Received: 30 June 2017 | Revised: 5 May 2018 | Accepted: 6 June 2018

DOI: 10.1002/pip.3060

EU PVSEC PAPER

WILEY PROGRESS IN PHOTOVOLTAICS

Machine learning PV system performance analyser

Sandy Rodrigues[1,2] | Helena Geirinhas Ramos[2] | Fernando Morgado-Dias[1,3]

[1] Madeira Interactive Technologies Institute, Funchal, Portugal

[2] Instituto de Telecomunicações, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal

[3] Universidade da Madeira, Funchal, Portugal

Correspondence
Sandy Rodrigues, Madeira Interactive Technologies Institute, Funchal, Portugal.
Email: sandycarmo@hotmail.com

Funding information
ARDITI PhD Studentship, Grant/Award Number: M1420-09-5369-FSE-000001; Projeto Estrategico LA 9, Grant/Award Number: UID/EEA/50009/2013

Abstract

Machine learning techniques (MLTs) can create accurate predictions of solar outputs that are used in photovoltaic system performance analysis. The issue with MLT application is the requirement for large amounts of historical data for training the prediction models that is not always available. Since the photovoltaic system behaviour is non-linear due to the unpredictable nature of the weather conditions throughout the year, MLT training requires annual historical data to create the prediction model. The photovoltaic system production meters only store up to 3 months of historical power values. This information served as motivation to research different types of MLTs, in search of one that would accurately predict the daily solar energy values of a photovoltaic system based on the available 3-month historical data. The aim of this work is to implement a photovoltaic system performance analyser that estimates daily solar alternating current energy outputs for any rooftop photovoltaic system, based on daily solar irradiation values, without being influenced by the seasons of the year nor the photovoltaic system installation location. Therefore, 5 MLTs were studied and compared, which include the regression tree, artificial neural network, multigene genetic programming, Gaussian process, and the support vector machine for regression. Results show that the regression tree MLT provides acceptable results to be used in all locations and all seasons of the year, while the support vector machine for regression is best for spring and summer training dataset months, and the Gaussian process is best for the autumn and winter training dataset months.

KEYWORDS

Genetic Programming, Limited Historical Data, Machine Learning Techniques, Machine Learning Tribes, Prediction Model, rooftop PV System, Seasons and Locations

1 | INTRODUCTION

A photovoltaic (PV) system performance analyser is used to monitor the PV system output efficiency to ensure maximum output and maximize monetary gain on the PV system investment and consequently to shorten the payback time on the return of the investment. A PV system performance analyser has predetermined thresholds and is constantly comparing historical values to the predicted values obtained by prediction models to verify the performance and efficiency of the alternating current (AC) output of the PV systems. When the predicted values lie beyond the threshold boundaries, an alert is emitted to an operations and maintenance (O&M) team to assess the PV system fault. These O&M trips cost money, and therefore, it is important for the prediction model to be accurate to minimize the number of O&M trips and maximize the monetary gain of the PV system investment and consequently the return of the investment and payback times. Model-based approaches have been mainly used in PV system performance monitoring, and recently, the data-driven approaches, which include machine learning (ML) techniques (MLTs), have been used to monitor the performance of PV systems, since they tend to deal better with the non-linear behaviour of the PV systems. However, both approaches are highly dependent on the available onsite historical data for comparison studies in performance analysis. For the case of PV system monitoring, annual onsite historical data are required due to the unpredictable nature of the weather conditions throughout the various seasons of the year that consequently produce non-linear PV system solar production outputs.[1] For this work, the onsite historical data include daily solar irradiation values from a weather station as the inputs and daily solar AC energy values from the PV system production meter as the outputs.

However, there are issues related to the gathering of the required onsite historical data, namely, that it is time consuming (wait a year to gather data) and that monitoring systems and pyranometers are costly. Another issue is related to the unavailability of the annual historical data that are unique to each PV system since the PV system behaviour is correlated to the specific weather conditions of its location. Regarding the solar irradiation historical data, an alternative to buying a pyranometer is to use solar irradiation measurements from a nearby weather station. Regarding the solar AC energy historical data, the PV system installation companies have access to 3 months of solar AC energy historical data stored by the onsite production meter. Regarding the unavailability of the unique historical data, research shows that a possible solution to solve this issue would be to apply MLTs to train prediction models and compensate for onsite historical data that are unavailable. There are a number of ways to predict PV solar AC energy output such as model-based approaches[1-4]; however, MLT has been mostly used in this type of prediction problem.[1-4]

This information served as motivation to research the possibility of creating an MLT prediction model that does not require the long-term (unavailable) onsite annual historical data from a given PV system and take advantage of the short-term (3 mo) onsite historical data to analyse the performance of any PV system. Therefore, the aim of this work is to implement an ML PV system performance analyser using MLTs to closely estimate the daily solar AC energy output of any PV system. For a PV performance analyser to correctly detect anomalies, the predicted values have to be very accurate. In other words, the predicted values have to be closely correlated to the values of the measured data. The MLT prediction values should not be influenced by the distance to the weather station, should be indifferent to the PV system location, and should not be influenced by the different seasons of the year. In other words, the proposed prediction model is a general model that could be used as a "one-size-fits-all" prediction model, which is able to estimate the daily solar production outputs of any rooftop PV system based on the historical data that are available, which are then used to analyse the performance of a given PV system.

The main objectives of this work are to determine which MLT presents the best results when dealing with different seasons of the year (spring, summer, autumn, and winter) and with PV systems in different locations. Achieving these main objectives is done by elaborating a MLT comparative study and associating a ranking score to each of the MLTs studied in this work.

There are a vast number of available MLTs, and recently, Domingos[5] suggested that all MLTs could be organized into 5 "tribes" or "schools," in which each one is represented by its own master algorithm, as presented in Table 1.[5] This concept of the 5 MLT tribes narrows down the vast number of MLTs and provides a MLT selection scheme that allows to cover all types of MLTs to research the ML PV system monitoring topic.

To achieve the aim of this work, an MLT data-driven approach is considered to analyse the errors of 5 MLTs, namely, the regression tree, the artificial neural networks (ANNs), the multigene (MG) genetic programming, the Gaussian process, and the support vector machine (SVM) for regression (SVR), as mentioned in Table 1. These 5 MLT estimates are then compared with the measured PV system solar AC energy production data. The ML models estimate the behaviour of the system based on historical time series of inputs and targets and has no knowledge about the internal characteristics and processes of the PV system. The error analysis is based on the values of the root mean square error (RMSE), and the normalized root mean square error (NRMSE). The error analysis of the MLTs was performed in 2 comparison experiments, namely, a seasonal comparison and a location comparison. The seasonal comparison considers the months of spring (March, April, and May), summer (June, July, and August), autumn (September, October, and November), and winter (December, January, and February). The location comparison considers 3 locations in the island of Madeira. Altogether, there are 2 experiments, 7 scenarios, and 5 MLTs.

The first set of results (Section 3.1) considers the months of spring as the training dataset to produce the prediction model, which was tested with the months of summer, autumn, and winter. Therefore, these results show the effect of using one season (spring) to predict the behaviour of the other seasons. This first set of results motivated us to apply the same methodology, but instead of using spring as the training dataset to produce the prediction model, the other seasons were used as the training datasets (Section 3.2).

Many research papers report the prediction/estimation of solar radiation values, but only recently, PV solar output prediction/estimation research has taken place and is rising due to the growing number of PV system installations occurring all over the world. Since there is a significant amount of PV solar production being injected into the energy distribution grid, the energy distributor companies feel a need to have knowledge about the loads that are being injected into the grid to adjust their energy management. Only papers that consider PV output prediction/estimation models were included in the

TABLE 1 The 5 MLT tribe concept suggested by Domingos[5]

#	Tribe	Master Algorithm	Selected MLT for This Work
1	Symbolists	Inverse deduction/induction	Regression tree
2	Connectionists	Backpropagation	ANN
3	Evolutionaries	Genetic programming	MG genetic programming
4	Bayesians	Probabilistic inference	Gaussian process
5	Analogizers	Kernel machines	SVR

Abbreviations: ANN, artificial neural network; MG, multigene; MLT, machine learning technique; SVR, support vector machine for regression.

literature review, which is organized according to each of the 5 MLTs used in this work.

Not much research on solar energy output predictions using regression trees has been done. Nonetheless, Zamo et al[6] conducted an ML comparison study that included the regression tree, the SVMs, and other MLTs that are not studied in this work. The results from this work show that the SVM prediction values were better than those from the regression tree.

The ANN is the MLT that is mostly used in the PV output prediction research field. Many solar energy output prediction research papers conduct comparisons between ANN models and other ML models just as done by Hossain et al.[7] This work conducts a comparison study between ML algorithms such as the ANN, support vector regression with radial kernel, and others that are not studied in this paper. The results show that the second-best prediction values were presented by the support vector regression with radial kernel and the best predictions came from the ANN algorithm.

The genetic programming algorithm used in this work is part of the GPTIPS 2 toolbox.[8] This toolbox has been used by Aliesfahani and Shahbazian[9] to implement a maximum power point tracker for a PV system based on a solar PV cell experiment and does not compare the results to other MLTs. The toolboxes used for genetic programming and solar energy output prediction are the Brain Project toolbox by Russo et al[10] and the Fuzzy Rule toolbox by Prokop et al.[11] The Brain Project is the software tool used for the formal modelling of the data using genetic programming.[10] It compares the genetic programming model results to the ones done by Pedro and Coimbra[2] and found that their genetic programming results were worse than the ANN results. Prokop et al[11] used the fuzzy rule genetic programming model to compare with the ANN and SVR models. The results showed that the genetic programming models outperformed the ANN and SVR model results.

Not much research on solar energy output prediction using Gaussian process has been done in the past. Only recently, this topic has been explored by Huang et al.[12] This work performed a comparison between solar energy prediction models such as the ANN model, the SVR model, and the Gaussian process model. The best results were presented by the Gaussian process model.

The SVR model is very popular in the solar energy output prediction research field just as is the ANN models. A model comparison study between an ANN backpropagation model and an SVR with radial kernel model was conducted by Yang et al.[13] The results show that the SVR model outperformed the ANN model. Another MLT comparative study was conducted by Junior et al,[14] which compares the SVR with the persistence method. The results show that the SVR model presented the best results for predicting daily power production.

This work contributes to further research in the PV system monitoring field by providing information about how

- solar AC energy production behaves in different seasons of the year and in different locations and
- different MLTs deal with known and unknown test data in the different seasons and locations of the Madeira Island.

The organization of the paper is as follows: Section 2 provides information about the methodology of the experimental design of this paper as well as information about the historical data and the ML parameter settings used in Matlab to conduct the results of this research paper. Section 3 presents and discusses the results of the experimental study, and finally, Section 4 concludes the main findings of the work.

The following subsection provides a brief description of the MLTs used in this work.

1.1 | ML algorithm overview

This section briefly describes the 5 ML algorithms studied in this work, which are the regression tree algorithm, the ANN algorithm, the MG genetic programming algorithm, the Gaussian process for regression algorithm, and the SVR algorithm.

A. Regression tree algorithm

Regression trees (also known as decision trees) are defined by partitioning the input space in a repetitive manner, in which each resulting region of input space defines a local model. This partitioning of the input space can be represented by a tree, in which each region is defined by a leaf.[15] The regression tree used in this work is the binary regression tree, which is easily interpreted since it can be represented by a decision tree. Each node in the decision tree is a criterion used to split the data, and each final leaf provides the predicted value.

The binary regression tree method consists in splitting the data into 2 groups repetitively according to a threshold of one of the predictors for the quantitative predictors used in this work. The predictor and the threshold are chosen to maximize the homogeneity of the corresponding observed values in each of the resulting groups with the homogeneity being computed within each of these groups. Until a stopping criterion is reached (minimum number of data or an insufficient decrease in the variance of the group), each of the resulting group is itself split into 2. Finally, the predicted value (mean value of the observed values belonging to the resulting group) of each resulting group or leaf is calculated.

Overfitting is avoided by trimming at the splitting level, which minimizes the squared error loss function estimated by cross-validation. In the case of a new prediction, the tree is routed with the predictor to path the way through the tree until the final leaf. The forecast value corresponds to the predicted value associated to the final leaf, which is the mean value of the predictor values grouped in this final leaf.[6]

B. ANN with backpropagation algorithm

The ANNs have the capability to deal with linear and non-linear approximation schemes since assumptions are not necessary to define to relate the input and output variables.[16] These input and output variables are mapped by the ANN through neuron signal sending. These neurons are arranged in layers, in which the first layer receives the inputs, the layers in between (hidden layers) contain hidden neurons, and the last layer produces the predicted outputs. The hidden neurons receive the weighted sum of the inputs, apply an activation function to this weighted sum, and produce the predicted output. After defining the ANN structure, the ANN undergoes a training process, in which the weights are adjusted to minimize the performance function with the mean square error. The weights are used to control the activation of neurons.[2]

In this work, a regression artificial feedforward neural network also known as multilayer perceptron with hyperbolic tangent activation function and a Levenberg-Marquart training algorithm are used.

C. MG genetic programming algorithm

Genetic programming is an evolutionary ML method that evolves computer programs to perform a certain task. This is done by randomly generating a population of tree structures to then mutate and cross over the best performing trees to create a new population. This process is iterated until the population contains programs that solve the task well. When the task is building a model from observed data, the genetic programming is often known as symbolic regression.[17] Unlike traditional regression analysis, where the model structure is defined, genetic programming automatically evolves both the structure and the parameters of the prediction model. The genetic programming toolbox used in this work is called GPTIPS2 and performs symbolic regression. The GPTIPS2 uses a unique type of symbolic regression called MG symbolic regression that evolves linear combinations of non-linear transformations of the input variables.[8]

A population of trees evolve when symbolic regression is performed by using genetic programming. Each population of trees encodes a mathematical equation that predicts an output vector ($N \times 1$) by using a corresponding input matrix ($N \times M$), where N is the number of observations of the response variable and M is the number of input variables.

On the other hand, each symbolic model and each member of the genetic programming population are weighted linear combinations of outputs from a number of genetic programming trees in MG symbolic regression, where each tree is a "gene." There is an option to control the maximum complexity of the evolved models by specifying the maximum number of genes $G_{\max}$ and the maximum tree depth $D_{\max}$ that any gene may have. The linear coefficients are estimated for each model from the training data by using the ordinary least squares techniques. Given this, the MG genetic programming algorithm is able to combine the power of the classical linear regression with the ability to capture non-linear behaviour without the need to prespecify the structure of the non-linear model.[8,18]

D. Gaussian process probabilistic inference algorithm

Gaussian processes may be a Bayesian alternative to the SVMs although the SVMs are sparser and therefore faster, they do not give well-calibrated probabilistic outputs.[15] Gaussian process regression (GPR) is a powerful tool for non-linear regression based on Bayesian theory and probability theory.[12] The semiparametric approach combines the interpretability of parametric models with the accuracy of nonparametric models.

E. SVR algorithm

The support vector MLTs are known to generalize well to unknown/unseen data. In this work, an SVR algorithm with a linear kernel is used by De Leone et al.[19]

The SVR models can be used for time series prediction models and obtain high accuracies. The goal of SVR is to determine an optimal function that has less than ε deviation from the target values for the training data, so that there are no errors that are less than ε and, at the same time, the regression hyperplane needs to be as flat as possible.[19]

2 | METHODOLOGY

This section describes the methodology that was used to determine which MLT would present the best results in predicting the daily solar AC energy values. This methodology describes the historical data and presents the experiment and scenario scheme. This is followed by the description of the raw data preprocessing method, ML process, error analysis, and finally the MLT parameter settings in Matlab.

The Statistics and Machine Learning toolbox from Matlab version 2016a is the software used to train and test 4 of the 5 MLTs studied in this work, while the GPTIPS2 toolbox for Matlab is used to train and test the MG genetic programming algorithm.[8]

2.1 | Scenarios for MLT testing

This section describes the historical data as well as the experiment and scenario scheme used in this work.

The historical data of the PV systems used for training and testing the MLT estimate models were measured in the island of Madeira (Portugal). This island is an ideal location for solar energy analysis since there are many microclimates in a small amount of space making weather prediction a non-linear and difficult task to carry through due to the orography features[20] of the island that include beach and mountainous regions very close to each other as can be seen in Figure 1.

The solar irradiation values (inputs) were acquired from one weather station of the Instituto Português do Mar e da Atmosfera institution located in the city of Funchal and are daily measured in kilojoules per square metre. The solar AC energy values (observed outputs or targets) for this work were acquired from a PV system installation company called Factor Energia, in 15-min power samples. Since the acquired solar irradiation values were only available in daily measurements, the solar power values were converted into solar AC energy values in kilowatt hour. Therefore, the proposed performance monitoring system analyses the PV system performance based on daily values.

Table 2 presents information about the PV system installations regarding the distance from the weather station, the altitude, and the PV system installed power expressed in direct current (DC). The PV system A installation is 440 m away from the weather station, PV system B is 2.8 km, and PV system C is 4.1 km. The PV system

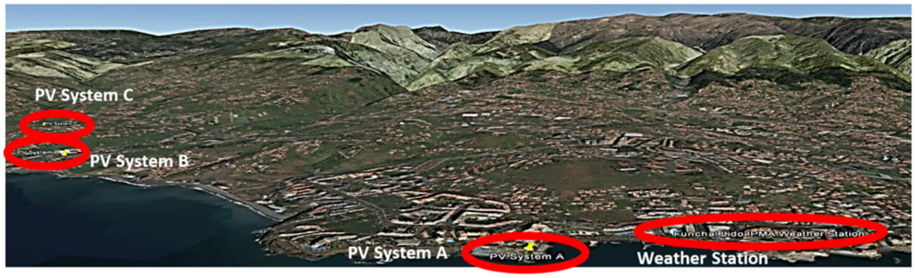

FIGURE 1 Photovoltaic (PV) system installation locations and weather station location in Funchal [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 2 PV system installation location, installed power, and sample information

	PV System A	PV System B	PV System C
Distance to weather station	440 m	2.8 km	4.1 km
Altitude	14 m	77 m	121 m
PV system installed power	4.14 kW DC	3.68 kW DC	4.14 kW DC
Train data	92 spring samples 89 summer samples 91 autumn samples 89 winter samples	...	...
Test data	92 spring samples 89 summer samples 91 autumn samples 89 winter samples 24 spring samples (May 8 to 31, 2015)	... ... 91 autumn samples 89 winter samples 24 spring samples (May 8 to 31, 2015)	... ... 24 spring samples (May 8 to 31, 2015)

Abbreviations: DC, direct current; PV, photovoltaic.

installations A and C have the same installed power of 4.14 kW, while PV system installation B has an installed power of 3.68 kW.

The input and target samples were measured in 2015 with a total of 361 daily samples from PV system A, a total of 260 daily samples from PV system B, and a total of 89 daily samples from PV system C. The training data that are used in this work came from the PV system A, which is closest to the weather station. The same weather station solar irradiation values are used in all scenarios of this work.

Table 2 indicates the number of input and target (measured data) samples that are available in each PV system dataset for training and testing the MLTs.

The training dataset is composed of spring dataset (92 samples), summer dataset (89 samples), autumn dataset (91 samples), and winter dataset (89 samples) from PV system A.

The testing dataset includes all the PV system A datasets used for training the MLTs, referred to as "known data," as well as the PV systems B and C datasets of spring, autumn, and winter referred to as "unknown/unseen" data.

The PV systems A, B, and C spring dataset from May all have 24 inputs and targets. The PV system C did not have available summer, autumn, nor winter historic data, and PV system B did not have available summer data. Therefore, the location experiment was only done in the spring, autumn, and winter seasons.

To determine which of the 5 MLTs presents the best results in PV solar AC output predictions, 2 types of experiments were conducted with the first experiment performing 4 different scenarios and the second scenario performing 3 different scenarios, as shown in Figure 2. Since there are 7 testing scenarios, there are 7 test datasets, and each scenario uses 5 MLTs.

In this work, all 5 MLTs are trained by using the train dataset from PV system A, which is the PV system that is closest to the weather station.

2.2 | Machine learning process

This section describes the preprocessing technique applied to the raw data, the ML process, the error analysis, and finally the MLT parameter settings in Matlab.

Before training the ML prediction models, the raw historical data are preprocessed by normalizing both the inputs and targets with a preprocessing technique called min-max normalization, which normalizes the values between 0 and 1. This preprocessing technique contributed to speeding up the ML model training time (not more than 10 min) since the normalized values are small and lie between 0 and 1. It was assumed that all inputs were normalized with min = 1 kJ/m^2 and max = 30 000 kJ/m^2 and all targets with min = 1 kWh and max = 30 kWh. The minimum and maximum values are based on the collected annual data and are used for generalizing the different PV system–installed power installations (up to 4.14 KW) and for detecting outliers. Since this normalizing technique is used before training the models, it then has to be reversed and compared against the predicted output values coming from the ML models to obtain the estimated daily PV solar AC energy values in kilowatt hours.

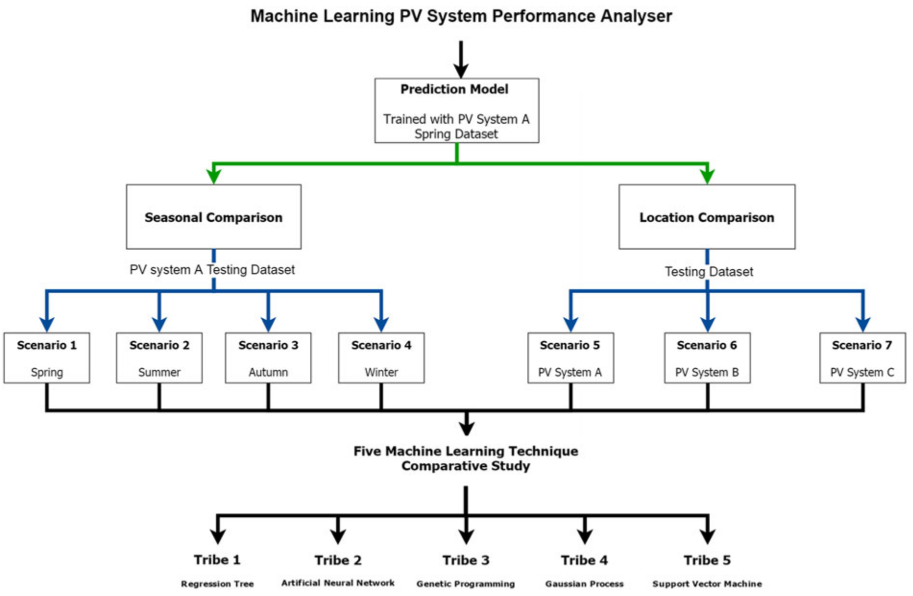

FIGURE 2 Experiment and scenario scheme of this work. PV, photovoltaic [Colour figure can be viewed at wileyonlinelibrary.com]

For a machine to execute a certain task, it needs to learn how to do the task by analysing the relationships and trends from historical data. The ML process for any MLT starts by training an ML model, then testing the ML model, to then comparing the measured data (targets) with the estimated data (outputs) that were generated by the ML model and finally performing an error analysis (Figure 3).

To compare the performance of the MLT, an error analysis is conducted by using 2 indicators, namely, the RMSE and the NRMSE, after de-normalizing the data. The RMSE is easily interpreted due to having the same units as the targets and outputs (killowatt hours) and computes the average error compared with the target (observed output) value (Equation (1)). A large positive RMSE value represents a large deviation scale in the prediction values from the target values. The accepted RMSE value in this work is an RMSE value lower or close to 6 kWh since this value represents a 2.40€/d loss (6 × 0.40€/kWh = 2.40€/d loss) assuming that the grid injection price is 0.40€/kWh (as practiced in Portugal). If this loss is an average daily loss over the course of 3 months, the total monetary loss is of 216€ (2€ × 90 d = 216€), which is quite a significant amount and is negatively reflected in increasing the investment payback time.

The NRMSE provides the average percentage value of the error (Equation (2)), in relation to the difference between the unnormalized minimum and maximum output values, which is 29 kWh (30 kWh – 1 kWh = 29 kWh) assumed in this work.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(Y_t - Y_p\right)^2}, \tag{1}$$

$$NRMSE = \frac{RMSE}{Y_{tmax} - Y_{tmin}}, \tag{2}$$

where Y_t is the observed targets, Y_p is the predicted outputs, and n is the number of observations.

The ML for regression model parameter settings for this work is mainly the Matlab default ML model settings. Validation and cross-validation were not performed to any of the models in this work unless automatically done by the fitting command of the Matlab toolbox. The MLTs were only trained with simple/default parameter settings and then tested.

The following subsections briefly explain the commands and parameter settings of each of the ML models.

A. Regression tree parameter settings

The *fitrtree* command returns a regression binary tree based on the input variables *x* and target variables *y*, where each branching node is split based on the values of a column of inputs. The rest of the settings used were the default settings.

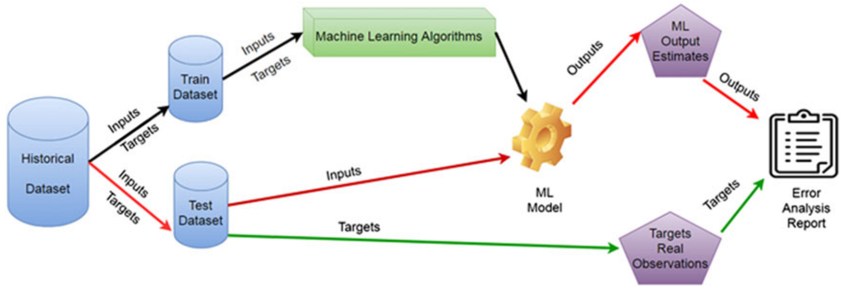

FIGURE 3 Machine learning (ML) process [Colour figure can be viewed at wileyonlinelibrary.com]

B. ANN parameter settings

The *train* command returns net model based on the following parameter settings. The parameters chosen for the ANN in this work include a multilayered feedforward backpropagation network using the Levenberg-Marquardt training algorithm, 10 neurons, 5000 epoch iterations, and the *mapminmax* normalization for the inputs and the targets. The multilayered transfer function used in this work was the tan-sigmoid and linear *purelin*. Since the initial weights of the ANNs in Matlab are random, the best model of 10 was chosen to use in this work.

C. MG genetic programming parameter settings

The model is trained by using the *rungp* training command and then to test it by using the *mymodel* testing command. The default control parameters to stop the tree evolution were set to population size of 250, timeout to 10 seconds, runs to 3, and max number of genes to 6. Since tree evolution is always different, the best model of 10 was chosen to use in this work.

D. Gaussian process parameter settings

The *fitrgp* returns a GPR model for inputs and continuous targets vector. In this work, the default kernel functions (squared exponential) and parameters (*sigma*) were used.

E. SVR parameter settings

The *fitrsvm* returns a full, trained SVM regression model trained using the input values in the matrix and the target values in the vector. The parameters chosen for the SVR include an *epsilon* value of 0.09, and the input data are standardized. The kernel used in this work for the SVR model was the default linear kernel.

3 | RESULTS AND DISCUSSION

In this section, the results of the 2 experiments and respective scenarios are presented in the tables and figures. First (Section 3.1), the results of the MLTs trained by the spring dataset of PV system A of experiments 1 and 2 (seasonal comparison and location comparison) as well as the ranking score of the MLTs are presented and discussed. Then the seasonal trends of the MLTs trained in the spring season are presented and discussed. Finally (Section 3.2), the results of the MLTs trained by the summer, autumn, and winter datasets from PV system A in both experiments are presented and discussed.

3.1 | Experiments 1 and 2 (seasonal comparison and location comparison) using the PV system A spring dataset for training

In this section, the models were trained using only the spring dataset (inputs and targets) of PV system A. The error analysis results are presented in Table 3. The highest RMSE value assumed to be acceptable in this work is 6 kWh due to the high monetary losses as explained in the error analysis description in Section 2. The corresponding NRMSE value of RMSE = 6 kWh is approximately 20.6896%, since the difference between the normalizing minimum and maximum values is 29 kWh.

Here, the MLT that presents the best daily PV solar AC energy RMSE value in scenarios 1, 5, 6, and 7 is the regression tree, in scenario 2 is the ANN, and in scenarios 3 and 4 is the SVR. These RMSE values range between 1.5228 and 5.0739 kWh in experiment 1 and between 1.2620 and 1.9410 kWh in experiment 2.

The MLT that performs second best in scenarios 1, 5, 6, and 7 is the ANN, in scenario 2 is the Gaussian process, in scenario 3 is the MG genetic programming, and finally in scenario 4 is the regression tree. All RMSE values are in the acceptable range except for scenario 4.

The third best MLT prediction values for scenarios 1 and 2 are the MG genetic programming algorithm and for the rest of the scenarios is the Gaussian process algorithm. Again, all RMSE values are acceptable (RMSE = 6 kWh) except for the winter scenario 4.

The fourth best MLT prediction values present acceptable RMSE values except for the winter scenario 4. The Gaussian process MLT presents a low RMSE value in scenario 1. For scenarios 2 and 3, the regression tree MLT presents the fourth best prediction values with

TABLE 3 Error analysis results of experiments 1 and 2 by training with the spring dataset

			Experiment 1—Seasonal Comparison				Experiment 2—Location Comparison		
Tribe	MLT	Error Analysis	Scenario 1 Spring	Scenario 2 Summer	Scenario 3 Autumn	Scenario 4 Winter	Scenario 5 PV System A	Scenario 6 PV System B	Scenario 7 PV System C
1	RT	RMSE, kWh NRMSE, %	1.5228 5.2511	2.6709 9.2101	3.5693 12.3079	5.9739 20.5997	1.2620 4.3516	1.9410 6.6931	1.3722 4.7319
2	ANN	RMSE, kWh NRMSE, %	1.8162 6.2626	2.5327 8.7334	25.7125 88.6638	15.3871 53.0590	1.6367 5.6437	2.2001 7.5864	1.5401 5.3105
3	MG genetic programming	RMSE, kWh NRMSE, %	2.0960 7.2276	2.5968 8.9544	3.4354 11.8463	6.0941 21.0141	2.2023 7.5940	2.7914 9.6255	2.1105 7.2775
4	Gaussian process	RMSE, kWh NRMSE, %	2.1128 7.2853	2.5532 8.8043	3.4595 11.9294	6.0252 20.7767	2.1985 7.5809	2.7621 9.5245	2.0089 6.9271
5	SVR	RMSE, kWH NRMSE, %	2.4515 8.4536	3.3579 11.5790	3.0377 10.4749	5.0739 17.4963	2.5274 8.7152	3.0424 10.4911	2.1229 7.3203

Abbreviations: ANN, artificial neural network; MG, multigene; MLT, machine learning technique; NRMSE, normalized root mean square error; PV, photovoltaic; RMSE, root mean square error; RT, regression tree; SVR, support vector machine for regression.

acceptable RMSE values. In scenarios 5, 6, and 7 of experiment 2, the fourth best MLT prediction values are presented by the MG genetic programming algorithm.

Finally, the worst prediction values belong to the SVR MLT in scenarios 1 and 2 from experiment 1 and scenarios 5, 6, and 7 from experiment 2. For scenarios 3 and 4, the worst prediction values belong to the ANN MLT.

Scenario 4 presents the worst prediction values when the MLTs are trained by the spring dataset of the PV system A, except when the SVR MLT is used, since the RMSE value is within the acceptable prediction values assumed in this work.

The RMSE values increase successively from scenarios 1 to 4 when using the regression tree, MG genetic programming, and Gaussian process MLTs. This does not take place with the other MLTs in experiment 1.

The ANN MLT does not deal well with the autumn and winter seasons, nor do the other MLTs except for the SVR. This is an indication of how well these MLTs deal with new unknown data that have different variations to those that take place in the spring season.

The PV systems of scenarios 5 and 7 from experiment 2 have an installed power of 4.14 kW, while the PV system in scenario 6 has a lower installed power of 3.68 kW. It is interesting to see that the PV systems with the same installed power (scenarios 5 and 7) present very similar RMSE values even though they are located 4 km away from each other with a difference in altitude of more than 100 m, which indicates that the weather conditions should be different (due to the orography of the island and consequent microclimates) in both locations and consequently affecting the solar AC energy output of the PV systems. On the other hand, the PV system with the lowest installed power (scenario 6) presents the highest RMSE values of the 3 scenarios in experiment 2. The similarity between the RMSE values of scenarios 5 and 7 might be due to the fact that both PV system installations have the same PV-installed DC power of 4.14 kW, which informs us that the model trains the behaviour of the PV solar AC energy production pattern and is sensitive to the behaviour patterns of the PV systems based on their installed power.

The prediction values of the spring season are better than the rest of the seasons. Generally, MLTs deal better with known data rather than with unseen or unknown data (summer, autumn, and winter), as shown in Table 3. The results of Table 3 indicate that the ANN and SVR MLTs are able to present the best prediction values when dealing with unknown data. All MLTs present acceptable values in all seasons of the year and all locations except for the ANN MLT in scenarios 3 and 4.

Table 4 presents the ranking score of the MLTs in each of the scenarios. This table visually represents Table 3 and places the MLTs in the correct ranking score order according to their respective scenarios.

Interestingly, the ANN and SVR MLTs trade places with each other since they either present the best (first and second place) or worst (fifth place) results. Consequently, the ANN MLT deals better with the known (scenario 1) and unknown (scenario 2) but similar test data, while the SVR MLT deals better with the unknown (scenarios 3 and 4) test data of experiment 1.

3.1.1 | Seasonal trends of the MLTs trained by the spring dataset of PV system A

The seasonal trends of the MLTs trained by the spring dataset of PV system A are visually illustrated in this section (Figure 4). This illustration provides a better understanding of the prediction value variation of the different MLTs throughout the different seasons of the year. Figure 4 provides visual information on how the predicted values react in the different months throughout the year compared with the measured data.

The bold blue line in all graphs of Figure 4 represents the measured data (targets) of the solar production output of the PV systems. The MLT estimates are very close to the measured data in the spring season since they were trained by the spring dataset. However, only in the month of April are the prediction values closely correlated to the measured values.

In the summer season, the MLTs tend to overestimate the solar production in all 3 months.

In the autumn months, there are outliers created by the ANN MLT that was also identified in Table 3. The ANN MLT does not know how

TABLE 4 Ranking score of the machine learning techniques trained by the spring season in each scenario

	Experiment 1 — Seasonal Comparison				Experiment 2 — Location Comparison		
Rank	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6	Scenario 7
	Spring	Summer	Autumn	Winter	PV System A	PV System B	PV System C
1º	Regression tree	ANN	SVR	SVR	Regression tree	Regression tree	Regression tree
2º	ANN	Gaussian process	MG genetic programming	Regression tree	ANN	ANN	ANN
3º	MG genetic programming	MG genetic programming	Gaussian process	Gaussian process	Gaussian process	Gaussian process	Gaussian process
4º	Gaussian Process	Regression tree	Regression tree	MG genetic programming	MG genetic programming	MG genetic programming	MG genetic programming
5º	SVR	SVR	ANN	ANN	SVR	SVR	SVR
	Known data		Unknown data		Known data	Unknown data	

Abbreviations: ANN, artificial neural network; MG, multigene; PV, photovoltaic; SVR, support vector machine for regression.

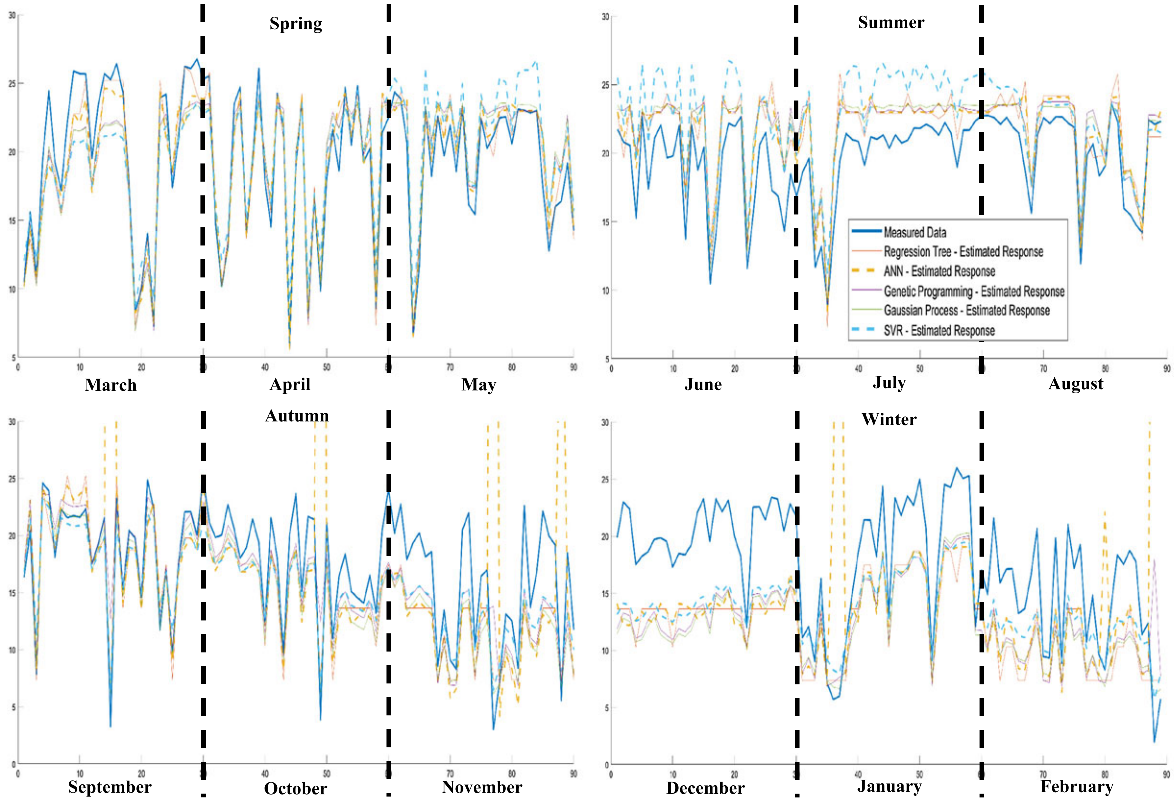

FIGURE 4 PV system A spring, summer, autumn, and winter test dataset of 5 machine learning technique prediction values compared with the measured solar AC energy values [Colour figure can be viewed at wileyonlinelibrary.com]

to react when presented with very low energy outputs (unknown test data), since these were not introduced in the spring database (known test data). It is curious to see that all the other MLTs know how to react when dealing with very low energy values or in other words unseen/unknown data. Here, in the autumn months, there is a tendency for the MLT to underestimate the solar production; however, September presents a close correlation between the measured data and the MLT estimates (Figure 5). This indicates that there could be a similar trend between the months of April and September, which needs further research (Figure 4).

Regarding the winter months, the predicted values are highly underestimated, particularly in the month of December (Figure 4).

Figure 5 is a resume of the correlated, overestimated, and underestimated predicted values in relation to the measured data presented in Figure 4. The closely correlated prediction values take place in the month of April and late September. The overestimated prediction values take place from the month of May right through to the mid-late September, while the prediction values are underestimated from the month of October right through to the month of April. The switch between the underestimated results with the overestimated results takes place in the month of April, while the overestimated results switch with the underestimated results in mid-late September. Therefore, the trend of the prediction values of the spring and summer seasons is overestimated in relation to the measured data, while the prediction values of the autumn and winter seasons are underestimated. Overall, the predicted values do not closely correlate with the measured data in all the months of the year, except for the months of April and mid-late September (Figure 5).

3.2 | Experiments 1 and 2 (seasonal comparison and location comparison) using the PV system A summer, autumn, and winter datasets for training

The results presented in the previous subsection served as motivation to pursue further testing by training the same MLTs with other seasons of the year to understand what would happen to the error analysis results of experiments 1 and 2 if the MLT was trained

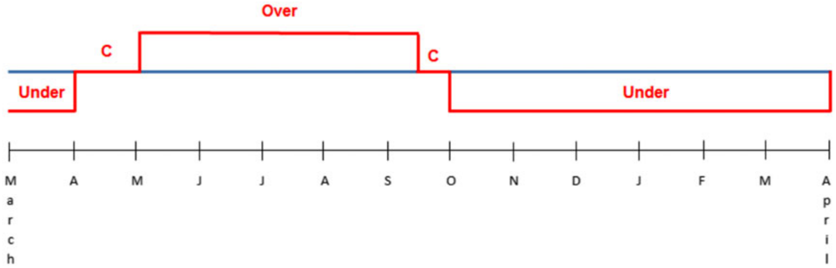

FIGURE 5 The yearly trend of the correlated, overestimated, and underestimated predicted values of the machine learning techniques. ANN, artificial neural network; SVR, support vector machine for regression [Colour figure can be viewed at wileyonlinelibrary.com]

by the summer, autumn, and winter datasets of PV system A (Tables 5, 6, and 7) instead of the spring dataset discussed in Section 3.1. The training of other seasons provides information about how each MLT reacts to each of the seasons of the year, and the winter training season is clearly the season of the year that presents the worst results. However, as explained in the historical data (Section 2.1), not all datasets were available for location comparison since PV system C did not have any data available for summer, autumn, nor winter. Photovoltaic system B did not have available historic data for the summer season.

The prediction value performance of all the MLT prediction models depends on the season by which it was trained with. From the results in Tables 5, 6, and 7, it is clear to see how the ANN and SVR MLTs work best when training the prediction models with the

TABLE 5 Error analysis results of experiments 1 and 2 by training with the summer dataset

			Experiment 1—Seasonal Comparison				Experiment 2—Location Comparison		
Tribe	MLT	Error Analysis	Scenario 1 Spring	Scenario 2 Summer	Scenario 3 Autumn	Scenario 4 Winter	Scenario 5 PV System A	Scenario 6 PV System B	Scenario 7 PV System C
1	RT	RMSE, kWh NRMSE, %	3.4829 12.0098	0.8880 3.0619	5.5039 18.9790	7.2667 25.0577	1.0870 3.7483	…	…
2	ANN	RMSE, kWh NRMSE, %	3.2738 11.2890	0.8868 3.0578	5.3700 18.5171	7.3322 25.2833	1.0573 3.6459	…	…
3	MG genetic programming	RMSE, kWh NRMSE, %	3.5071 12.0933	1.1273 3.8873	6.4151 22.1211	17.7196 61.1020	1.0358 3.5716	…	…
4	Gaussian process	RMSE, kWh NRMSE, %	3.2575 11.2329	1.1658 4.0201	5.1770 17.8516	7.3878 25.4750	1.0061 3.4693	…	…
5	SVR	RMSE, kWh NRMSE, %	3.0352 10.4661	1.5234 5.2531	4.1476 14.3022	6.1295 21.1360	0.8103 2.7940	…	…

Abbreviations: ANN, artificial neural network; MG, multigene; MLT, machine learning technique; NRMSE, normalized root mean square error; PV, photovoltaic; RMSE, root mean square error; RT, regression tree; SVR, support vector machine for regression.

TABLE 6 Error analysis results of experiments 1 and 2 by training with the autumn dataset

			Experiment 1—Seasonal Comparison				Experiment 2—Location Comparison		
Tribe	MLT	Error Analysis	Scenario 1 Spring	Scenario 2 Summer	Scenario 3 Autumn	Scenario 4 Winter	Scenario 5 PV System A	Scenario 6 PV System B	Scenario 7 PV System C
1	RT	RMSE, kWh NRMSE, %	3.0907 10.6575	3.9930 13.7691	1.6342 5.6351	3.1660 10.9171	1.7776 6.1296	2.5160 8.6758	…
2	ANN	RMSE, kWh NRMSE, %	3.7077 12.7852	5.2992 18.2731	1.6400 5.6552	3.0800 10.6208	1.6695 5.7571	1.9922 6.8697	…
3	MG genetic programming	RMSE, kWH NRMSE, %	3.1355 10.8121	3.7960 13.0896	1.9438 6.7029	3.1541 10.8760	4.0811 14.0729	5.2796 18.2055	…
4	Gaussian process	RMSE, kWh NRMSE, %	2.8362 9.7800	3.2236 11.1160	1.9237 6.6334	3.0970 10.6793	2.0228 6.9752	1.7145 5.9121	…
5	SVR	RMSE, kWh NRMSE, %	4.3235 14.9085	7.1960 24.8136	2.2995 7.9293	3.6439 12.5651	2.6724 9.2152	1.7209 5.9342	…

Abbreviations: ANN, artificial neural network; MG, multigene; MLT, machine learning technique; NRMSE, normalized root mean square error; PV, photovoltaic; RMSE, root mean square error; RT, regression tree; SVR, support vector machine for regression.

TABLE 7 Error analysis results of experiments 1 and 2 by training with the winter dataset

			Experiment 1—Seasonal Comparison				Experiment 2—Location Comparison		
Tribe	MLT	Error Analysis	Scenario 1 Spring	Scenario 2 Summer	Scenario 3 Autumn	Scenario 4 Winter	Scenario 5 PV System A	Scenario 6 PV System B	Scenario 7 PV System C
1	RT	RMSE, kWh NRMSE, %	4.6302 15.9663	5.6297 19.4127	3.3102 11.4144	0.9284 3.2015	1.9695 6.7915	4.0044 13.8084	…
2	ANN	RMSE, kWh NRMSE, %	4.6079 15.8893	5.6035 19.3223	3.2647 11.2578	1.3928 4.8028	1.9553 6.7423	3.9248 13.5336	…
3	MG genetic programming	RMSE, kWh NRMSE, %	30.7015 105.8671	338.7078 1.1680e + 03	17.0545 58.8087	1.4986 5.1675	266.4703 918.8631	283.1825 976.4913	…
4	Gaussian process	RMSE, kWh NRMSE, %	4.6017 15.8680	5.5285 19.0637	3.3387 11.5129	1.5444 5.3254	1.6565 5.7120	3.9536 13.6332	…
5	SVR	RMSE, kWh NRMSE, %	9.5057 32.7783	13.8199 47.6548	4.6513 16.0391	2.0535 7.0810	1.5874 5.4739	3.4295 11.8258	…

Abbreviations: ANN, artificial neural network; MG, multigene; MLT, machine learning technique; NRMSE, normalized root mean square error; PV, photovoltaic; RMSE, root mean square error; RT, regression tree; SVR, support vector machine for regression.

summer training dataset just as seen in Table 5 and when using the spring training dataset (Section 3.1), while the ANN and the Gaussian process MLTs work best when training the prediction model with the autumn and winter training datasets. This indicates that spring and summer seasons have similar trends and work best with ANN and SVR MLTs, while the autumn and winter seasons have similar trends and work best with ANN and Gaussian process MLTs. Interestingly, the ANN MLT only works best with the testing season that is most similar to the training season. The spring and summer seasons are considered as the "similar" seasons in this work as well as the autumn and winter seasons. An example of this can be verified in Table 6 when the training season is autumn; the ANN MLT has the best performance values when tested with the winter season (scenario 4).

When comparing all the MLT results of all the training seasons (Tables 3, 5, 6, and 7), only the regression tree MLT provides acceptable results (RMSE ~ 6 kWh) in all the seasons of the year and in PV system installation locations. Therefore, the regression tree MLT can be used as the one-size-fits-all (generic) prediction model of the PV performance analyser proposed in this work for rooftop PV systems of up to 4.14 kW installed power. However, the best results are provided by the ANN, Gaussian process, and SVR MLTs and should be used accordingly as explained in the paragraph above.

Of all the training seasons (Tables 3, 5, 6, and 7), the training season that presents the best MLT results all year round is the autumn months (Table 6), with the exception of the SVR MLT in the summer months (scenario 2), since the RMSE value is much higher than the accepted value of 6 kWh assumed for this work. Curiously, the autumn months include the month of September that was identified earlier as presenting the prediction values very closely correlated to the measured data.

4 | CONCLUSION

The ideal ML PV performance analyser would have a prediction model that is not influenced by the different seasons of the year nor the distance between the PV system installation and the weather station. The challenge of this work was to determine whether the 3-month (or 89 samples) daily historical data of a given rooftop PV system installation could be used for another in a different location and/or for the same PV system installation but in a different season of the year and attempt to accomplish the one-model-fits-all approach.

Five MLTs were tested in 2 experiments (seasonal comparison and location comparison) with a total of 7 scenarios, as explained in Section 2.

In conclusion, the aim of this work is achieved, since it is possible to implement a one-size-fits-all prediction model for rooftop PV systems of up to 4.14 kW of DC-installed power that does not need more than 3 months of daily historical data and that is not influenced by the seasons of the year nor the PV system installation location. This is possible by training the regression tree MLT with at least 3 months of daily historical data (solar irradiation input and solar AC energy output) from any season of the year. However, results show that the regression tree MLT provides acceptable results to be used in all locations and all seasons of the year, while the SVR is best for spring and summer training dataset months, and the Gaussian process is best for the autumn and winter training dataset months.

The ANN also provides good results but only for the season that is similar to the trained season. The spring and summer seasons are considered "similar seasons," as are the autumn and winter seasons. Therefore, the ANN MLT provides good results for the summer tested season when trained with the spring season and vice versa. Thus, the ANN also provides good results for the winter tested season when trained with the autumn season and vice versa.

Regarding the location of the PV system, the prediction values are very similar for the PV systems of the same installed power in different locations; however, if the MLT is trained with a database of a 4.14 kW PV system, the prediction values will not be as accurate for PV systems of a lower installed power (PV system C) in different locations. Even though PV system C is 4.1 km away from the weather station, all the MLTs provide accurate prediction values.

The overall trend of the prediction values of the spring and summer seasons is overestimated in relation to the measured data, while the prediction values of the autumn and winter seasons are underestimated. In general, the predicted values do not closely correlate with the measured data in all the months of the year, except for the months of April and mid-late September.

Overall, the training season that presents the best results all year round in all MLTs is the autumn months. Curiously, the autumn months include the month of September that was earlier identified as presenting the prediction values very closely correlated to the measured data.

It was not possible to find any work that allows for a direct comparison of the results obtained in this paper since some of the characteristics include

- hourly-based predictions instead of daily based,
- the weather stations are always onsite,
- different PV system–installed powers are used,
- other inputs besides solar irradiation are used,
- no details about the parameters used for MLT techniques are supplied, and
- the data are different and with different lengths.

Nevertheless, the following general comparisons can be made: Zamo et al[6] present better SVM results compared with regression tree results. In this work, regression tree results are better than the SVR results. Hossain et al[7] present better ANN results compared with SVR. In this work, ANN is better than SVR only when using similar or known testing data (same season) and the other way around when using unknown data for testing. Prokop et al[11] present better genetic programming results compared with ANN and SVR. In this work, the genetic programming results are not better than the ANN and SVR results. Pedro and Coimbra[2] present better ANN results compared with genetic programming results, which is in accordance with what is presented in this work. Huang et al[12] present better Gaussian process results compared with ANN and SVR. In this work, Gaussian process is better than ANN when using unknown data for testing and the other way around when using similar and known data.

Regarding the SVR performance, the Gaussian process results are better than SVR results when dealing with similar and known testing data and the other way around when using unknown data for testing. Yang et al[13] present better SVR results compared with ANN. In this work, the SVR results are better than the ANN results when dealing with unknown data. ANN presents better results compared with SVR when dealing with similar and known data for testing.

This ML PV system performance analyser can be used in real-world applications. This analyser can be used by the PV system installation companies to offer new services to customers, since it has been proven to work with rooftop PV system in any season of the year and in any location of the Madeira Island. The services would include PV system performance analysis for PV systems that have at least 3 months of weather and solar AC energy historical data, for PV systems that do not have any available historical data and for PV systems that have only the solar AC energy historical data but do not have the weather historical data. This new service can be used in real time, which allows for real-time problem solving by implementing an alarm that would go off in the event of poor PV output performance.

The limitations related to this work include the fact that the MLTs were trained and tested by using the parameters of the toolbox default settings, in which validation and cross-validation were not performed unless automatically performed by the fitting command of the toolboxes that were used in this work. The other limitation would be related to the available number of samples that were provided to be used for training and testing the MLTs. The maximum total of daily samples that were able to be provided was 361 daily samples instead of the 365 daily samples expected to analyse a whole year.

Future work would include further exploring the more recently used MLTs such as the random forests (tribe 1), extreme ML (tribe 2), and *k*-nearest neighbour (tribe 5). Future work would also include further research on the trend that takes place in the months of April and September.

ACKNOWLEDGEMENTS

The authors acknowledge the PV system installation company Factor Energia for providing the historical data from the 3 PV systems located in the island of Madeira, Portugal, and the meteorological institution IPMA (IPMA—Instituto Português do Mar e da Atmosfera) for providing the daily solar irradiation historical data. The PhD grant financial support from ARDITI (Agência Regional para o Desenvolvimento da Investigação, Tecnologia e Inovação) under the scope of the project M1420-09-5369-FSE-000001, is gratefully appreciated. Acknowledgments to the Portuguese Science and Technology Foundation (FCT) for their support through Projeto Estratégico LA 9 - UID/EEA/50009/2013. This work was also developed under the Instituto de Telecomunicações and it was supported in part by the Portuguese Science and Technology Foundation (FCT) projects: UID/EEA/50008/2013. This support is gratefully acknowledged.

ORCID

Sandy Rodrigues http://orcid.org/0000-0003-1989-4852
Helena Geirinhas Ramos http://orcid.org/0000-0002-4931-7960
Fernando Morgado-Dias http://orcid.org/0000-0001-7334-3993

REFERENCES

1. Raza MQ, Nadarajah M, Ekanayake C. On recent advances in PV output power forecast. *Sol Energy*. 2016;136(136):125-144. https://doi.org/10.1016/j.solener.2016.06.073
2. Pedro HTC, Coimbra CFM. Assessment of forecasting techniques for solar power production with no exogenous inputs. *Sol Energy*. 2012;86(7):2017-2028. https://doi.org/10.1016/j.solener.2012.04.004
3. Inman RH, Pedro HTC, Coimbra CFM. Solar forecasting methods for renewable energy integration. *Prog Energy Combust Sci*. 2013;39(6):535-576. https://doi.org/10.1016/j.pecs.2013.06.002
4. Antonanzas J, Osorio N, Escobar R, Urraca R, Martinez-de-pison FJ, Antonanzas-torres F. Review of photovoltaic power forecasting. *Sol Energy*. 2016;136:78-111. https://doi.org/10.1016/j.solener.2016.06.069
5. Domingos P. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books; 2015.
6. Zamo M, Mestre O, Arbogast P, Pannekoucke O. A benchmark of statistical regression methods for short-term forecasting of photovoltaic electricity production, part I: deterministic forecast of hourly production. *Sol Energy*. 2014;105:792-803. https://doi.org/10.1016/j.solener.2013.12.006
7. Hossain R, Maung A, Oo T. Ali a BMS. Hybrid prediction method for solar power using different computational intelligence algorithms. *Smart Grid Renew Energy*. 2013;4(February):76-87.
8. Searson D, Leahy D, Willis M. GPTIPS: An open Sources Genetic programming toolbox for multigene symbolic regression. *Proceedings of the International of the MultiConference of Engineers and* Computer *Scientists*. 2010;I:17–20. http://citeseerx. ist. psu.edu/viewdoc/download? doi=10.1.1.177.522&rep=rep1&type=pdf.
9. Aliesfahani S, Shahbazian M. Maximum Power Point Tracking for Solar Photovoltaic System Using Genetic Programming Toolbox for Identification of Physical System. *Int. J Autom Control*. 2015;3(1):25-28. https://doi.org/10.12691/automation-3-1-4
10. Russo M, Leotta G, Pugliatti PM, Gigliucci G. Genetic programming for photovoltaic plant output forecasting. *Sol Energy*. 2014;105:264-273. https://doi.org/10.1016/j.solener.2014.02.021
11. Prokop L, Misák S, Snásel V, Platos J, Kromer P. Supervised learning of photovoltaic power plant output prediction models. *Neural Netw World*. 2013;23(4):321-338.
12. Huang C, Bensoussan A, Edesess M, Tsui KL. Improvement in artificial neural network-based estimation of grid connected photovoltaic power output. *Renew Energy*. 2016;97:838-848. https://doi.org/10.1016/j.renene.2016.06.043
13. Yang H-T, Huang C, Huang Y-C, Pai Y-S. A hybrid method for one-day ahead hourly forecasting of PV power output. *Trans Sustainable Energy*. 2014;5(3):526-531.
14. Junior JG da SF, Oozeki T, Takashima T, Koshimizu G, Uchida Y, Ogimoto K. Use of support vector regression and numerically predicted cloudiness to forecast power output of a photovoltaic power plant in Kitakyushu, Japan. *Prog Photovolt: Res Appl*. 2012;20(7):874-882. http://dx. https://doi.org/10.1002/pip.1152
15. Murphy K. *Machine Learning. A Probabilistic perspective*. MIT Press; 2012. doi: https://doi.org/10.1038/217994a0., 217, 5133, 994
16. Wan C, Zhao J, Member S, Song Y. Photovoltaic and solar power forecasting for smart grid energy management. *Int J Power Energy Syst*. 2015;1(4):38-46. https://doi.org/10.17775/CSEEJPES.2015.00046
17. Vladislavleva EJ, Smits GF, den Hertog D. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *IEEE Trans Evol Comput*. 2009;13(2):333-349. https://doi.org/10.1109/TEVC.2008.926486
18. Nguyen NT, Kowalczyk R. Hybrid single node genetic programming for symbolic regression. *Trans Comput Collective Intell XXII*. 2016. https://doi.org/10.1007/978-3-662-53525-7_4

19. De Leone R, Pietrini M, Giovannelli A. Photovoltaic energy production forecast using support vector regression. *Neural Comput & Applic*. 2015;26(8):1955-1962. https://doi.org/10.1007/s00521-015-1842-y

20. Chen F, Duic N, Manuel Alves L, da Graça Carvalho M. Renewislands-renewable energy solutions for islands. *Renew Sustain Energy Rev*. 2007;11(8):1888-1902. https://doi.org/10.1016/j.rser.2005.12.009

How to cite this article: Rodrigues S, Ramos HG, Morgado-Dias F. Machine learning PV system performance analyser. *Prog Photovolt Res Appl*. 2018;26:675–687. https://doi.org/10.1002/pip.3060