A Hybrid Intelligent System to Forecast Solar Energy Production

Nuño Basurto[a], Ángel Arroyo[a], Rafael Vega[b], Héctor Quintián[b], José Luis Calvo-Rolle[b], Álvaro Herrero[a]

[a] Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Ingeniería Civil, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n, 09006, Burgos, Spain
nbasurto@ubu.es, aarroyop@ubu.es, ahcosio@ubu.es

[b] Department of Industrial Engineering, University of Coruña, Avda. 19 de febrero S/N 15405, Ferrol - Coruña, Spain
rafael.alejandro.vega.vega@udc.es, hector.quintian@udc.es, jlcalvo@udc.es

Abstract. There is wide acknowledgement that solar energy is a promising and renewable source of electricity. However, complementary sources are sometimes required, due to its limited capacity, in order to satisfy user demand. A Hybrid Intelligent System (HIS) is proposed in this paper to optimize the range of possible solar energy and power grid combinations. It is designed to predict the energy generated by any given solar thermal system. To do so, the novel HIS is based on local models that implement both supervised learning (artificial neural networks) and unsupervised learning (clustering). These techniques are combined and applied to a real-world installation located in Spain. Alternative models are compared and validated in this case study with data from a whole year. With an optimum parameter fit, the proposed system managed to calculate the solar energy produced by the panel with an error that was lower than 10^{-4} in 86% of cases.

Keywords: Hybrid Intelligent System, Clustering, Regression, Neural Networks, Solar Energy, Renewable Energies.

1 Introduction and Previous Work

Renewable Energy (RE) has a key role to play in the field of increased sustainability and is one of the most relevant technologies in that respect [1]. As a consequence, many buildings and especially new ones now incorporate RE facilities. This general European trend has also been applicable to Spain over recent years, where the rate of new RE installations, in general, and solar thermal energy, in particular, is increasing. One reason is found in the Spanish legal regulation on this matter [2], which states that solar system installations are mandatory in new buildings. Obviously, the energy generated by these installations implies a saving on other sources of energy that would otherwise have been used for that purpose. If the energy needs are known in advance, such information may be used to predict the energy thresholds and when the available energy will no longer be demanded. [3]. As a result, it will be possible to take corrective actions with the purpose of reducing the energy from non-renewable energy sources. Various works have addressed that challenge in different ways. In [4], a new strategy was proposed for the optimal scheduling problem, taking into account the impact of uncertainties in wind, solar PV, and load demand forecasts.

Artificial Intelligence techniques and paradigms have been combined in the Hybrid Intelligent Systems (HIS) and applied to a wide variety of problems, ranging from environmental issues [5] [6] [7] to cybersecurity [8] [9].

The present paper addresses the above-mentioned forecasting problem by proposing the novel HIS, designed to predict the energy generated by solar electricity systems. The model works with the information obtained from monitoring radiation and consumption. When applied to certain problems, HIS is required to manage a huge amount of information, which implies the use of smart computing methods [10].

Intelligent techniques have previously been applied to solar-energy management: the performance of a solar collector system using Phase Change Material was experimentally investigated for 1 month in [11]. Useful energy and collector efficiency were predicted by means of Artificial Neural Networks (ANN), Adaptive-Network-Based Fuzzy Inference Systems, and Support Vector Machines (SVMs). The authors of [12] proposed the application of ANN, Random Forest, and Smart Persistence to forecast the three components of solar irradiation (global horizontal, beam normal, and diffuse horizontal). They compared the three methods when predicting hourly solar irradiations for time horizons between h+1 and h+6.

According to [13], machine-learning-based prediction models can be categorized into:

- Global models: a model, based on an available training dataset is obtained, with the aim of optimal error reduction. ANN [14] and SVMs [15] are included in the set of techniques that can be considered in this category.
- Local models: the dataset is divided into some groups with similar characteristics, by using the *k*-means clustering algorithm [16], the SOM (Self-Organizing Maps) [17], or neural gas [18]. Local models [19] are considered very helpful methods for time-series predictions.

As previously stated, a local model-based system will be proposed in this research work, in order to make reliable predictions of the power generated at a solar thermal plant. Its novelty relies on the application of some regression techniques over previously-obtained clusters, to obtain the best performing local model. The *k*-means clustering algorithm was applied at the clustering step.

The content of the following sections of this paper will be organized as follows: the proposed HIS and the applied models will be described in Section 2. The case study to which the HIS was applied will be detailed in Section 3; together with the results that will be presented in Section 4. Finally, the conclusions to the present study will be outlined in Section 5.

2 Hybrid Intelligent System

As previously stated, both supervised and unsupervised learning models are combined in the framework of an HIS to improve regression results. New models are combined to fit this prediction problem from a previously proposed diagram (Fig. 1), successfully demonstrated in [20].

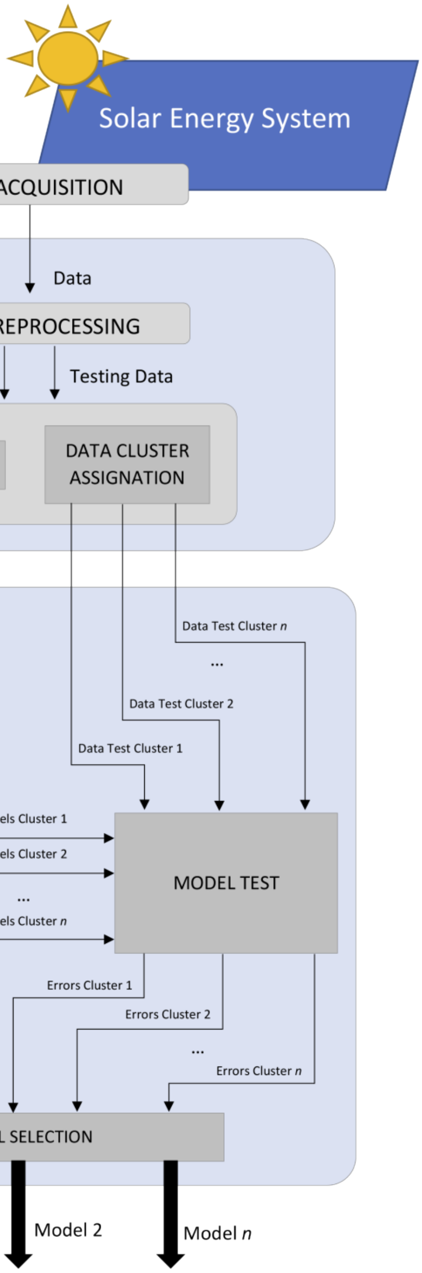


Fig. 1. Hybrid Intelligent System for Solar Energy Prediction.

The model developed in [20] is based on a combination of clustering and prediction algorithms, which obtain local models of lower complexity and accuracy than a general model for the whole system. In a first step, a cluster algorithm was applied to the dataset to identify clusters of samples with similar behavior. In a second step, a prediction algorithm was trained for each the previously identified clusters. The complete process to arrive at the final models can be summarized as follows (see Fig. 1):

- Data Acquisition: the Solar Energy System is sampled and the data are gathered.
- Data Pre-processing:
  - Filtering: the dataset is filtered and incorrect (empty and/or negative values) measures are removed.

  - Training and testing: once filtered, the dataset is divided into 2 subsets. There is a subset for training (two thirds of the filtered data: 23,091 samples) and another for testing (one third of the filtered data: 11,545 samples).

- Clustering: a clustering method is applied to the whole dataset. As a result, every training and testing sample is assigned to one of the given clusters.
- Regression: using the samples previously assigned to the clusters, a regression model is trained for each cluster and the best model is selected on the basis of the testing error.

In the first step of the original formulation of this HIS [20], both Principal Component Analysis (PCA) and the SOM were applied for data clustering. In the present paper, clustering was performed by an alternative algorithm: *k*-means. Both the Multilayer Perceptron (MLP) and the Least Square-Support Vector Machine (LS-SVM) were applied as regression models. The Radial-Basis Functions Network (RBFN) was introduced as a new regression model and the MLP was once again applied, for comparative purposes. In the interests of a comprehensive comparison, 5 different learning algorithms were tested when training the MLP.

As in the original paper, a 10-fold cross validation [21] scheme was followed, in order to carry out exhaustive tests on the regression techniques.

2.1 Local Models

When a system presents different behaviors that can be clustered into groups, the application of machine-learning techniques to multiple local models will obtain better results than a single global model [13] [20].

In a nutshell, local models can be defined as low-complexity models that are created over small regions of the input space. On the contrary, a global model works over the whole input space. Consequently, the knowledge extracted from each local model (associated to a certain region of the input space) provides a more comprehensive and interesting vision of the dataset to be analyzed. The final system of local models consist of a group of experts with specific training over specific regions of the dataset. Then, the final goal function (*f(x)*) can be seen as the union of the goal functions (*f_i(x)*) of each local model (*M*), as stated in (1).

$$f(x) = \bigcup_{i=1}^{M} \hat{f}_l(x) \qquad (1)$$

In the present research, the local models were based on samples of the system working in a similar area of behavior. For each local model, various regression techniques were applied, in order obtain the best fit with the behavior of the real system.

Local models [22] can be classified into classical (linear) models and extended (non-linear) models. In this research, the non-linear local models were applied.

Several algorithms can be used during the clustering step, to reveal the structure of the analyzed dataset, such as *k*-means, standard winner-take-all competitive learning, SOM, fuzzy competitive learning, and PCA, among others. In the present study, the widely-used *k*-means algorithm was applied to define the clusters from the original dataset that were consequently used for training each local model.

2.2 *k*-means

Cluster analysis organizes data, grouping data samples according to a given criteria (mainly distance). Two individuals in a valid group will be much more similar than those in different groups. The clustering *k*-means algorithm [23] groups data samples into a previously defined number of groups. Two input parameters are required, in order to apply it: the number of clusters (*k*) and their initial centroids. Firstly, each data sample is assigned to the cluster with the nearest centroid. Once the groups are defined, the centroids are recalculated and a reallocation of the samples takes place. Those steps are repeated until there is no further modification of the centroids. The quality criterion to measure the grouping is the Sum of Squared Errors (SSE). The algorithm intended to minimize it can be defined as follows:

$$SSE = \sum_{j=1}^{k} \sum_{x \in G_j} \frac{p(x_i, c_j)}{n} \tag{2}$$

where, k is the number of groups, p is the proximity function, c_j is the centroid of group j, and n is the number of data samples. From among all the proposed distances, the authors applied the main ones, for this clustering algorithm [24]. After comparing the results, the Cityblock distance was selected. It is a distance measure where each centroid is placed in the component-wise median of all the samples in the group. The distance from point x to each of the centroids was calculated as:

$$d_{st} = \sum_{j=1}^{n} \left| x_{sj} - y_{tj} \right| \tag{3}$$

where, j is an instance of the vector, x.

2.3 Radial-basis Function Network

The RBFN [25] is a neural network where a centroid is associated with each node in the hidden layer. For each of the input vectors, $x = (x_1, x_2, \ldots, x_n)$, it computes the distance between the node centroid and x. The output of the unit is then calculated as a non-linear function of this distance. Finally, the output of the hidden nodes is weighted and combined in the nodes of the output layer.

In the case of r input nodes and m output nodes, the response function of each one of the output nodes can be calculated [25] as:

$$\sum_{i=1}^{M} W_i * k\left(\frac{x - z_i}{\sigma_i}\right) = \sum_{i=1}^{M} W_i * g\left(\frac{\|x - z_i\|}{\sigma_i}\right) \tag{4}$$

where, x is an input vector, M Є $\mathbb{N}$ is the number of hidden units; W_i □ $\mathbb{R}^m$ are the weights linking the ith hidden-layer unit to the output nodes; K is a kernel function that is radially symmetric; σ_i is the smoothing factor of the ith kernel node; z_i is the centroid factor of the ith kernel node; and, g: $[0, \infty) \rightarrow \mathbb{R}$ is the activation function.

2.4 *Multilayer Perceptron*

The MLP is a well-known ANN consisting of several layers of nodes. There are weights associated with the connected nodes and the output signals are generated by calculating the activation from the sum of the inputs. Its architecture consists of an input layer that passes the input vector to the other layers of the network. The terms “input vectors” and “output vectors” refer to the inputs and outputs of the MLP and are represented as single vectors [26]. Additionally, an MLP has one or more hidden layers, together with the output layer. MLPs are fully connected; that is, every node is connected to each one of the nodes in the previous and the following layer.

During training, the updating of weights is performed with the backpropagation algorithm. In this study, the following implementations of this algorithm were applied: Bayesian Regularization (BR); Scaled Conjugate Gradient (SCG); Batch Training with bias and weight learning rules (RB); Gradient Descent with adaptive learning rates and momentum (GDX) and, the Levenberg-Marquardt (LM) algorithm.

2.5 Multiple Linear Regression

Multiple Linear Regression (MLR) models the relationship between a target variable and some explanatory variables by generating a linear equation for the observed data [27]. A value of the target variable (y) is linked to a value of the independent variable (x). The regression line for p explanatory variables $(x_1, x_2, ..., x_p)$ is defined as follows:

$$u_y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p \tag{5}$$

The fitted values b_0, b_1, ..., b_p estimate the parameters β_0, β_1, ..., β_p of the regression line. Then, equation 5 describes the variation of the mean response, u_y, according to the explanatory variables. The observed values for y change around the mean response, u_y, and the same standard deviation is computed (σ).

The model can be expressed as DATA = FIT + RESIDUAL, where the "RESIDUAL" term refers to the deviations of the observed values, y, from the mean response, u_y, and "FIT" is represented by (5).

2.6 Multiple Non-Linear Regression

A different type of regression was also applied in this study, called Multiple Non-Linear Regression (MN-LR). Here, the data are modelled by a function that depends on (one or more) independent variables [28] where the model parameters are non-linear combinations. The parameters can be of any type of non-linear function, such as trigonometric, exponential, etc. An iterative algorithm is used to determine the non-linear parameter:

$$y = f(X, B) + \varepsilon \tag{6}$$

where, X is the criterion variable; B is the non-linear parameter estimate to be computed; and, ε is an error term.

3 Case Study: Solar Energy Prediction

The proposed HIS was used for the prediction of energy obtained from a real-life installation. The system under analysis was a solar thermal panel forming part of the RE systems installed in a bioclimatic house in Galicia (north-western Spain). This thermal system is shown in Fig. 2 and can be divided into two functional sections: energy capture and energy storage. The continuous red line in Fig. 2 demarcates the part of the thermal system that corresponds to hot liquid, while the dotted blue line is associated with cold liquid. The solar thermal part is based on solar panels placed on the north façade of the building, inclined at 19º. This part is connected to the hot liquid accumulator through a closed circuit employing ethyleneglycol. The fluid is circulated by a hydraulic pump that transports the ethyleneglycol to the solar panel inputs (S1 and S2). The solar panels are divided into two sections working in parallel, each consisting of four simple thermal solar panels. The ethyleneglycol in circulation through the panels makes it possible to capture the heat from the sun. The fluid then flows out through S3 and S4 at a higher temperature. After that step, the hot fluid is taken to the heat exchanger of the accumulator (S8) where the heat is transferred to the stored water. The liquid is then pumped back to the thermal solar panels as it leaves the accumulator (S6).

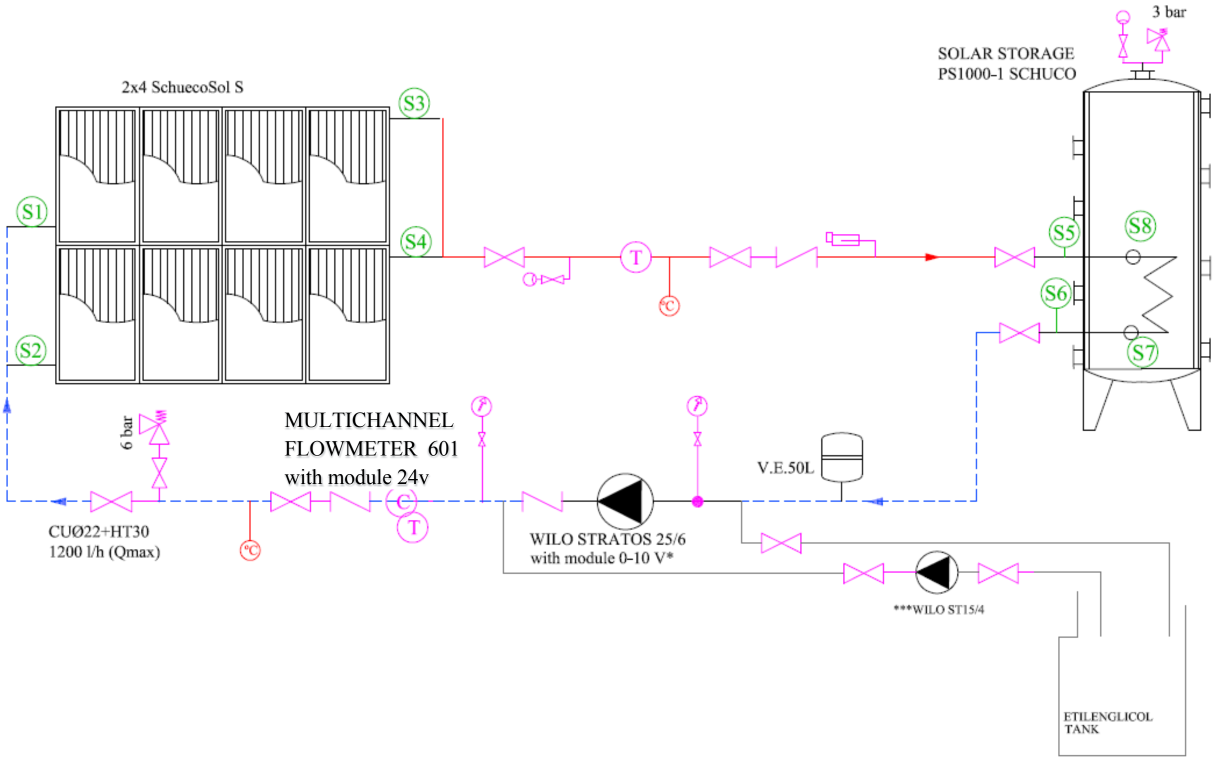


Fig. 2. Case Study under analysis: solar thermal system.

When the bioclimatic house was designed, it was established that the solar thermal installation would cover approximately 90% of the hot water needs. The performance of the solar panels was 78.1%, and losses of between 5% and 10% were estimated, due to their inclination. The specific location of the bioclimatic house required an inclination of 43º. However, the criterion was to assume the losses, but taking the same inclination as the roof of the building.

3.1 *Dataset*

The proposed HIS was validated with data for one year, after the energy from the power system installed in the previously described bioclimatic house had been sampled. These data samples were split into 12 different datasets, by the month of the year in which they were collected (from September to December, 2010 and from January to August, 2011).

The equipment used for their measurement was as follows:

- Power meter: Kamstrup type Multichannel 601, capable of measuring temperature, flow, and thermal power.
- Radiation meter: Apogee model PYR-P, capable of measuring solar radiation with a sensitivity of 0.200 mV per Wm^{-2}.

The output for each data sample was the thermal power generated (in Watts). On the other hand, the following six inputs were associated with each data sample:

- Flow: water flow of the solar thermal system (in m^3/h)
- Solar Radiation: measured radiation level over the panels (W/m^2)

- Ta: input Temperature on the lower panel (ºC)
- Tb: input Temperature on the upper panel (ºC)
- Tc: output Temperature on the lower panel (ºC)
- Td: output Temperature on the upper panel (ºC)

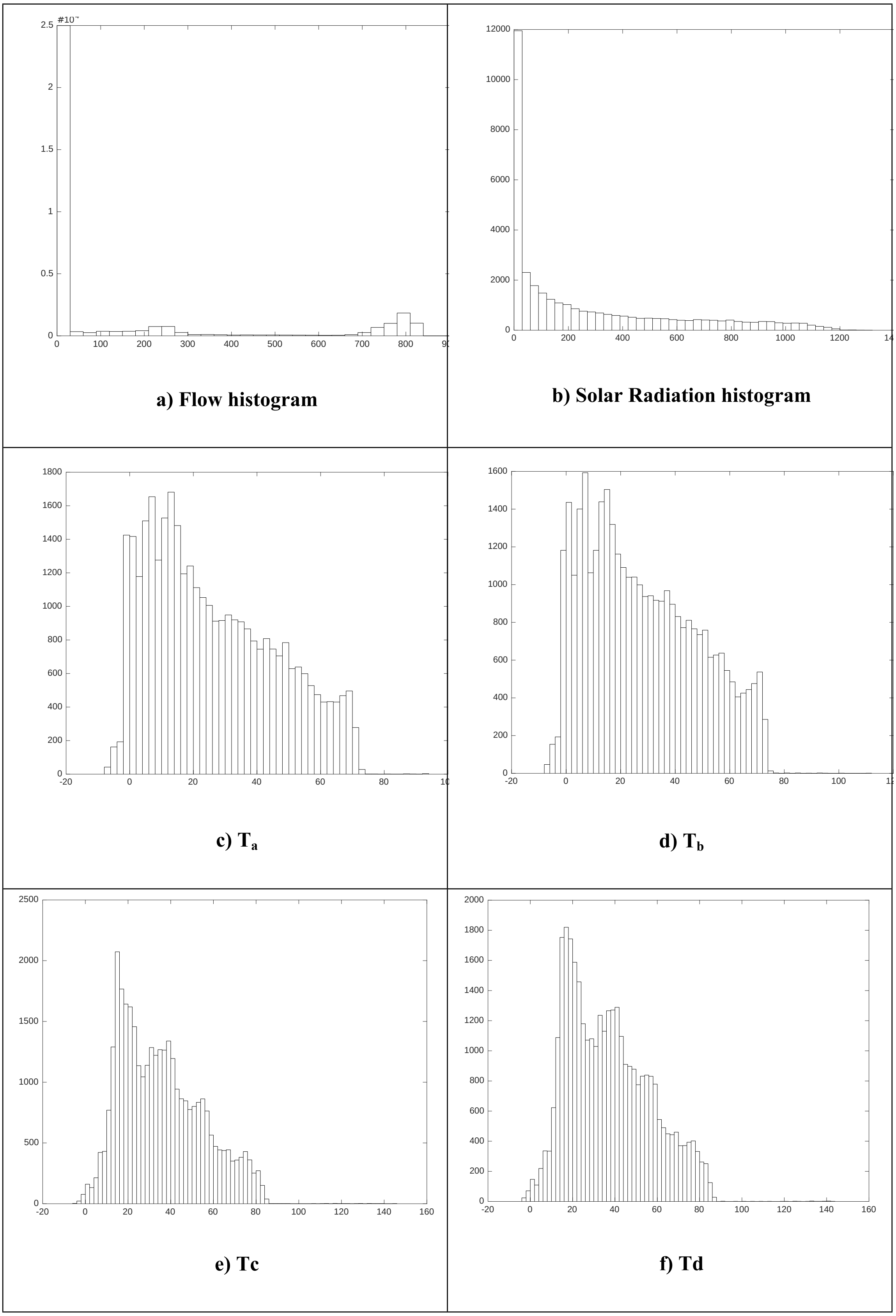
Each component (both input and output) was recorded with a 10-minute measurement period. The dataset analyzed by the HIS was previously described in [20]. It was obtained by pre-processing the raw data and removing all outliers, obtaining a final dataset composed of 34,636 samples.

In Table 1, the dataset was presented in terms of the diversity and the scale of each variable, representing each variable and its maximum, minimum, and average values. Based on those results, a normalization process between [0,1] was needed, as the variables presented very different ranges of values.

Table 1. Range of values for each variable: maximum, minimum, average.

	Flow	Solar Radiation	Ta	Tb	Tc	Td
Maximum	**1200.00**	**1303.69**	**93.65**	**110.33**	**144.95**	**142.88**
Minimum	**0.00**	**-0.24**	**-7.29**	**-7.11**	**-4.13**	**-3.94**
Average	**130.30**	**175.49**	**20.79**	**22.36**	**33.12**	**34.37**

The frequency distribution histograms of each variable are shown in figure 3 (from 3.a to 3.f), in order to present the diversity of the dataset variables. Comparing the histograms, it can be concluded that variable flow and solar radiation presented similar sample distributions with a remarkable tail to the left of the histogram. It corresponds to periods of time when no power was produced (very small flow and very little radiation).



a) Flow histogram

b) Solar Radiation histogram

c) T_a

d) T_b

e) Tc

f) Td

Figure 3. Frequency distribution histograms of each variable

The correlation matrix between all variables is shown in Table 2. There is only one clear correlation between variables Tc and Td.

Table 2. Variables correlation matrix

	Flow	Solar Radiation	Ta	Tb	Tc	Td
Flow	1	0.50572176	0.04717336	0.05336151	-0.0574075	-0.0355189
Solar Radiation	--	1	0.18187044	0.18509537	-0.0767962	-0.0431432
Ta	--	--	1	0.23368599	0.25093049	0.27284253
Tb	--	--	--	1	0.22828923	0.24805207
Tc	--	--	--	--	1	0.93438843
Td	--	--	--	--	--	1

Finally, after a normalization process for each variable [0,1], all of them are compared by means of a boxplot in Figure 4. The image shows several outliers that are removed in the case of variable flow, to produce the final dataset.

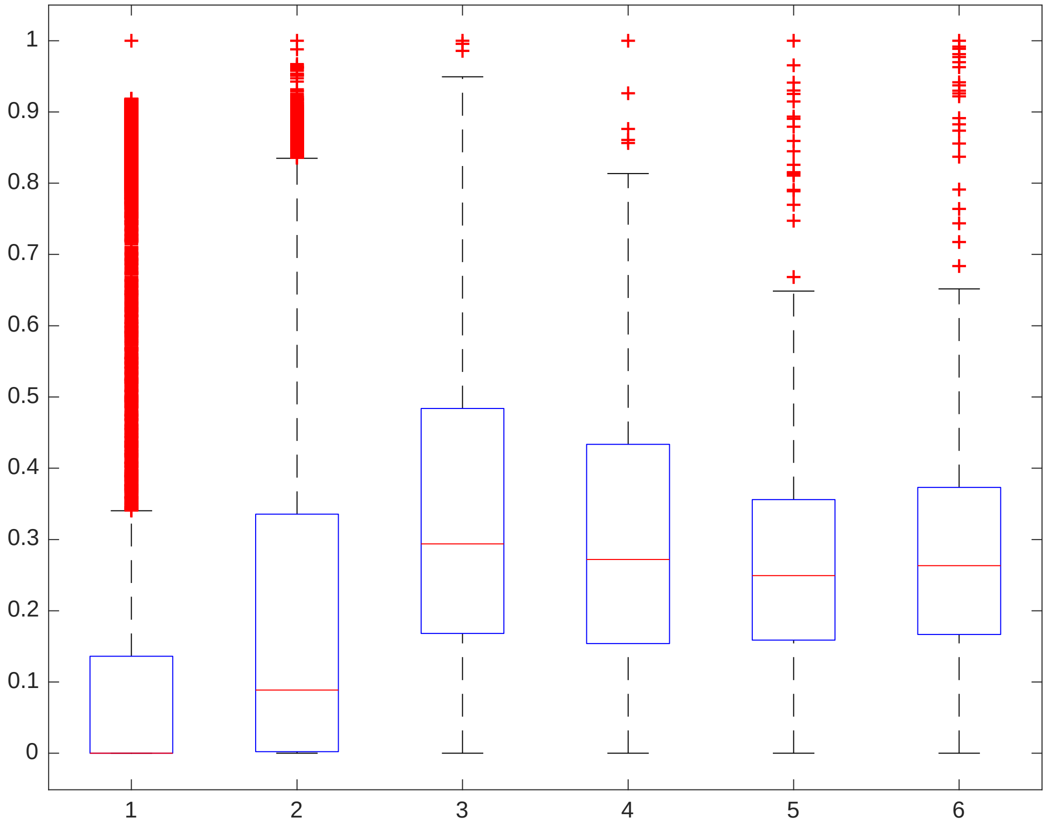


Figure 4. Boxplot of each variable after the normalization process

4 Results & Discussion

The HIS described in Section 2 was applied to the different datasets described in Section 3.1:

1. 12 datasets corresponding to the 12 months of the year without grouping the data (12-Month datasets).
2. 36 datasets, three groups for each of the 12 months, resulting from applying the *k*-means clustering technique to the whole dataset (36-Grouped datasets). A value of 3 was selected for the *k* parameter, as it is the optimum number of groups previously identified [20] for the dataset under analysis.

The regression methods were applied to these two datasets, in order to predict the thermal power generated by the solar system. They were validated by the *n*-fold Cross-Validation (CV) scheme. CV is a technique that splits the data, in order to measure the error of each algorithm, into two subsets (training and testing). The training samples were used for training each algorithm, while the testing samples were used for its validation. Finally, the algorithm with the smallest CV-estimated error was selected [29]. The number of the *n* parameter (data partitions) was set to 10 (standard value) for all the experiments in the present study. The total number of samples for each month before and after clustering (with *k*-means and Cityblock distance measure) are shown in Table 3. The data were split in training and testing subsets for the CV.

Table 3. Total number of samples for each month (both training and testing subsets).

Month	Total		Cluster 1		Cluster 2		Cluster 3	
	Train	Test	Train	Test	Train	Test	Train	Test
January	2979	331	2716	301	180	20	84	9
February	2977	330	2500	277	267	29	211	23
March	3011	334	2305	256	283	31	423	47
April	2456	272	2328	258	84	9	44	5
May	2717	301	1628	181	532	59	557	61
June	2745	305	1745	194	509	56	500	55
July	2700	300	1813	201	419	46	469	52
August	2699	300	1802	200	451	50	447	49
September	2546	282	1787	198	395	43	365	40
October	1885	209	1241	137	305	34	340	37
November	1767	196	1578	175	152	16	38	4
December	2695	299	2388	265	276	30	32	3

The process of training the neural models was repeated 10 times (once for each *fold*). Moreover, training was repeated 10 times for each training algorithm with the same combination of parameters in the case of MLP. The main purpose of this repetition was to reduce the effect of randomness and obtain more representative results. The Normalized Mean Squared Error (NMSE) for all ten *folds* is presented for all the experiments (Tables 4 to 11). The NMSE is a regression performance metric calculated as the mean of the squared errors.

In the case of the MLP and RBFN, experiments with a varying number of hidden neurons (10, 20, and 30) were conducted. For the sake of brevity, only the results obtained with a configuration of 10 hidden neurons are shown. The main reason is that adding more neurons would significantly increase the execution times, but not the accuracy of the results, according to the NMSE. The training algorithm had a stronger influence on the results than the number of neurons in the hidden layer. The best balance between execution speed and NMSE was achieved when using 10 hidden neurons.

A sigmoid activation function in the hidden layer and a linear activation function in the output layer was employed by MLP. A radial-basis transfer function was applied in the case of RBFN.

4.1 Results from the 12-month Datasets

The results (considering both NMSE and execution time) of applying the previously described models to the 12-month datasets are presented in this section (Tables 4 to 7). To begin with, the results obtained by MLR and MN-LR are shown in Table 4.

Table 4. MLR and MN-LR results for the 12-Month datasets.

Month	MLR		MN-LR	
	NMSE	Time (s)	NMSE	Time (s)
January	2.6E-04	0.1617	**7.7E-05**	0.2513
February	2.5E-04	0.1270	**6.2E-05**	0.1777
March	2.3E-04	0.1227	**3.9E-05**	0.1909
April	3.4E-04	0.1274	**3.1E-04**	0.2006
May	2.2E-04	0.1214	**4.2E-05**	0.1929
June	2.3E-04	0.1228	**4.1E-05**	0.2007
July	2.4E-04	0.1247	**4.2E-05**	0.2078
August	2.3E-04	0.1242	**3.9E-05**	0.2013
September	2.6E-04	0.1269	**7.5E-05**	0.2121
October	3.3E-04	0.1255	**5.1E-05**	0.1856
November	4.4E-04	0.1234	**1.6E-04**	0.1800
December	2.8E-04	0.1256	**8.3E-05**	0.1986

In Table 4, similar results over the 12-month period may be seen for both regression techniques. The results were constant for all the datasets in the case of MLR. The MN-LR technique obtained lower error rates than MLR for all the datasets. On the contrary, the execution times of MN-LR were higher than those of MLR. Better values can be observed for both algorithms in the NMSE for the spring and summer seasons.

Table 5. RBFN results for the 10 folds of the 12-Month datasets.

Month	**NMSE**		**Time (s)**	
	Mean	**STD**	**Mean**	**STD**
January	2.1E-05	6.1E-07	0.0641	0.1270
February	2.3E-05	5.6E-07	**0.0374**	0.0816
March	**1.6E-05**	3.2E-07	0.0378	0.0813
April	2.3E-05	1.1E-06	0.0388	0.0836
May	2.3E-05	4.9E-07	0.0384	0.0835
June	2.3E-05	3.5E-07	0.0421	0.0955
July	2.0E-05	3.6E-07	0.0386	0.0837
August	2.1E-05	5.9E-07	0.0403	0.0880
September	3.1E-05	7.5E-07	0.0403	0.0882
October	2.4E-05	6.5E-07	0.0402	0.0879
November	3.0E-05	1.2E-06	0.0404	0.0889
December	2.8E-05	1.0E-06	0.0394	0.0862

In the case of RBFN (Table 5), constant error rates and execution times were obtained for the 12 months. In comparison with previous results (Table 4), RBFN obtained lower error rates in all months. Considering the execution times, RBFN also proved to be faster than the regression techniques. Likewise, worth noting are the low values of the standard deviation (STD) of the NMSE, which confirms the heterogeneity in the results of the 10 *folds* over all twelve months. The month of March received a lower value in the calculation of the NMSE and achieved a lower NMSE, see Table 4.

Table 6. MLP results for the different training algorithms and 10 folds of the 12-Month datasets.

Month	**NMSE - Mean and STD**									
	LM		**GDX**		**RB**		**SCG**		**BR**	
	Mean	**STD**	**Mean**	**STD**	**Mean**	**STD**	**Mean**	**STD**	**Mean**	**STD**

January	4.2E-06	8.9E-07	9.6E-04	8.2E-04	5.0E-04	3.9E-04	5.4E-06	3.7E-07	**3.5E-06**	2.9E-07
February	2.4E-06	2.0E-07	2.9E-03	3.1E-03	5.2E-04	4.5E-04	3.8E-06	1.2E-06	**2.1E-06**	1.9E-07
March	1.4E-06	1.6E-07	2.1E-03	2.0E-03	2.5E-04	1.8E-04	2.5E-06	3.9E-07	**1.2E-06**	1.1E-07
April	**6.3E-07**	2.5E-07	2.0E-03	5.2E-04	7.3E-04	1.5E-04	1.7E-06	1.1E-06	6.8E-07	1.2E-07
May	6.6E-06	2.8E-07	1.6E-03	1.3E-03	4.7E-04	2.3E-04	8.1E-06	2.8E-07	**6.3E-06**	2.6E-07
June	5.4E-06	4.3E-07	1.7E-03	1.4E-03	6.3E-04	5.2E-04	7.7E-06	2.2E-07	**5.1E-06**	3.3E-07
July	6.0E-06	2.0E-07	2.2E-03	1.5E-03	5.6E-04	2.7E-04	7.5E-06	1.8E-07	**5.8E-06**	2.5E-07
August	4.4E-06	1.8E-07	1.7E-03	1.8E-03	3.9E-04	2.5E-04	6.7E-06	5.8E-07	**4.1E-06**	1.2E-07
September	6.0E-06	5.9E-07	4.0E-03	5.5E-03	4.7E-04	2.9E-04	8.9E-06	1.5E-06	**5.2E-06**	6.8E-07
October	2.6E-06	3.9E-07	1.7E-03	1.5E-03	4.7E-04	2.6E-04	3.4E-06	1.5E-07	**2.4E-06**	4.4E-07
November	7.2E-06	1.3E-08	1.9E-03	1.6E-03	6.9E-04	3.0E-04	1.0E-05	1.1E-07	**6.2E-06**	9.5E-09
December	8.9E-06	1.2E-05	2.2E-03	6.6E-04	6.6E-04	3.6E-04	7.2E-06	1.0E-06	**4.5E-06**	4.6E-07

In Table 6, the results obtained by MLP are shown. As can be seen, the best results in the calculation of the NMSE were obtained by the BR training algorithm in 11 of the 12 months, with similar results obtained by the LM training algorithms. Furthermore, those three training algorithms obtained better results than RBFN (Table 5), in terms of the NMSE. In a similar way to the content of Tables 4 and 5, the month of March once again showed the best results in terms of NMSE, which may be due to low variability of the input parameters to the solar panel.

Table 7. MLP execution times for the different training algorithms and 10 folds of the 12-month datasets.

Month	**Time (s) - Mean and STD**									
	LM		**GDX**		**RB**		**SCG**		**BR**	
	Mean	**STD**	**Mean**	**STD**	**Mean**	**STD**	**Mean**	**STD**	**Mean**	**STD**
January	**0.071**	0.105	0.185	0.015	0.414	0.083	0.182	0.015	0,117	0,093
February	**0.062**	0.070	0.167	0.007	0.335	0.032	0.109	0.008	0.091	0.062
March	**0.066**	0.072	0.177	0.007	0.332	0.044	0.058	0.011	0.068	0.058
April	**0.067**	0.069	0.176	0.007	0.328	0.027	0.215	0.028	0.387	0.054
May	**0.068**	0.064	0.180	0.010	0.351	0.032	0.081	0.006	0.123	0.011
June	**0.068**	0.071	0.192	0.012	0.349	0.032	0.071	0.011	0.241	0.015
July	**0.072**	0.062	0.190	0.015	0.359	0.029	0.074	0.009	0.091	0.064
August	**0.075**	0.056	0.193	0.010	0.411	0.039	0.084	0.011	0.076	0.049
September	**0.070**	0.067	0.189	0.009	0.381	0.026	0.175	0.028	0.208	0.084

October	**0.070**	0.068	0.179	0.010	0.363	0.036	0.080	0.009	0.157	0.025
November	**0.070**	0.066	0.177	0.009	0.353	0.032	0.058	0.007	0.070	0.024
December	**0.072**	0.065	0.186	0.010	0.390	0.079	0.179	0.010	0.088	0.118

In Table 7, the results obtained in terms of execution time by the MLP are shown. As can be seen, the LM training algorithm obtained the best results over the 12 months. The LM training algorithm also obtained slightly shorter execution times than the RBFN algorithm, as can be seen in Table 5. Comparing these results with those shown in Table 6, it can be concluded that the LM algorithm achieved good execution times and was one of the most effective at NMSE calculation.

4.2 Results from the 36-Grouped Datasets

The results in terms of NMSE (Tables 8 to 10) and execution time (Table 9), obtained when applying the same models to the 36-Grouped datasets, are presented below. The results are shown according to the cluster (C1, C2, or C3) to which each data item is assigned. First, the results obtained by MLR and MN-LR are shown in Table 8.

Table 8. MLR and MN-LR results for the different clusters in the 36-Grouped datasets.

Month	**MLR (NMSE)**			**MN-LR (NMSE)**		
	C1	**C2**	**C3**	**C1**	**C2**	**C3**
January	1.8E-04	6.1E-04	8.9E-05	1.6E-04	4.6E-04	**8.8E-05**
February	1.8E-04	4.1E-04	1.5E-04	1.6E-04	3.4E-04	**7.5E-05**
March	2.1E-04	3.7E-04	8.9E-05	1.9E-04	3.0E-04	**1.9E-05**
April	**2.4E-05**	1.4E-04	2.0E-03	2.4E-05	5.7E-05	6.5E-04
May	2.6E-04	2.1E-04	7.2E-05	2.2E-04	1.6E-04	**3.8E-05**
June	2.1E-04	2.1E-04	8.3E-05	1.8E-04	1.5E-04	**4.3E-05**
July	2.4E-04	2.7E-04	7.6E-05	2.1E-04	2.1E-04	**3.6E-05**
August	2.6E-04	2.5E-04	8.2E-05	2.2E-04	1.6E-04	**3.8E-05**
September	2.0E-04	3.0E-04	1.2E-04	1.8E-04	2.2E-04	**7.4E-05**
October	3.8E-04	3.3E-04	1.0E-04	3.4E-04	2.0E-04	**3.0E-05**
November	3.2E-04	6.4E-04	3.6E-04	2.9E-04	5.4E-04	**2.3E-04**
December	2.4E-04	4.0E-04	1.9E-04	2.1E-04	2.4E-04	**1.8E-04**

Comparing the results of these two regression techniques (Table 8), they can be said to be similar for all the months. The best results in the calculation of the NMSE were obtained by the MN-LR regression technique for 11 (out of 12) months and in the

C3 cluster. Additionally, the regression techniques applied to the grouped data outperformed their application to the 12-month datasets (Table 4).

Table 9. RBFN results for the 10 folds of the 36-Grouped datasets.

Month	NMSE - Mean			Time (s) - Mean		
	C1	C2	C3	C1	C2	C3
January	**5.6E-05**	2.9E-04	7.5E-05	0.0109	0.0107	0.0102
February	**5.3E-05**	2.2E-04	7.1E-05	0.0103	0.0096	0.0092
March	**1.8E-04**	2.0E-04	**1.8E-05**	0.0099	0.0097	0.0099
April	**4.2E-06**	4.2E-05	5.9E-04	0.0178	0.0103	0.0101
May	1.1E-04	1.2E-04	**3.5E-05**	0.0096	0.0097	0.0098
June	6.9E-05	1.3E-04	**4.0E-05**	0.0104	0.0099	0.0096
July	1.8E-04	1.7E-04	**3.6E-05**	0.0095	0.0095	0.0094
August	8.2E-05	1.3E-04	**3.4E-05**	0.0099	0.0096	0.0104
September	1.7E-04	1.4E-04	**7.3E-05**	0.0099	0.0098	0.0097
October	3.2E-04	1.5E-04	**2.9E-05**	0.0100	0.0107	0.0099
November	2.8E-04	3.1E-04	**2.1E-04**	0.0096	0.0095	0.0098
December	**1.1E-04**	1.7E-04	1.6E-04	0.0110	0.0095	0.0093

After analyzing the results of RBFN for the 36-Grouped datasets (Table 9), it can be highlighted that the NMSE values were, in general terms, lower than those obtained by the regression techniques on the same data and better than the results obtained by RBFN on the 12-Month dataset (Table 5). In a similar way to Table 8, cluster C3 obtained the lowest NMSE in 8 (out of 12) months. Obviously, execution times were in this case also smaller, as the data had been split into 3 different clusters. If these results are compared with those obtained in Table 4, a better NMSE was only obtained in cluster C3. Once again, the results underline that the lowest NMSE values were obtained in March.

Table 10. MLP results for the 10 folds of the 36-Grouped datasets.

Month	Cluster	NMSE - Mean				
		LM	GDX	RB	SCG	BR
January	C1	4.7E-05	1.8E-03	9.5E-04	2.8E-05	3.4E-05
	C2	8.2E-05	4.5E-03	8.6E-04	1.0E-04	1.0E-04
	C3	2.3E-05	1.6E-03	3.7E-04	7.1E-06	**7.9E-06**

February	C1	5.5E-05	2.1E-03	8.5E-04	2.4E-05	6.3E-05
	C2	5.8E-05	3.0E-03	8.8E-04	7.7E-05	6.1E-05
	C3	6.7E-06	1.3E-03	4.1E-04	6.2E-06	**3.6E-06**
March	C1	1.1E-04	1.8E-03	1.1E-03	2.6E-05	8.9E-05
	C2	9.1E-05	2.8E-03	6.9E-04	1.1E-04	9.0E-05
	C3	1.0E-06	1.1E-03	2.6E-04	1.8E-06	**9.9E-07**
April	C1	8.5E-07	1.8E-03	3.4E-04	1.8E-06	6.9E-07
	C2	**4.7E-07**	9.2E-04	4.6E-04	9.6E-07	5.6E-07
	C3	2.4E-04	3.5E-03	2.3E-03	4.1E-05	8.2E-04
May	C1	3.1E-05	2.2E-03	9.0E-04	4.4E-05	3.0E-05
	C2	4.7E-05	1.3E-03	4.9E-04	6.4E-05	5.5E-05
	C3	**1.3E-05**	1.1E-03	2.3E-04	2.1E-05	2.0E-05
June	C1	9.1E-05	3.3E-03	7.7E-04	3.7E-05	2.8E-05
	C2	6.9E-05	1.7E-03	5.8E-04	8.1E-05	7.5E-05
	C3	1.5E-05	5.6E-04	2.2E-04	1.5E-05	**9.6E-06**
July	C1	3.1E-05	1.8E-03	6.8E-04	3.9E-05	3.1E-05
	C2	7.7E-05	1.5E-03	6.2E-04	9.1E-05	8.7E-05
	C3	2.7E-05	1.4E-03	1.9E-04	1.9E-05	**1.5E-05**
August	C1	3.3E-05	1.9E-03	1.1E-03	3.4E-05	2.2E-05
	C2	5.0E-05	2.7E-03	7.6E-04	6.1E-05	5.4E-05
	C3	1.7E-05	6.9E-04	2.4E-04	1.6E-05	**1.1E-05**
September	C1	3.7E-05	1.2E-03	5.4E-04	3.8E-05	2.8E-05
	C2	6.9E-05	1.6E-03	4.7E-04	9.2E-05	7.8E-05
	C3	2.0E-05	8.1E-04	2.5E-04	1.9E-05	**1.2E-05**
October	C1	1.0E-04	1.4E-03	8.6E-04	4.0E-05	1.5E-04
	C2	2.5E-04	1.9E-03	6.4E-04	6.9E-05	6.3E-05
	C3	**1.9E-06**	1.4E-03	4.3E-04	3.5E-06	2.2E-06
November	C1	7.0E-05	2.1E-03	1.3E-03	2.9E-05	8.2E-05
	C2	3.1E-04	2.1E-03	1.1E-03	1.1E-04	1.4E-04
	C3	**9.3E-06**	2.4E-03	6.0E-04	2.7E-05	3.2E-05
December	C1	5.8E-05	1.4E-03	1.5E-03	2.4E-05	7.2E-05

	C2	1.3E-04	2.4E-03	1.1E-03	4.4E-05	3.9E-05
	C3	3.3E-04	2.0E-03	5.6E-04	**9.5E-06**	1.8E-05

The results of applying the MLP to the 36-Grouped datasets are shown in Table 10. The best results (according to the mean NMSE) were obtained when applying the BR algorithm, and similar ones were obtained by the LM algorithm. In a similar way to Table 6, both algorithms obtained the best results, with slightly better results from the clustered data (mainly cluster C3). Figures 5 and 6 show the boxplots corresponding to the data in Table 10. In Figure 5, the results are grouped by cluster (including all the months, folds and algorithms) and in Figure 6 they are grouped by training algorithm.

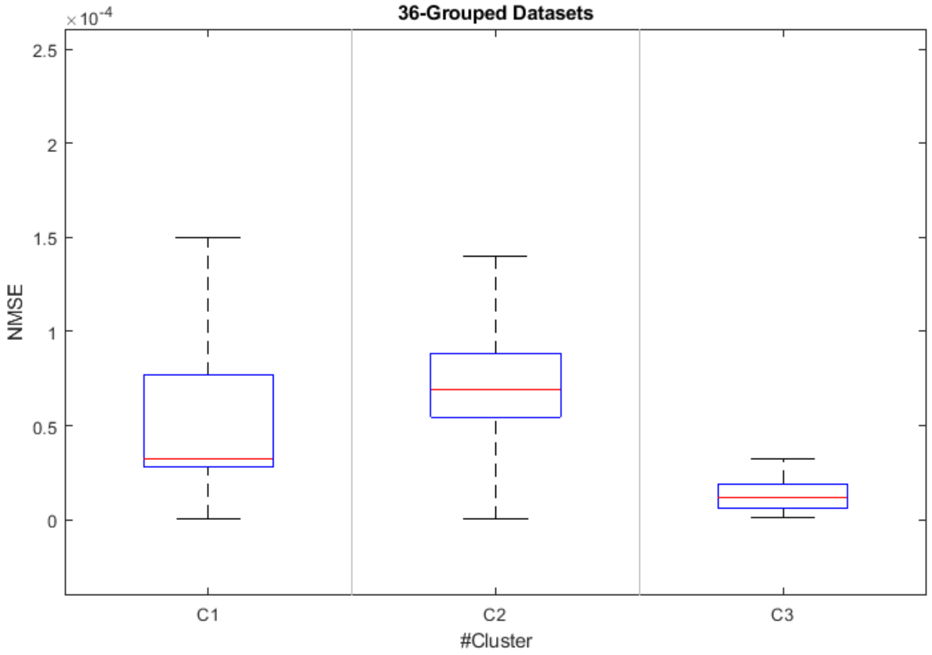


Fig. 5 Boxplot of the results in Table 8, grouped by Clusters.

It can be seen from Fig. 5 that cluster C3 has the lowest mean and deviation, while cluster C1 has the highest deviation and cluster C2 the highest mean NMSE.

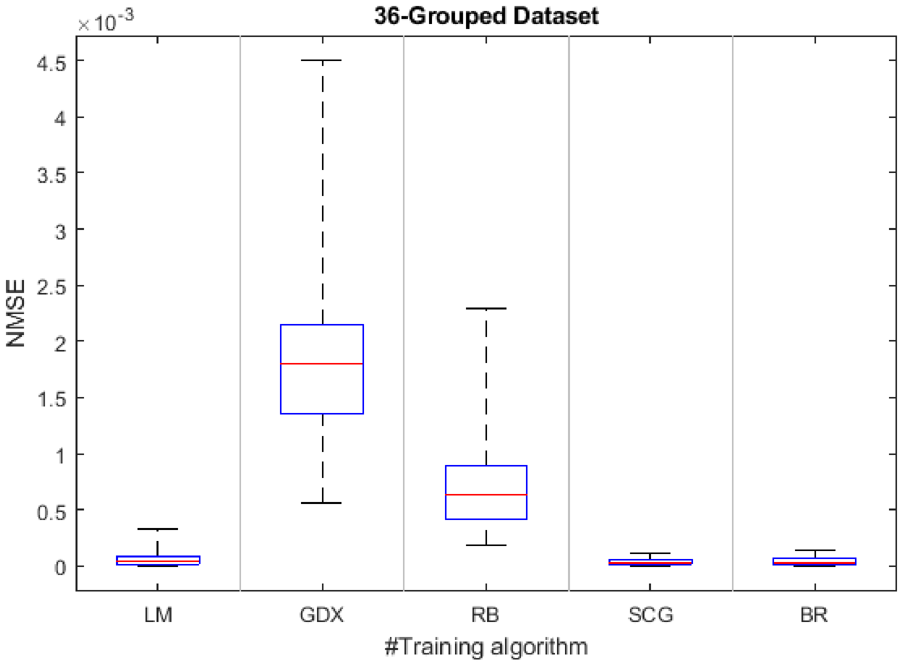


Fig. 6 Boxplot of the MLP results (by training algorithm) when applied to the 36-Grouped Datasets.

In Fig. 6, the training algorithms, LM, SCG, and BR with the lowest mean values of NMSE and the smallest deviations can be clearly seen.

Fig. 7 shows the fitting between the predicted values (red solid line) and the real output (blue dashed line) for October, when applying the MLP with the BR training algorithm. The error for each data point is also shown (black stars).

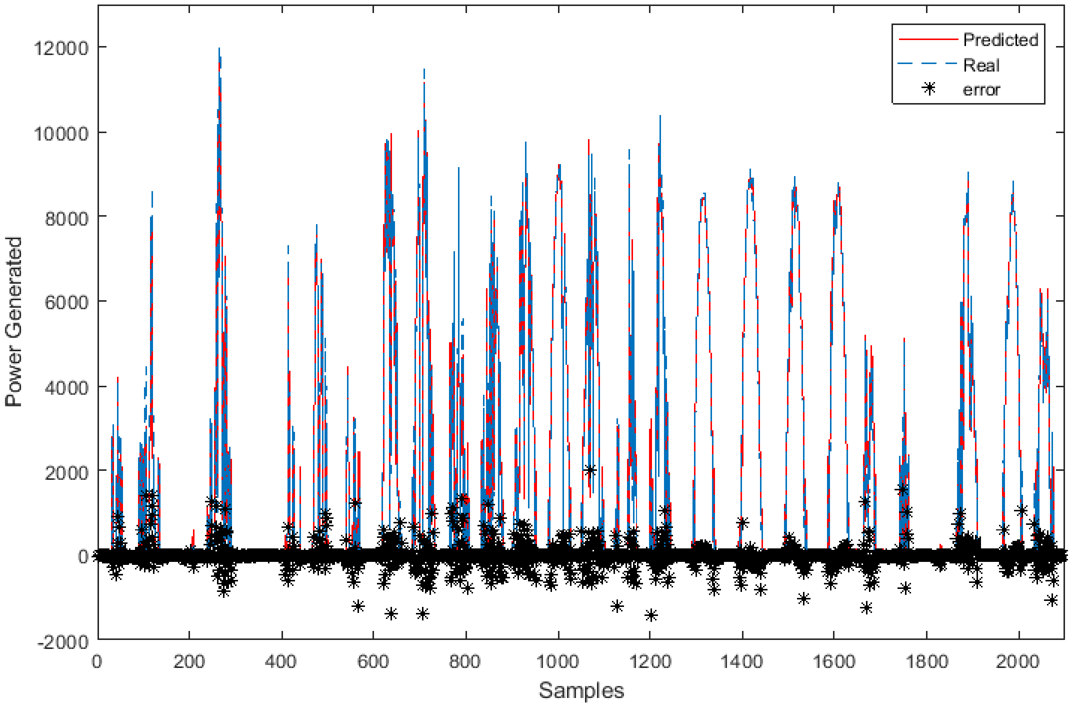


Fig. 7: Fit between the October dataset and the BR training algorithm (MLP).

The MLP results in this study (Tables 6 and 10) were better than those obtained by the original HIS. The predicted values were also of greater accuracy, thanks to the local model regressions (clustering with *k*-means algorithm) that were applied. It is important to emphasize the differences found between the five training algorithms applied, not only in the execution times but in the NMSE that was obtained. Also, the applications of CV techniques together with the multiple executions of each experiment, together with the adjustment of the parameters in both RBF and MLP, all contribute to make these results possible.

5 Conclusions and Future Work

The present study has attempted to forecast the energy generated within a solar thermal system by performing a regression on the objective feature.

The dataset (six inputs and an output) has been tested with 2 multiple regression (linear and non-linear) techniques and 2 neural models (RBFN and MLP trained with different algorithms) have been compared and validated through a 10-fold cross-validation. All these techniques have been applied to the 12-month datasets and to the 36-gouped datasets (generated by applying the *k*-means clustering technique with the Cityblock distance).

Some conclusions can be drawn from the results shown in section 4:

1. For the 12-Month datasets, without clustering (Section 4.1), very low and constant values of mean NMSE were obtained for the twelve months. The best results were obtained when applying the MLP trained with the BR and the LM training algorithms (Table 6). The MLR and MN-LR regression techniques obtained the worst results (Table 4).
2. The results for the 36-Grouped datasets, Section 4.2, were in general terms slightly better than those described in Section 4.1, The ANNs (RBFN and MLP) obtained better results than the regression techniques (MLR and MN-LR in Table 8). As happened for the previous datasets, the best results were obtained by MLP trained with the BR and LM algorithms (Table 10).
3. The data split in the clusters led to better results in terms of the mean NMSE for most of the data, but not in all cases (with respect to the results on the ungrouped data). The data clustering by *k*-means greatly improved the results when compared to the SOM that was applied in the original HIS.

Future work will focus on predicting the energy generated by some other renewable sources as well as applying more advanced regression models to reduce prediction error.

Funding: This research has received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

1. Demirbas, A. (2005). Potential applications of renewable energy sources, biomass combustion problems in boiler power systems and combustion related environmental issues. Progress in energy and combustion science, 31(2), 171-192.
2. Real Decree 1027/2007 of 20 July, which approves the Regulation Thermal Installations in Buildings (BOE No. 207 of 29.08.2007).
3. Martín, L., Zarzalejo, L. F., Polo, J., Navarro, A., Marchante, R., & Cony, M. (2010). Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning. Solar Energy, 84(10), 1772-1781.
4. Reddy, S. S. (2017). Optimal scheduling of thermal-wind-solar power system with storage. Renewable Energy, 101, 1357-1368.
5. Arroyo, Á., Herrero, Á., Corchado, E., & Tricio, V. (2017). A hybrid intelligent system for the analysis of atmospheric pollution: a case study in two European regions. Logic Journal of the IGPL, 25(6), 915-937.
6. Horenko, I. (2010). On clustering of non-stationary meteorological time series. Dynamics of Atmospheres and Oceans, 49(2-3), 164-187..
7. Woźniak, M., Graña, M., & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. Information Fusion, 16, 3-17.
8. Vega, R. V., Quintián, H., Calvo-Rolle, J. L., Herrero, Á., & Corchado, E. Gaining deep knowledge of Android malware families through dimensionality reduction techniques. Logic Journal of the IGPL, 27(2), 160-176.
9. Pinzon, C. I., De Paz, J. F., Herrero, A., Corchado, E., Bajo, J., & Corchado, J. M. (2013). idMAS-SQL: intrusion detection based on MAS to detect and block SQL injection through data mining. Information Sciences, 231, 15-31.
10. Corchado, E., Abraham, A., & de Carvalho, A. (2010). Hybrid intelligent algorithms and applications. Information Sciences: an International Journal, 180(14), 2633-2634.
11. Varol, Y., Koca, A., Oztop, H. F., & Avci, E. (2010). Forecasting of thermal energy storage performance of Phase Change Material in a solar collector using soft computing techniques. Expert Systems with Applications, 37(4), 2724-2732.
12. Benali, L., Notton, G., Fouilloy, A., Voyant, C., & Dizene, R. (2019). Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components. Renewable energy, 132, 871-884.
13. Martínez-Rego, D., Fontenla-Romero, O., & Alonso-Betanzos, A. (2011). Efficiency of local models ensembles for time series prediction. Expert Systems with Applications, 38(6), 6884-6894.
14. Bishop, C.M. and S. ligne, Pattern recognition and machine learning. 2006: Springer New York.
15. Ye, J., & Xiong, T. (2007, March). Svm versus least squares svm. In Artificial Intelligence and Statistics (pp. 644-651).
16. He, L., Long, L. R., Antani, S., & Thoma, G. R. (2011, July). Multiphase level set model with local K-means energy for histology image segmentation. In Healthcare Informatics, Imaging and Systems Biology (HISB), 2011 First IEEE International Conference on (pp. 32-39). IEEE.
17. Cherif, A., Cardot, H., & Boné, R. (2011). SOM time series clustering and prediction with recurrent neural networks. Neurocomputing, 74(11), 1936-1944.
18. Qin, A. K., & Suganthan, P. N. (2005). Enhanced neural gas network for prototype-based clustering. Pattern recognition, 38(8), 1275-1288.
19. Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. Neural computation, 3(1), 79-87.
20. Quintián, H., Calvo-Rolle, J. L., & Corchado, E. (2014). A hybrid regression system based on local models for solar energy prediction. Informatica, 25(2), 265-282.
21. Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).
22. Souza, L. G. M., & Barreto, G. A. (2010). On building local models for inverse system identification with vector quantization algorithms. Neurocomputing, 73(10-12), 1993-2005.
23. MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability(Vol. 1, No. 14, pp. 281-297).
24. Zhuang, W., Ye, Y., Chen, Y., & Li, T. (2012). Ensemble clustering for internet security applications. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(6), 1784-1796.
25. Park, J., & Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. Neural computation, 3(2), 246-257.
26. Pal, S. K., & Mitra, S. (1992). Multilayer Perceptron, Fuzzy Sets, Classification. IEEE Transactions on neural networks, 3(5), 683-697.
27. Yale University. Multiple Linear Regression. 2017; Available from: http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm

28. Kutner, M. H., Nachtsheim, C., & Neter, J. (2004). Applied linear regression models. McGraw-Hill/Irwin.
29. Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. Statistics surveys, 4, 40-79.