Energy Production Estimation for PV Plants. A Methodological Comparative Study

Marcelo Cortés-Carmona[1,2,3], Mauricio Trigo-González[2], Pablo Ferrada-Martínez[2,3], Javier Batlles[4,5], Juan Carlos Acosta[1], and Joaquín Alonso-Montecinos[4,5]

[1] Department of Electrical Engineering, University of Antofagasta, 02800 Antofagasta, Chile

[2] Centre for Energy Development (CDEA), University of Antofagasta, Antofagasta, Chile

[3] Solar Energy Research Centre (SERC-Chile), Santiago (Chile)

[4] Department of Chemistry and Physics, University of Almería, 04120 Almería, Spain

[5] CIESOL, Joint Centre University of Almería-CIEMAT, 04120 Almería, Spain

Abstract

Nowadays, due to the increasing integration of solar energy into electrical systems, it is necessary to improve methodologies used to analyze this type of technology. In this regard, a topic that has had permanent attention is solar irradiation forecast and photovoltaic (PV) plant energy production estimation. This study analyzes the performance of three methodologies commonly used in PV plant energy production estimation: Multiple Linear Regression (MLR), Artificial Neural Networks (ANNs), and Support Vector Machines (SVMs). Calculations lead to the conclusion that SVMs usually produce lower training errors. However, the CPU used by this type of algorithm is significantly higher than those required by ANN and MLR. Considering this, an MLR-based algorithm was developed to deliver training errors similar to those delivered by ANNs and SVMs, but with significantly less processing time and use of computing resources. The algorithm proposed (F-MLR-solar2) is at least 1000000 times faster than SVM and 5000 times faster than ANN. Additionally, the proposed algorithm allows obtaining a simple equation that can be used analytically.

Keywords: Solar energy, PV energy estimation, Support Vector Machines, Multiple Linear Regression.

1. Introduction

Today, human civilization requires abundant, clean, and safe energy supply at a reasonable cost. In response to this need, massive integration of renewable energies can provide a path to tackle current world issues. Abundant sources like solar energy and their implementation at residential, commercial or utility-scale level are showing a great potential and becoming an important economic activity. Photovoltaics has many advantages, one of them being the option to install just a few panels or thousands of them, or the possibility to inject electricity into the grid. However, PV plants can convert sunlight into electricity only during daylight. In addition, even at daylight, solar irradiance can change drastically in a minute (C.A.B. Vaz, L.N. Canha, 2018). These issues are reflected in the non-optimal operation of a PV plant or, even more drastic, clouds passing over a large PV installation may produce strong instabilities in the electrical system (M. Ebad, W.M. Grady, 2016).

Plant operation is exposed to external factors like variability of solar resources, weather and environmental conditions, soiling, and material degradation. Therefore, sensing the main influencing parameters as well as measuring electrical output values are needed. This action generates large datasets, which must be stored and processed. One way to deal with these data is to use machine learning methods, tools that have provided good results in problems of classification, energy estimation, irradiance prediction, and failure prediction, among others (Bishop, 2006). These approaches can be used for solar radiation estimation (Voyant et al., 2017) and PV plant production (Assouline et al., 2017). There is a broader view for categorizing machine learning methods, namely, supervised (Dacheng Tao et al., n.d.), semi-supervised, and unsupervised learning (Greene et al., 2008). While supervised approaches are based on datasets containing both input and output values, unsupervised approaches contain only inputs. Therefore, supervised methods are often used for solar energy applications. The unsupervised group can be used to find commonalities within a dataset.

A great research effort has been made to develop methodologies to predict the production of a PV plant (J.

doi:10.18086/swc.2019.14.03 Available at http://proceedings.ises.org

Antonanzas, 2016). The problem can be classified into methodologies oriented to production forecast with a short-term (1 min - 3 hours), medium-term (day-ahead), and long-term horizon and methodologies oriented to production estimation. Trigo et al. (2019) propose a methodology to estimate the hourly energy production in terms of energy yield, i.e, kWh/kWp, within infra hour time frame using a minimal knowledge of weather conditions. The methodology used was Multiple Linear Regression (MLR) applied to distinct technologies in different Chilean Regions.

This study aims to compare three methodologies to estimate the hourly energy production of PV plants. The methodologies applied are MLR, Artificial Neural Networks (ANN) and Support Vector Machines (SVM). In addition, an efficient methodology is proposed to estimate PV plant production at a very low computational cost.

2. Location, instrumentation, and photovoltaic (PV) system

This section presents the main background of the photovoltaic installation used in this study. It also explains the procedures used to condition information measured.

2.1 Climatology

University of Almeria, UAL, (36°49′N 2°24′W) is located on the coastal area of Almeria city, southeastern of Spain. It is one of the Iberian Peninsula areas with the most hours of sunshine, 2965 per year.

It has the driest climate -semiarid to arid- not only of Andalusia and the Iberian Peninsula but also of the whole continental Europe in relation to geographic and dynamic factors already known. Average annual temperatures range between 15.5ºC and 19ºC, the upper end being the highest in Andalusia as a whole. Maximum annual averages are remarkably high - between 20.5ºC and 24.8ºC -, as a result of scarce maritime influence and frequent winds from the west, causing absolute summer values of up to 40ºC. Minimum average records are moderately mild, not below 10.6 º C inland - with occasional winter frosts - and up to 14 º C in warmer coastal locations. Rainfall records are generally below 350 mm/year. The torrential rainfall regime, with a very marked autumn maximum and extreme dryness during summer months fit into the pluviometric model of the Mediterranean macroclimate (Gómez-Zotano, 2015). Its climatology can be defined as a warm steppe climate with a warm summer, or BSh according to Köppen-Geiger classification (Peel, 2007).

2.2. PV system

On the roof of the CIESOL building there is a PV plant with a total installed capacity of 9.324 kWp connected to an electricity grid. It has 42 polycrystalline modules (mc-SI) of 222 Wp power each. The arrangement consists of three strings (14 modules in series) facing south with 22 degrees of inclination in operation since 2009. See Fig 1.

Fig. 1. Upper view of the photovoltaic system located on the roof of CIESOL building, at University of Almeria, Spain.

The cell model is A-222P with 13.63% efficiency. Its short circuit current (ISC) and open circuit voltage (Voc) are 7.96 A and 37.2 V, respectively. Current (Impp) and voltage (Vmpp) at the maximum power point are 7.44

A and 29.84 V, respectively. Temperature coefficients are -0.46%/ºC for Pmpp, -0.35%/ºC for Voc and 0.05%/ºC for Isc. These parameters were obtained under laboratory conditions of 1000 W/m^2, 25°C, and AM1.5 solar spectrum.

On the other hand, variables recorded in the PV plant were irradiance in the inclined plane (*G*), ambient temperature (T_b), and output electrical power (*Pout*) in the inverter. The acquisition system corresponds to PLC National instruments. Measures were recorded every 5 minutes from January to December 2010, obtaining 220736 samples.

2.3. Data analysis

First, data are filtered to eliminate atypical data that may cause errors in the generation of models. New theoretical variables are then processed and calculated to create a more robust database.

2.3.1. Normalized energy (*nEac*) calculation

The power measured, Pout (*W*), is transformed into energy (*Wh*) and divided by the installed PV plant capacity. This allows obtaining a normalized pattern to compare PV plants of different sizes and technologies. The production curve was obtained by using the trapezoidal rule, where the equation used is shown by Eq. 1

$$E_{ac} = \frac{(P_{out}+P_{out+1})*(t_n-t_{n+1})}{2}. \quad \text{(eq. 1)}$$

P_{out} is inverter output power (*W*) and t_n is recording time (minutes).

In order to normalize PV production, the energy obtained from (eq. 1) is divided by installed capacity (P_{PV}) (see eq. 2).

$$nEac = \frac{Eac}{P_{PV}} \quad \text{(eq. 2)}$$

2.3.2. Solar irradiation data

To complement data measured, variables related to solar resource and the position of the sun are calculated. They are: zenith angle (*Sza*), zenith angle cosine (*CosSza*), and extraterrestrial irradiance (G_0) (Iqbal, 1983). The zenith angle is formed from the local zenith vertical line to the normal line of the sun position and a reference point on the earth's surface (see Fig. 2). With this information, the time and height of the sun position can be determined.

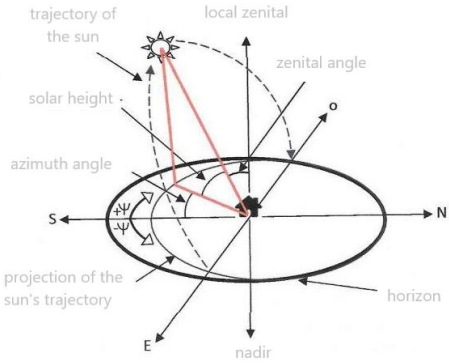

Fig. 2. Sun position.

To calculate the zenith angle, the cosine equation of the zenith angle obtained by (eq. 3) can be used.

$$CosSza = \sin(\delta)\sin(\phi) + \cos(\delta)\cos(\phi)\cos(\omega), \quad \text{(eq. 3)}$$

where δ is solar declination, ϕ is latitude, and ω is the hour angle.

Extraterrestrial irradiation (G_0) is obtained from (eq. 4). With these data, the amount of solar resource can be obtained at the level of the atmosphere, which can help identify the atmospheric attenuation of the area.

$$G_0 = Isc \cdot Eo \cdot \cos(Sza). \quad \text{(eq. 4)}$$

The term Isc is the solar constant measured in 1367 W/m^2 and Eo is the eccentricity of the sun – earth distance.

2.3.2. Performance ratio

Performance ratio (*PR*) is the ratio of actual energy production to the theoretical yield of a photovoltaic plant (see eq. 5). With this variable, the energy loss ratio can be identified due to the dust deposited from the PV panels and obtain a real approximation of the system efficiency.

$$PR = \frac{E/P_{stc}}{G_i/G_{stc}} \quad \text{(eq. 5)}$$

3. Methodologies

This section describes the algorithms used for the comparative study of models for PV plant energy production.

3.1 Multiple linear regression (MLR)

Let $T = \{\vec{x}_j, y_j\}_{j=1}^{M}$ be a dataset, where $\vec{x}_j = [x_{j1}\ x_{j2} \dots x_{jn}] \in \Re^{M\times n}$ is an n-dimensional input vector of real variables and let $y_j \in \Re^{M\times 1}$ be an output vector, M being the amount of records. These sets contain the variables associated with PV plant energy production. The components of $\vec{x}_j$ are usually solar irradiance, ambient temperature, solar zenith angle, and wind speed, among others; whereas energy production corresponds to variable y_j. The simplest way to build a prediction model for energy production is shown in (eq. 6), where $\langle \vec{w}, \vec{x}_j \rangle$ is the dot product between the vectors $\vec{w}$ y $\vec{x}_j$. Thus, ye is the estimated energy produced by the PV plant.

$$ye_j = \sum_{i=1}^{n} w_i x_{j,i} + b = \langle \vec{w}, \vec{x}_j \rangle + b \quad \text{(eq. 6)}$$

A way to find the solution to the least square problem is to minimize the squared error between measured and estimated values through (eq. 6). Thus, (eq. 7) shows the optimization problem that must be solved.

$$min\left\{\sum_{j=1}^{M}\left(y_j - ye_j\right)^2\right\} \quad \text{(eq. 7)}$$

The solution to (eq. 7) can be found by solving the system of linear equations defined by (eq. 8) and (eq. 9), with $\vec{1} = [1\ 1\ ..\ 1]^t \in \Re^{M\times 1}$, vector of 1s.

$$\sum_{k=1}^{n}\langle \vec{x}_k, \vec{x}_i \rangle w_k + \langle \vec{x}_i, \vec{1} \rangle b = \langle \vec{y}, \vec{x}_i \rangle,\ i = 1,..,n \quad \text{(eq. 8)}$$

$$\sum_{k=1}^{n}\langle \vec{x}_k, \vec{1} \rangle w_k + Mb = \langle \vec{y}, \vec{1} \rangle, \quad \text{(eq. 9)}$$

For PV plant production estimation problems, the number of unknowns required by MLR is usually low (less than 20). This fact makes the complexity of the problem be low, which allows to obtain the solution in a very short time, some milliseconds. Although there are databases with a high number of records (for instance, if records are taken every 5 minutes for 1 year, the number of records will be 105120), MLR can obtain the solution very quickly (some milliseconds) because these large databases are operated by making dot products such as $\langle \vec{x}_k, \vec{x}_i \rangle$, which are processed at a high speed. The only limitation could be the size of the computer memory. If so, the problem can be split into smaller problems and continue solving the sub-problem even on small computers. This technique is used in this study.

3.2. Artificial neural networks (ANN)

Let us consider a problem where, after an experiment has been conducted, information relating a variable to other *n* independent variables is available, that is, $y = f(\vec{x})$ where $f(\vec{x})$ is a function relating *n* input variables $\vec{x}$ with an output variable *y*.

Let ANN be the one shown in Fig. 3, with inputs x_1, x_2, and output *y*, and two hidden layers with three neurons each. The weighting factors connecting neuron *i* with neuron *j* is denoted as w_{ij}. (P. Isai, 2004):

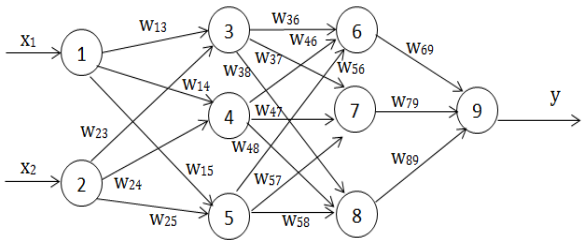

Fig. 3: Artificial Neural Network with two inputs, two layers, and one output

The output of the first layer is given by (eq. 10):

$$h^{c1} = x^t W_1 = (x_1 \quad x_2)\begin{pmatrix} w_{13} & w_{14} & w_{15} \\ w_{23} & w_{24} & w_{25} \end{pmatrix} = (h_3 \quad h_4 \quad h_5). \qquad \text{(eq. 10)}$$

The output of each neuron can be affected by an activation function, so that, $y^{c1} = f(h^{c1})$, where $f(x)$ can be any function, for example $f(x) = x$, $f(x) = x^2$, $f(x) = \frac{1}{1+e^{-x}}$ (sigmoid), etc.

By analogy, (eq. 11) gives the output of layer 2,

$$y^{c2} = f(y^{c1} W_2) = f\left(f(h_3 \quad h_4 \quad h_5)\begin{pmatrix} w_{36} & w_{37} & w_{38} \\ w_{46} & w_{47} & w_{48} \\ w_{56} & w_{57} & w_{58} \end{pmatrix}\right). \qquad \text{(eq. 11)}$$

Finally, the output in the last layer is given by (eq. 12):

$$ye(\vec{w}) = f(h_9) = f(y^{c2} W_3) = f\left(f(h_6 \quad h_7 \quad h_8)\begin{pmatrix} w_{69} \\ w_{79} \\ w_{89} \end{pmatrix}\right) = f(f(f(x^t W_1)W_2)W_3) \qquad \text{(eq. 12)}$$

Numerically speaking, (eq. 12) shows that the output of a neural network is a sequential product of vectors by matrices. The network is completely characterized when the values of the input variables and the weighting coefficients are known. For the network above mentioned there is an $n_v = n \cdot n_{c1} + n_{c1} \cdot n_{c2} + n_{c2}$ coefficient, where n is the number of input variables and n_{c1} and n_{c2} are the number of neurons in layers 1 and 2. These parameters are the unknowns of the problem. The structure defined in (eq. 10) – (eq. 12) can be used to find a prediction model for PV plant production. For this purpose, it is necessary to formulate an optimization problem. In this regard, the objective function can be defined as the mean squared error between measured values for a PV plant and the output delivered by ANN. The optimization problem to be solved is shown in (eq. 13):

$$min\left\{\sum_{j=1}^{M}\left(y_j - ye_j\right)^2\right\}. \qquad \text{(eq. 13)}$$

In eq. 13 the design variables are the weighting factors w_{ij} of the network and y corresponds to energy produced by the PV plant, which was measured over a long period. Additionally, ye is the estimated energy produced by the PV plant which is obtained from the network defined in (eq. 10) – (eq. 12).

Equations (10) – (12) were programmed in MATLAB, obtaining an algorithm that allowed training a network under the supervised learning scheme. The Nelder-Mead Simplex method was used as an optimization technique as it delivered the best results in the simulations. The program allows the use of up to two layers of a neural network and an arbitrary number of neurons per layer. It can be used as an activation function that can be linear, quadratic or sigmoid, the latter delivering the best results.

3.3. Support vector machines

SVMs are nonlinear models based on theoretical results from the statistical learning theory. This theoretical framework formally generalizes the empirical risk minimization principle that is usually applied for ANN training. An SVM classifier minimizes the generalization error by optimizing the tradeoff between empirical training errors and the so-called Vapnik-Chervonenkis (VC) dimension, which is a new concept of complexity measure Vapnik (2000). In this section, a very brief introduction to SVMs is presented.

Consider the training set defined in section 3.1, that is, $T = \{\vec{x}_j, y_j\}_{j=1}^{M}$, where $y_j = \{-1, +1\}$ is a label that determines the class of $\vec{x}_j$. SVMs employed for two classes of problems are based on hyperplanes that separate data so that all the points with the same label are on the same side of the hyperplane. An orthogonal vector $\vec{w}$ and a bias b, which identifies the point that satisfies $\langle \vec{w}, \vec{x} \rangle + b = 0$, determine the hyperplane. Among the separating hyperplanes, there is one whose distance to the closest points is maximum. This hyperplane is called optimal separating hyperplane (OSH), for which the distance to the closest point is defined by $1/\|\vec{w}\|$. The quantity $\rho = 2/\|\vec{w}\|$ is called margin. Thus, OSH is the separating hyperplane which maximizes margin ρ. The margin can be seen as a measure of the generalization ability. The hyperplane with the largest margin on the training set can be completely determined by the nearest point to itself, those being called support vectors (SVs) because the hyperplane (i.e. the classifier) depends entirely on them.

In general, the binary classification using SVM is formulated as a problem of quadratic optimization, as shown in (eq. 14) and (eq. 15) (K. Muller, 20001).

$$min\left\{\frac{1}{2}\|\vec{w}\|^2 + C\sum_{j=1}^{M}\xi_j\right\} \quad \text{(eq. 14)}$$

$$s.t.\, y_i\left[\langle \vec{w}, K(x, \vec{x}_j)\rangle + b\right] \geq 1 - \xi_j,\ \xi_j \geq 0, j = 1, .., M, \quad \text{(eq. 15)}$$

where ξ is a slack-variable, $K(\vec{x}, \vec{x}_j) = \langle \Phi(\vec{x}), \Phi(\vec{x}_j)\rangle$ is a kernel function, and C is a regularization constant that determines the tradeoff between the empirical error $\sum_{j=1}^{M}\xi_j$ and the complexity term. A large C corresponds to the assignation of a higher penalty to the training error.

The problem defined in (eq. 14) and (eq. 15) cannot be solved directly since $\vec{w}$ lies in the features space. This drawback is overcome when the dual optimization problem defined in (eq. 16) and (eq. 17) is set out.

$$max\left\{\sum_{j=1}^{M}\alpha_j - \frac{1}{2}\sum_{j,i=1}^{M}\alpha_j\,\alpha_i y_j y_i K(\vec{x}_j, \vec{x}_i)\right\} \quad \text{(eq. 16)}$$

$$s.t.\ 0 \leq \alpha_j \leq C, j = 1, .., M,\ \sum_{j=1}^{M}\alpha_j y_j = 0 \quad \text{(eq. 17)}$$

Once vector $\vec{\alpha}^0$, the solution of the maximization problem (16) and (17) has been found, the optimal separating hyperplane $(\vec{w}^0, b^0)$ has the expansion defined in (eq. 18) and (eq. 19):

$$\vec{w}^0 = \sum_{j=1}^{M}\alpha_j^0 y_j \Phi(\vec{x}_j) \quad \text{(eq. 18)}$$

$$b^0 = \frac{1}{|J|}\sum_{j\in J}\left(y_j - \sum_{j=1}^{M} y_j\,\alpha_j^0 K(\vec{x}_j, \vec{x}_i)\right), \quad \text{(eq. 19)}$$

with $J = \{j: 0 \leq \alpha_j \leq C\}$

Considering the expansion (eq. 20) of $\vec{w}^0$ and b^0, the hyperplane decision function can be written as,

$$f(\vec{x}, \vec{\alpha}) = sgn\left(\sum_{j=1}^{M}\alpha_j^0 y_j K(\vec{x}_j, \vec{x}_i) + b^0\right). \quad \text{(eq. 20)}$$

SVMs can also be applied to regression problems by introducing an alternative loss function (SVR). The loss function must be modified to include a distance measure. Vapnik (2000) proposed an ϵ-insensitive loss function that enables a sparse set of support vectors to be obtained. In this study SVR is used with an ϵ-insensitive loss function. The ϵ-insensitive loss function is shown in (eq. 21).

$$|y - f(\vec{x}, \vec{\alpha})|_\epsilon = \begin{cases} 0, & if\ |y - f(\vec{x}, \vec{\alpha})| \leq \epsilon \\ |y - f(\vec{x}, \vec{\alpha})| - \epsilon, & otherwise \end{cases}. \quad \text{(eq. 21)}$$

3.4. Feature space

As presented in the sections 3.1 and 3.3, linear regression and SVM use hyperplanes to identify a model for data values. The key point of the model is to find the $\vec{w}$ and b coefficients of the hyperplane$\langle \vec{w}, \vec{x}_j\rangle + b$. This idea presupposes that data follow a linear law. When the data trend is non-linear, an alternative is to modify the feature space of the input data. In this case, input vectors need to be mapped to an input feature data space having a larger dimension (Vapnik, 2000). This concept is clarified with the following example. Let us suppose that a 2-dimensional space follows a quadratic law of form $x_1^2 + 2x_1x_2 + x_2^2$. A new feature space with 5 dimensions can be defined. Its structure is the following: $z_1 = x_1$, $z_2 = x_2$ $z_3 = x_1^2$, $z_4 = x_2^2$, $z_5 = 2x_1x_2$. In

this way, data in the 2-dimensional space are mapped to the 5-dimensional space. Once they are in the new space, the problem is solved. In that case, the hyperplane to be determined has the following structure: $f(\vec{z}) = \langle \vec{a}, \vec{z} \rangle + b$.

The feature space is defined as follows:

$$\Phi(x_1, x_2) = (x_1, x_2, x_1^2, x_2^2, 2x_1x_2) = (z_1, z_2, z_3, z_4, z_5). \quad \text{(eq. 22)}$$

This methodology has been broadly used in SVM where Kernel concept was also developed, avoiding exhaustive mapping. In this case, the dot product between the vectors of the original space is required and Kernel is obtained from: $K(\vec{x}, \vec{y}) = \langle \Phi(\vec{x}), \Phi(\vec{y}) \rangle$. Equations (eq. 14) – (eq. 20) already include this methodology.

In this study, the strategy of changing the feature space is applied to MLR. The objective is to evaluate how effective it is to use a change in the feature space for this case.

The feature space of the input data for this study has a dimension of six and corresponds to the following variables: solar irradiance (G), ambient temperature (T_b), cosine of solar zenith angle (*CoSza*), solar zenith angle (*Sza*), extraterrestrial irradiance (G_0), and performance ratio (PR). Solar irradiance G and ambient temperature T_b are measured data, whereas *CoSza*, *Sza* and G_0 are calculated theoretically. PR is obtained from measurements of the energy produced and G_0 . Considering previous studies and simulations, feature spaces defined in Table 1 were selected for analysis. In the selection process, measured data were not eliminated, trying to reduce the number of variables as much as possible.

Table 1: Selected feature spaces

Name	Dimension	Var 1	Var 2	Var 3	Var 4
linearp	4	G	T_b	*CoSza*	*PR*
solar1	4	G	T_b	*CoSza*	$G \cdot e^{PR}$
solar2	4	G	T_b	*CoSza*	$G \cdot PR$
poly2	14	$(G, T_b, CoSza, PR)^2$			

2.5. Handling large volumes of data

When long periods of measurements are analyzed, the size of databases grows exponentially. For a year in which data are collected every 5 minutes, the number of recorded values is 105120. Depending on the type of methodology used, MLR, ANN or SVM, this amount of information cannot always be managed with usual computers. For this purpose, the database can be split into various clusters and calculations can be performed separately. Thereafter, creating ensembles for the remaining coefficients is required. Two techniques were considered in this study. The first one consists of taking the average for the resulting $\vec{w}$ y b. The second one is to concatenate coefficients in a unique database. Results show that both methods provide similar results. Consequently, the method of concatenating weighters into a single list was used.

4. Results

4.1. Analysis with *linearp* feature spacing

In this study, records from the experimental PV plant located at Solar Energy Research Center (CIESOL) in the south east of Spain were used. The input variables used for the training process are the feature space "linearp" that considered the following variables: Global Irradiance G, ambient temperature T_b, cosine of zenith angle θ_z, and Performance Ratio PR. The output variable, nE_{ac} defined in (eq. 2), is PV plant energy production.

To perform simulations with MLR and ANN methodologies, a program was developed in Matlab. For SVR, Toolbox Gun (1998) was modified in order to convert it into a more efficient code. ANN and SVR require numerical optimization, while MLR is solved analytically. This particularity of the MLR algorithm leads to a great difference in processing times. To obtain the minimum error in the estimation, the parameters of the ANN and SVR methodologies were calibrated by trial and error. ANN was configured with one hidden layer

of 8 neurons. For SVR, an Exponential Radial Basis (eRBF) Kernel function type was used, with parameter σ = 3. The upper bound parameter was defined as C=∞ and the ϵ-insensitive parameter of loss function was set to ϵ = 0. Table 2 shows the results of standard Root Mean Square Error (nRMSE) for different sample rates of each algorithm for the test data set.

Table 2: Estimation errors for test data with different algorithms

Sampling rate [min]	Data quantity	Train data quantity	Test data quantity	MLR nRMSE [%]	SVM nRMSE [%]	ANN nRMSE [%]
5	33729	23610	10119	15.6001	2.5149[(a)]	5.4669
10	17002	11901	5101	15.5883	2.1580[(b)]	4.5139
15	11243	7870	3373	14.1898	2.0326	6.2276
20	8646	6052	2594	16.5159	2.5961	5.1467
30	5875	4113	1762	17.8498	2.2149	5.5325
60	3094	2166	928	17.8342	2.9554	6.8773
Mean				**16.2630**	**2.4120**	**5.6275**
Std				**1.4313**	**0.3427**	**0.8280**

[(a)] 3 clusters were used; [(b)] 2 clusters were used;

Table 2 shows that SVR provides the lowest nRMSE for the analyzed cases. In order to have more information for analysis, Table 3 shows the number of variables that each methodology requires, along with the associated process times.

Table 3: Resources required by each algorithm

Algorithm	Train data	Input variables	Unknowns	Excecution time [s]	Solution method
SVM	11901	4	23802	2.92E+04	Numerical optimization
MLR	11901	4	5	1.56E-02	Theoretical solution
ANN	11901	4	40	7.50E+01	Numerical optimization

Table 3 shows the resources required to solve the problem with each methodology. Clearly, MLR is the most efficient from the viewpoint of computational cost and processing time. Its limitation is that it responds adequately insofar as data follow a linear trend. On the other hand, SVR requires a high computational load and high processing time. However, it allows achieving low errors. It is a machine designed for non-linear behavior. This characteristic is given by the Kernel functions. ANN is an intermediate methodology in resource use, being oriented to non-linear models. Its results are inferior to those of SVR.

4.2. Exploring other feature spaces

Table 2 shows that MLR nRMSE is quite away from what SVM and ANN deliver. In this section, the feature space is modified prior to solve MLR. The aim is to create a robust methodology having a high numerical performance and comparable to SVM and ANN results. Table 4 summarizes the results obtained with MLR for the different feature spaces defined in Table 1.

Table 4 shows a significant improvement in MLR performance. For the case of feature space solar1, the mean error was reduced from 16.263% to 4.139%, where the hyperplane is defined by (eq. 23).

$$h = w_1 \cdot G + w_2 \cdot T_b + w_3 \cdot coSza + w_4 \cdot G \cdot e^{PR} + b. \quad \text{(eq. 23)}$$

Likewise, for the feature space solar2, the mean error decreased from 16.263% to 2.534%, where the hyperplane is defined by (eq. 24).

$$h = w_1 \cdot G + w_2 \cdot T_b + w_3 \cdot coSza + w_4 \cdot G \cdot PR + b. \quad \text{(eq. 24)}$$

Table 4: MLR results for different feature spaces

Sampling rate [min]	Data quantity	Train data quantity	Test data quantity	nRMSE Φ=solar1 [%]	nRMSE Φ=solar2 [%]	nRMSE Φ=poly2 [%]
5	33729	23610	10119	4.6115	2.7774	2.7443
10	17002	11901	5101	4.1695	2.1976	2.1891
15	11243	7870	3373	3.8395	1.8381	1.8321
20	8646	6052	2594	4.4359	2.7571	2.7535
30	5875	4113	1762	4.3685	2.4872	2.4762
60	3094	2166	928	4.4915	3.1483	3.1282
Mean				**4.3194**	**2.5343**	**2.5206**
Std				**0.2770**	**0.4661**	**0.4603**

Finally, for space poly2, the error is reduced to 2.520%. The difference between poly2 and solar2 is 0.03%; however, poly2 has a dimension of 14. Despite the low error of poly2 and based on practical reasons, the chosen model solar2 defined by (eq. 23) has the advantage that a simple physical interpretation can be obtained. The simple methodology proposed competes in quality with results delivered by SVM and ANN. Table 5 shows the nRMSE average for the different methodologies studied, adding also the results obtained with ANN Matlab Toolbox.

Table 5: Comparison of sRMSE between algorithms

MLR nRMSE Φ=solar2 [%]	SVM nRMSE [%]	ANN nRMSE [%]	ANNT[(a)] nRMSE [%]
2.5343	2.4120	5.6275	2.2584

[(a)] ANN Matlab Toolbox

Table 5 shows that MLR with the modification of the feature space (F-MLR) included is competitive when compared to current methodologies, as it differs 0.27% with ANNT. However, F-MLR is 4.8 x 10^3 times faster than ANN and 1.825 x 10^6 times faster than SVM.

4.3. Application of the proposed methodology to real plants

In order to validate the proposed methodology, this section presents the application of F-MLR methodology to a life-size plant operating in the Chilean electricity system. Atacama Solar PV Plant is located at 2690 m.a.s.l. in the Antofagasta region of Chile. The PV plant has an installed capacity of 23 MWp with single axis tracker North-South, containing 77520 polycrystalline modules of 300 Wp. The variables recorded in the PV plant were: irradiance in the plane of array G, ambient temperature T_b, and electrical power output (*Pout*). The record was measured every 15 minutes from January to December 2018, obtaining 14380 samples. The sampling rate was 15 min. Table 6 shows the results of statistical error analysis in terms of nRMSE.

Table 6: Comparison of sRMSE between algorithms for Atacama Solar plant

Indicator	MLR Solar2 [%]	MLR Poly2 [%]	SVM [%]	ANN [%]	ANNT [%]
nRMSE [%]	4.6974	3.8986	3.7500	4.6425	4.2224
CPU Time [s]	0.002	0.109	2563.400	211.050	7.520
Time ratio	1	73	1,708,933	140,700	5,013

Results shown in Table 6 validate the conclusions reached in section 4.2. The F-MLR algorithm using the solar characteristic Solar2 achieves an error similar to that obtained with ANN and ANNT in a shorter process time. Fewer errors can be achieved by changing the characteristic space to poly2, but this means increasing the space dimension and process times. However, F-MLR-poly2 process time is significantly smaller than any

conventional algorithm. Finally, the last row in Table 6 shows the quotient between the CPU time used by an algorithm and the time used by F-MLR-solar2. So, these values show that F-MLR-solar2 is 5013 times faster than ANNT and 1708933 times faster than SVM.

5. Conclusions

In this study, the performance of three methodologies for PV plant energy production estimation was analyzed. Calculations reveal that SVMs usually deliver lower training errors. However, the CPU used by this type of algorithm is significantly higher than those required by ANN and MLR. This means that searching for smaller training errors requires high computational costs. Considering this aspect, an MLR-based algorithm was developed to deliver training errors similar to those delivered by ANNs, but with significantly less processing time and use of computational resources. As compared to SVM, the proposed algorithm is at least 1000000 times faster, while compared to ANN it is 5000 times faster. Finally, as shown in (eq. 24) F-MLR-solar2 has a reasonably simple formula, which allows using it analytically.

6. Acknowledgments

The authors would like to thank projects CONICYT/FONDAP 15110019 Solar Energy Research Center SERC-Chile, ING2030 CORFO 16ENI2-71940 and the PVCastSOIL Project (ENE2017-83790-C3-1, 2 and 3), which was funded by the Ministerio de Economía y Competitividad and co-financed by the European Regional Development Fund.

7. References

Antonanzas, J., Osorio, N., Escobar, R., Urraca, R., Martinez-de-Pison, F.J., Antonanzas-Torres, F., 2016, Review of photovoltaic power forecasting, Solar Energy 136, 78-111.

Assouline, D., Mohajeri, N., Scartezzini, J.-L., 2017. Quantifying rooftop photovoltaic solar energy potential: A machine learning approach. Sol. Energy 141, 278–296.

Bishop, C.M., 2006. Pattern recognition and machine learning. Springer.

Dacheng, T., Xuelong, L., Weiming, H., Maybank, S., Xindong Wu, n.d. Supervised Tensor Learning, in: Fifth IEEE International Conference on Data Mining (ICDM'05). IEEE, pp. 450–457.

Ebad, M., Grady, W.M., 2016, A cloud shadow model for analysis of solar photovoltaic power variability in high-penetration PV distribution networks, in: 2016 IEEE Power Energy Soc. Gen. Meet., IEEE, 1–5.

Gómez-Zotano, J., Alcántara-Manzanares, J. J., Olmedo-Cobo, A. Martínez-Ibarra, E., 2015, La sistematización del clima mediterráneo: identificación, clasificación y caracterización climática de Andalucía (España), Rev. Geogr. Norte Gd., no. 61, pp. 161–180.

Greene, D., Cunningham, P., Mayer, R., 2008. Unsupervised Learning and Clustering, in: Machine Learning Techniques for Multimedia. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 51–90.

Gunn, S., 1998. Support vector machine for classification and regression, ISIS Technical report, University of Southampon.

Iqbal, M, An Introduction to Solar radiation, 1983, Academic Press, Canada.

Isasi, P.; Galván, 2004, I. Redes de Neuronas Artificiales: Un Enfoque Práctico; Person Education S.A.: Madrid, Spain, pp. 1–22.

Müller, K., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B., An Introduction to Kernel-Based Learning Algorithms, IEEE Transactions on Neural Networks, vol. 12, no. 2, March 2001, pp. 181-201.

Peel, M. C., Finlayson, B. L., and McMahon, T. a., 2007, Updated world map of the Köppen-Geiger climate classification, Meteorol. Zeitschrift, vol. 15, pp. 259–263.

Trigo-González, M., Batlles, F.J., Alonso-Montesinos, J., Ferrada, P., del Sagrado, J., Martínez-Durbán, M., Cortés, M., Portillo C., Marzo, A., 2019, Hourly PV production estimation by means of an exportable multiple

linear regression model, Renewable Energy 135, 303-312.

Vaz, C.A.B., Canha, L.N., 2018, Metaheuristic applied to very short term dispatch microgrids based on cloud coverage, in: 2018 IEEE PES Transm. Distrib. Conf. Exhib. - Lat. Am., IEEE, 1–5.

Vapnik V., 2000. The Nature of Statistical Learning Theory, 2nd ed. Springer Science, New York.

Voyant, C., Notton, G., Kalogirou, S., Nivet, M.L., Paoli, C., Motte, F., Fouilloy, A., 2017. Machine learning methods for solar radiation forecasting: A review. Renew. Energy 105, 569–582.