428	23.8140	23.1283 ± 0.0949	23.2522 ± 0.1008
Average	**24.3801**	**26.1594**	**23.4323**	**22.7543 ± 0.1001**	**23.2972 ± 0.1533**

For ARIMA, the training time corresponds to the parameter estimation step. No training step is required for kNN.

TABLE III
Training time in seconds.

Train time			
ARIMA	**kNN**	**ELM**	**MA-ELM**
38.6510	-	8.2736	0.9367

TABLE IV
Average test time in seconds.

Test time			
ARIMA	**kNN**	**ELM**	**MA-ELM**
0.2410	0.1652	0.0113	0.0162

The results denote equivalent performances in accuracy for ELM and MA-ELM, which suggest that they offer good response for seasonal changes if compared to the other methods. An important aspect of the MA-ELM model is that its training time is much lower than the ELM. It means that the new model can split the forecasting problem into a smaller and simpler set of ELM with no substantial loss of accuracy. This can be an important improvement for very short term forecasting problems, where a usually higher amount of data needs to be processed at a reduced time. Other aspect observed in the experiments was that, while the MA-ELM takes an additional step when forecasting is performed (deciding which ELM model will be used), it has no substantial effect in performance. The representative elements of each PSA cluster show very different curves representing daily solar irradiance, which suggest that along the year different groups of load profiles occur, influenced by different weather characteristics. Finding correlations between these profiles and other weather variables can be a useful task to address seasonality problems.

VI. Conclusion and Future Work

The multi-agent architecture provides comparable results to classical forecasting methods, still taking the advantage of being composed by a set o agents that can work autonomously and asynchronously. This aspect encourages improvements in the model in terms of readaptation. Since the training step for the MA-ELM is not an expensive task, an additional agent, responsible for retraining could be incorporated to the model. Some variables could be monitored in order to decide when the agent should trigger the retraining, such as the feedback of forecasting. Other aspect to be carried out aiming at improving the results is the relation between the diversity of the clusters [16] and the training errors. This equation, if valid, could help to automate some parts of the ELM modelling.

As discussed, this paper was focused on evaluating forecasting issues of the multi-agent model applied to one station. Other aspects to be explored are related to the entire system, composed by agents distributed along the network. An important task corresponds to the communication between these agent systems, which could provide useful information to improve the forecasting system as a whole. For example, situations when the model of a station had to be retrained (e.g. due to higher forecasting errors) could be informed to neighboring stations since this event might be caused by weather changes that would affect other stations in the near future. The communication between the multi-agent systems could also respond to a central system, where all the informations are gathered to help the electrical grid management.

Acknowledgment

This work has been supported by the Brazilian agency CAPES.
The authors would like to thank the support given by the Brazilian agency FAPEMIG.
The authors would also like to thank the Solar Energy Institute of Singapore (SERIS) for providing the data used in this work.

References

[1] J. Gifford, “BNEF: global solar to grow to close to 60 gw in 2015,” http://www.pv-magazine.com/news/details/beitrag/bnef--global-solar-to-grow-to-close-to-60-gw-in-2015_100017122, 2014.

[2] A. M. Nobre, “Short-term solar irradiance forecasting and photovoltaic systems performance in a tropical climate in Singapore,” Ph.D. dissertation, Federal University of Santa Catarina, Brazil, 2015.

[3] M. Q. Raza and A. Khosravi, “A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings,” *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 1352–1372, 2015.

[4] A. Mellit and A. M. Pavan, “A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy,” *Solar Energy*, vol. 84, no. 5, pp. 807–821, 2010.

[5] H. T. C. Pedro and C. F. M. Coimbra, “Assessment of forecasting techniques for solar power production with no exogenous inputs,” *Solar Energy*, vol. 86, no. 7, pp. 2017–2028, 2012.

[6] K. S. Yap and H. J. Yap, “Daily maximum load forecasting of consecutive national holidays using OSELM-based multi-agents system with weighted average strategy,” *Neurocomputing*, vol. 81, pp. 108–112, 2012.

[7] L. Hernandez, C. Baladron, J. M. Aguiar, B. Carro, A. J. Sanchez-Esguevillas, J. Lloret, D. Chinarro, J. J. Gomez-Sanz, and D. Cook, “A multi-agent system architecture for smart grid management and forecasting of energy demand in virtual power plants,” *Communications Magazine, IEEE*, vol. 51, no. 1, pp. 106–113, 2013.

[8] R. A., R. S. D., and J. N. R., “Challenges for autonomous agents and multi-agent systems research,” *Twenty-Sixth AAAI Conference on Artifcial Intelligence (AAAI-12)*, pp. 2166–2172, 2012.

[9] N. C. P. and F. A. J., “Agent based restoration with distributed energy storage support in smart grids,” *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 1029–1038, June 2012.

[10] N. Lidula and A. Rajapakse, “Microgrids research: A review of experimental microgrids and test systems,” *Renewable and Sustainable Energy Reviews*, vol. 15, no. 1, pp. 186 – 202, 2011.

[11] T. Logenthiran, D. Srinivasan, and D. Wong, “Multi-agent coordination for der in microgrid,” in *2008 IEEE International Conference on Sustainable Energy Technologies*, Nov 2008, pp. 77–82.

[12] G. H. Merabet, M. Essaaidi, H. Talei, M. R. Abid, N. Khalil, M. Madkour, and D. Benhaddou, “Applications of multi-agent systems in smart grids: A survey,” in *Multimedia Computing and Systems (ICMCS), 2014 International Conference on*, April 2014, pp. 1088–1094.

[13] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci, and T. Funabashi, “Multi-agent systems for power engineering applications x2014;part i: Concepts, approaches, and technical challenges,” *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1743–1752, Nov 2007.

[14] 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, 2006.

[15] S. Ding, H. Zhao, Y. Zhang, X. Xu, and R. Nie, “Extreme learning machine: algorithm, theory and applications,” *Artificial Intelligence Review*, vol. 44, no. 1, pp. 103–115, 2015.

[16] S. Salomon, G. Avigad, A. Goldvard, and O. Schütze, “PSA - a new scalable space partition based selection algorithm for MOEAs.” in *EVOLVE*, ser. Advances in Intelligent Systems and Computing, vol. 175. Springer, 2012, pp. 137–151.

[17] G. Box, G. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting and Control, fourth ed.* Wiley, 2008.

[18] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[19] “Solar energy research institute of singapore,” http://www.seris.sg/.

[20] P. G. Zhang and M. Qi, “Neural network forecasting for seasonal and trend time series,” *European Journal of Operational Research*, vol. 160, pp. 501–514, 2005.