electronics

MDPI

Article

Machine Learning Based Approaches for Modeling the Output Power of Photovoltaic Array in Real Outdoor Conditions

Malvoni Maria [1,*] and Chaibi Yassine [2]

[1] School of Electrical and Computer Engineering, National Technical University of Athens, 15780 Athens, Greece

[2] SmartiLab Laboratory, Moroccan School of Engineering Sciences (EMSI), Rabat 10090, Morocco; chaibi.yassine@gmail.com

* Correspondence: maria.malvoni@gmail.com

Received: 27 December 2019; Accepted: 28 January 2020; Published: 12 February 2020

Abstract: It is important to investigate the long-term performances of an accurate modeling of photovoltaic (PV) systems, especially in the prediction of output power, with single and double diode models as the configurations mainly applied for this purpose. However, the use of one configuration to model PV panel limits the accuracy of its predicted performances. This paper proposes a new hybrid approach based on classification algorithms in the machine learning framework that combines both single and double models in accordance with the climatic condition in order to predict the output PV power with higher accuracy. Classification trees, k-nearest neighbor, discriminant analysis, Naïve Bayes, support vector machines (SVMs), and classification ensembles algorithms are investigated to estimate the PV power under different conditions of the Mediterranean climate. The examined classification algorithms demonstrate that the double diode model seems more relevant for low and medium levels of solar irradiance and temperature. Accuracy between 86% and 87.5% demonstrates the high potential of the classification techniques in the PV power predicting. The normalized mean absolute error up to 1.5% ensures errors less than those obtained from both single-diode and double-diode equivalent-circuit models with a reduction up to 0.15%. The proposed hybrid approach using machine learning (ML) algorithms could be a key solution for photovoltaic and industrial software to predict more accurate performances.

Keywords: PV modules modeling; equivalent-circuit models; prediction of performances; machine learning; classification algorithms

1. Introduction

Due to the high increase of petroleum prices and imposed politics on industrial countries to reduce CO_2 levels, the use of renewable sources to produce energy has become an obligation. Accordingly, different solutions are used to cover energy needs while respecting clean and eco-friendly requirements [1]. Generally, wind, hydro, and solar sources of energy show an appropriate solution ensuring green electricity for diverse industrial and domestic applications. In particular, photovoltaic systems display a good balance between investment cost and performance [2].

The modeling task is a substantial procedure to analyze the electrical performances of the photovoltaic (PV) cell/module/array. Equivalent-circuit models are mainly implemented to predict the long-term potential of the photovoltaic device. Single-diode model (SDMs) and double-diode models (DDMs) represent the most used configurations [3]. Furthermore, the single-diode model is less complicated compared to the double-diode configuration; this is because of the limited number of parameters needed. The SDM requires five while DDM requires seven [4]. To achieve a high

level of accuracy, several modeling processes have been developed. Either based on analytical or metaheuristic methods, the predicted performances of PV modules are still limited in accuracy due to the use of only one equivalent-circuit configuration (SDM or DDM) to electrically model the PV cell behavior. This limitation is demonstrated under the variation of solar irradiance and temperature [4,5]. Nevertheless, several studies discuss the influence of climatic conditions. Ishaque et al. claimed that the single-diode model exhibits modest performances compared to the DDM under low-irradiance variation; this is demonstrated by comparing generated current voltage (I–V) curves and adopting both models with experimental data [4,6]. Also, Ishaque et al. proved that the DDM is more accurate under partial shading conditions. Et-torabi et al. [7] adopted iterative methods and proved that SDM can reach high accuracy for high irradiance and low-temperature levels. In addition, Villalva et al. demonstrated that the SDM is more simple and pertinent for all variations [8]. Chaibi et al. proposed a combination of SDM and DDM according to climatic changes in order to improve accuracy and get high prediction performance [5].

In the last years, machine learning (ML) techniques have been widely adopted to estimate the long-term performances, and predict the output power, of PV plants [9,10]. Theocharides et al. [11] assessed the PV generation using three ML techniques, such as artificial neural network (ANN), support vector machine (SVM), and regression tree. The results proved that the ANN presents high accuracy compared to other examined algorithms. Several approaches based on SVMs have been proposed to forecast the PV output power. Shi et al. [12] predicted the PV power for a 20 kW grid-connected PV installation located in China according to weather classifications (rainy, foggy, cloudy, and sunny). An SVM used real outdoor data of solar irradiance and ambient temperature to forecast PV power generation 24 h ahead [13]. In another study [14], an SVM combined with three-dimensional wavelet transform predicted the PV power of distributed PV plants using historical time series data. Furthermore, Wang et al. proposed a short-term power prediction using the gradient boost decision tree (GBDT) algorithm adopting historical weather data together with PV output powers [15].

The main objective of this work is to present a machine learning based approach that combines both single-diode and double-diode equivalent-circuit models according to climatic conditions in order to predict the output PV power with high accuracy. Therefore, the performances of single-diode and double-diode models are investigated under different levels of solar irradiance and ambient temperature. A comparison of the PV power estimated by using two equivalent-circuit models with real recorded data is performed in this study. Later, the effectiveness of machine learning techniques to select models according to corresponding accuracy is assessed. Therefore, an approach based on ML classification algorithms is proposed to prioritize single-diode and double-diode equivalent-circuit models for a given solar irradiance and temperature. Six classification algorithms, such as classification trees (coarse tree), k-nearest neighbors (cubic), discriminant analysis (quadratic), Naïve Bayes (kernel), support vector machines (Gaussian), and classification ensembles (boosted trees-AdaBoost) are implemented to categorize SDMs and DDMs into specific "classes" in order to predict the output PV power using real outdoor operating conditions. The accuracy of all classifiers is evaluated under real outdoor conditions of a 114 kW grid-connected PV plant located in Southern Italy. The novelty of this work is to evaluate the effectiveness of the proposed approach; the classification algorithms are able to select between single- and double-diode equivalent-circuit models according to different levels of solar irradiance and temperature, in order to ensure high accuracy in the output power predicting.

The present paper is arranged as follows: PV modeling and the classification algorithms are presented in Section 2. Then, the proposed hybrid approach is introduced in Section 3. The results and discussion are presented in Section 4. Finally, we give some conclusions.

2. Materials and Methods

2.1. PV Cell Modeling

To evaluate the electrical behavior of the PV device (cell), plenty of equivalent-circuit models are proposed. In the literature, there are two configurations widely used. Namely, the single- and the double-diode model.

2.1.1. Single-diode Model

This model is developed by adding a series and a shunt resistance to the ideal model to represent the losses of the module [16]. By applying the Kirchhoff laws on the scheme in Figure 1, the output current of the PV panel is given by the following Equation [16]:

$$I = I_{ph} - I_{os}\{\exp[A(V + IR_s) - 1]\} - \frac{V + R_s I}{R_{sh}}, \quad (1)$$

where, I_{ph} is the light-generated current, I_{os} is the diode saturation current, and R_s and R_{sh} are respectively the series and the shunt resistance. The value of A depends on the thermal voltage $V_t = \frac{kT}{q}$ and is expressed as follows:

$$A = \frac{1}{V_t N_{\text{cell}}}, \quad (2)$$

γ is the ideality factor of the diode, and N_{cell} represents the number of cells that compose the PV module.

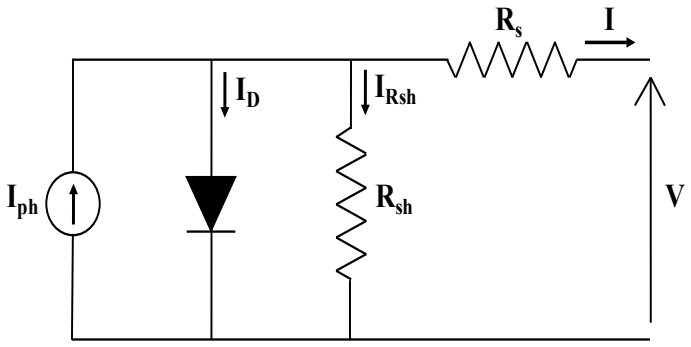


Figure 1. Single-diode equivalent-circuit model.

2.1.2. Double-Diode Model

As another configuration to model the PV cell, the double-diode is an improvement of what was claimed about the single-diode accuracy [17]. Recently, the DDM has become a primary solution for several authors due to its improvements especially at low-irradiances [4,18]. This model considers two diodes in parallel with the current source as shown in Figure 2, and the output current is given by Equation (3):

$$I = I_{ph} - I_{os1}\{\exp[A_1(V + IR_s) - 1]\} - I_{os2}\{\exp[A_2(V + IR_s) - 1]\} - \frac{V + R_s I}{R_{sh}}, \quad (3)$$

where, A_1 and A_2 depend respectively on the values of the ideality factor of each diode (γ_1 and γ_2) and cell temperature. In addition, I_{os1} and I_{os2} are the saturation current of each diode separately.

The second step of PV cell modeling is to evaluate the unknown parameters of Equations (1) and (3). The number of parameters to determine depends on the used equivalent-circuit model (e.g., five parameters for the single-diode model ($I_{ph}, I_{os}, R_s, R_{sh},$), and seven parameters for the double-diode model ($I_{ph}, I_{os1}, I_{os2}, R_s, R_{sh}$,$_1$, $_2$, γ_1, γ_2)) [5].

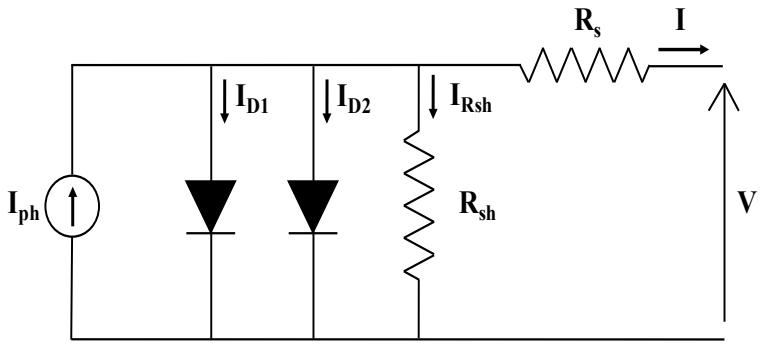


Figure 2. Double-diode equivalent-circuit model.

To estimate these unknown parameters, many techniques are proposed using different approaches. Whether it is a numerical, analytical, or metaheuristic based method [19–21], most of these approaches use I–V experimental data or datasheet information to simplify the estimation of the parameters. For this reason, the photo-generated current is generally calculated using Equation (4) [22]:

$$I_{ph} = [I_{sc} + K_i(T - 298.15)]\frac{\lambda}{1000}\,, \tag{4}$$

where, I_{sc} is the short-circuit current, K_i is the temperature coefficient related to I_{sc}, and λ and T are respectively the solar irradiance level and cell temperature.

2.1.3. Maximum Power Point

The maximum power point (MPP) represents the optimal operating of the PV module regardless of climate variations [23]. Mathematically at this point, the derivative of PV power with respect to PV voltage is expressed as follows:

$$\frac{\partial P}{\partial V} = I + V\frac{\partial I}{\partial V} = 0\,, \tag{5}$$

In order to compute the maximum power point for both adopted equivalent-circuit models, the current expressions in Equations (1) and (2) are calculated using Equation (5). Then, the optimal currents of the single-diode and double-diode models are respectively expressed by the following equations:

$$I_{\mathrm{MPP}} = (V_{\mathrm{MPP}} - R_s I_{\mathrm{MPP}})\left\{AI_{os}\exp[A(V_{\mathrm{MPP}} + R_s I_{\mathrm{MPP}})] + \frac{1}{R_{sh}}\right\}, \tag{6}$$

$$I_{\mathrm{MPP}} = (V_{\mathrm{MPP}} - R_s I_{\mathrm{MPP}})\{I_{os1}A_1\exp[A_1(V_{\mathrm{MPP}} + R_s I_{\mathrm{MPP}})] + I_{os2}A_2\exp[A_2(V_{\mathrm{MPP}} + R_s I_{\mathrm{MPP}})] + \tfrac{1}{R_{sh}}\Big\}, \tag{7}$$

where, V_{MPP} and I_{MPP} are respectively the PV output voltage and current at the maximum power point. The V_{MPP} and I_{MPP} coordinates are then used to estimate the output power $P_{\mathrm{MPP}} = V_{\mathrm{MPP}} * I_{\mathrm{MPP}}$ for both adopted equivalent-circuit models.

In order to obtain the maximum power from a PV plant, a maximum power point tracking (MPPT) algorithm controls and adjusts the operating voltage to reach the maximum output power. MPP losses occur when the MPPT is not able to find the MPP rapidly. Typical MPP loss values are lower than 0.5%. Furthermore, the operating voltage of the PV array depends on the DC cable length, cross-section, and temperature that can lead to current and power losses, and the connection of modules in series can cause the mismatching between the I–V characteristics of the module (mismatch losses). In the present work, when modeling PV, estimation of the MPPT and DC losses are not considered.

2.2. *Adopted Machine Learning Algorithm*

Supervised machine learning (ML) techniques are able to predict the response for a given measurement set of the predictor variables on the base of a model built on a known observation set noted as the training dataset.

Classification algorithms are a type of supervised ML technique in which an algorithm learns to separate the data into specific "classes" in order to predict categorical responses [24]. The most popular classification algorithms include [25]:

- classification trees (coarse tree)
- k-nearest neighbors (cubic)
- discriminant analysis (quadratic)
- Naïve Bayes (kernel)
- support vector machines (Gaussian)
- classification ensembles (boosted trees-AdaBoost)

2.2.1. Classification Trees

The classification tree technique also noted as a decision tree is one of the common approaches applied in data mining to predict the class response for a given observation by using specific predictor variables [26]. The classification tree learning maps the observations as a tree structure to model its target value [27]. In the tree, each node represents a feature and each path corresponds to what is associated. The decision tree learner can identify each node that classifies the best value of the feature within the dataset according to some criterion. The terminal nodes are marked according to the classes into which the instances are to be classified. During the testing, the value of the instance is compared with the value labeled at each path (branch). If the value at the node matches the value of the node then the classification will continue through the path until it meets the terminal node as shown in Figure 3 [28]. The terminal leaf nodes are shown as orange and yellow squares according to the classes. In each tree, the instance is shown in a blue path. In Figure 3a–c the tree predicts the yellow class, unlike in Figure 3d the instance is in the orange class, so the classifier will assign it to the yellow class by a 3 to 1 majority voting.

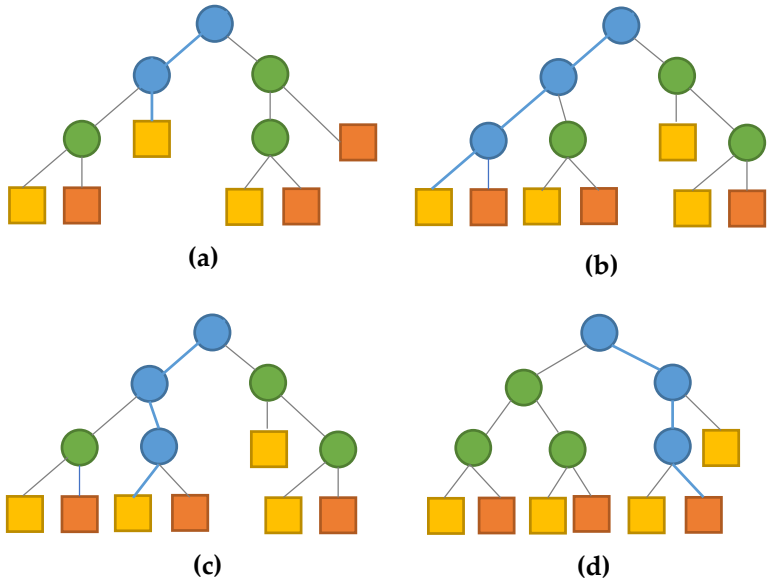


Figure 3. Graphical representation of classification trees. (**a**) yellow class prediction by the first path, (**b**) yellow class prediction by the second path, (**c**) yellow class prediction by the third way, (**d**) orange class prediction.

2.2.2. k-Nearest Neighbors

The k-nearest neighbors algorithm (kNN) is a ML method applied for classification where an instance represents a point in a d-dimensional space and each dimension corresponds to one of the d features. So, the instances which present the same properties would be close to each other in the d-dimensional space [29]. In order to predict the class, the kNN algorithm finds k nearest instances by computing the distance between them. The predicted class is represented by the minimum distance among instances. The Euclidean distance is usually applied as the distance metric [30,31]. The k-nearest neighbor classification algorithm is listed in Appendix A. Figure 4 shows the kNN classification concept. The green instance will be assigned to the blue class for $k = 1$. For $k = 3$ it will be classified as the blue class by a 2 to 1 majority and finally, it will be assigned to the orange class by 3 to 2 cases for $k = 5$. Therefore, the k-nearest-neighbors classifier assigns to a test sample the majority class of its k-nearest training samples.

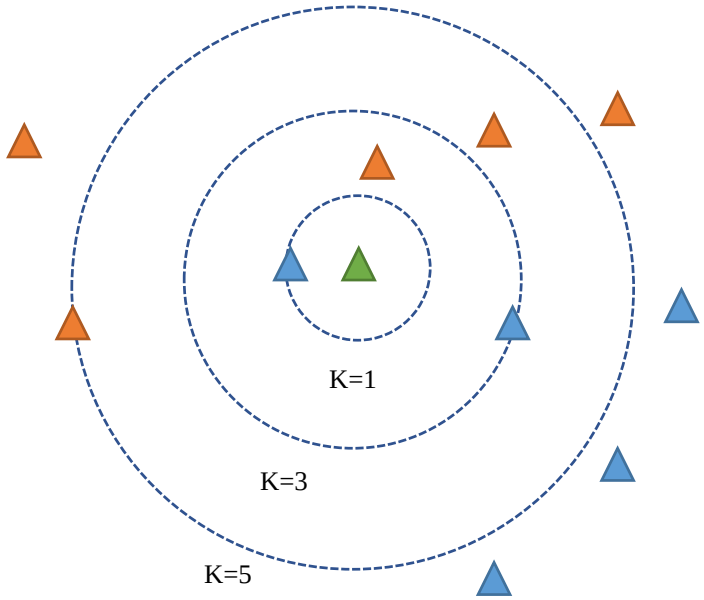


Figure 4. Graphic depiction of a k-nearest neighbors (kNN) classification model.

2.2.3. Discriminant Analysis

Discriminant analysis (DA) finds a predictive equation based on independent variables to classify the instances into classes [32]. Discriminant analysis is very similar to regression analysis, where the dependent variables become the independent variables in the discriminant analysis. The mathematical formulation is presented in Appendix A. It can be considered as dimensionality reduction technique, reducing the sample space into a smaller dimension while retaining as much information as possible. Discriminant analysis can be distinguished into two categories in according to the boundary between the classes: linear discriminant analysis (LDA) or quadratic discriminant analysis (QDA) [33]. LDA adopts the coordinate axes to transform data by reducing the two-dimensional space into a one-dimensional space using a linear boundary. The QDA can be considered as an extension to the LDA. It classifies two or more classes by a quadratic model as a surface.

2.2.4. Naïve Bayes

Naïve Bayes classification algorithm is one of the most popular statistical learning methods based on the Bayes theorem related to the conditional probability, predicting the most probable class.

Given an instance and its occurring probability $P(d)$, the Bayes theorem says:

$$P(c_i|d) = \frac{P(d|c_i) * P(c_i)}{P(d)}, \tag{8}$$

where:

$P(c_i|d)$ is the probability of the instance d being in the class c_i;
$P(d|c_i)$ is the probability of observing d in a domain where c holds;
$P(c_i)$ is the prior probability of c_i;

The Naïve Bayes classifier computes the probability of each instance for all classes in c and selects the class c_i with the highest probability (Figure A1). Generally, the features are assumed to have a Gaussian probability distribution. When the features do not follow a Gaussian distribution, the kernel density method [34] is applied to estimate the probability distribution. More details are provided in Appendix A.

2.2.5. Support Vector Machines (SVM)

Support vector machines are supervised learning models able to analyze data and learn a classifier [35]. An SVM finds the optimal separating hyperplane as a decision surface to separate the data in different classes. First, the SVM method transforms predictors to high-dimensional feature space and successively solves a quadratic optimization problem to find an optimal hyperplane in order to classify the transformed features into classes [36].

Figure 5 [37] shows the optimal separating hyperplane in two dimensions. The yellow plane divides the support vectors into two classes (red squares and blue dots).

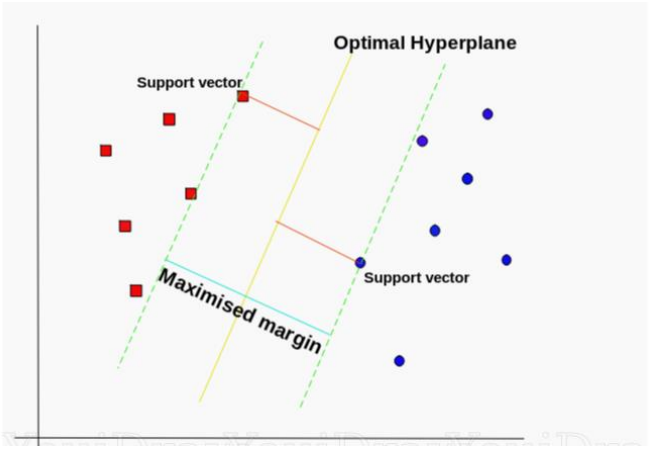


Figure 5. Optimal hyperplane [37].

2.2.6. Classification Ensembles

A classification ensemble combines different machine learning techniques models to improve the model performance by decreasing variance (bagging), bias (boosting), or improving predictions (stacking) [38]. One of the most popular ensembles learning algorithms is adaptive boosting (AdaBoost) that uses the boosting method to convert weak learners to strong learners [39]. Given a dataset of N data points, the AdaBoost algorithm firstly initializes the weights for each data point. Then it fits weak classifiers to the data set and selects the one with the lowest weighted classification error. For each iteration, it computes the weight for each weak classifier related to each data point. The final classifier can be expressed as:

$$F(x) = sign(\sum_{m=1}^{M} \theta_m f_m(x)) \tag{9}$$

where, f_m represents the m-th weak classifier and θ_m is the corresponding weight. Therefore, the final classifier (strong classifier) $F(x)$ is given by a weighted summing of M weak classifiers as Figure 6 shows.

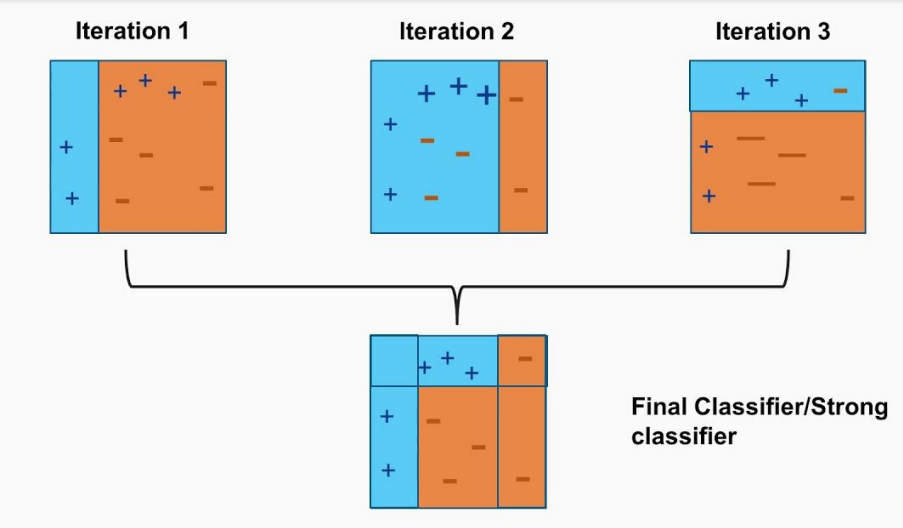


Figure 6. Adaptive boosting (AdaBoost) algorithm.

3. Methodology

This section presents the adopted methodology to identify the most suitable model between the SDM and the DDM under different levels of solar irradiance and temperature by using the ML classification algorithms.

The data collected from supervisory control and data acquisition (SCADA) of a 113.85 kW_P grid-connected PV plant located in southeast Italy (latitude 40°37′55 N, longitude 17°56′9 E) is adopted to carry out the investigation. The PV system includes 414 polycrystalline silicon PV modules with a nominal power of 275 W. The modules are connected in 23 strings of 18 modules, oriented south, and inclined at a tilt angle of 30°. Data of the solar irradiation, ambient temperature, and DC power are collected according to the International Standard IEC 61724. A mean value of one hour of measurements relative to solar irradiance of the array, ambient temperature, and DC output power from 1 October 2017 to 20 September 2018 (8760 sample) is considered in the present study.

Figure 7 shows the hourly solar irradiance incident on the plane of the array and the output power over one year. The hourly output power increases linearly with the increase of solar irradiance on the tilted plane with a strong correlation (R^2 = 0.9897).

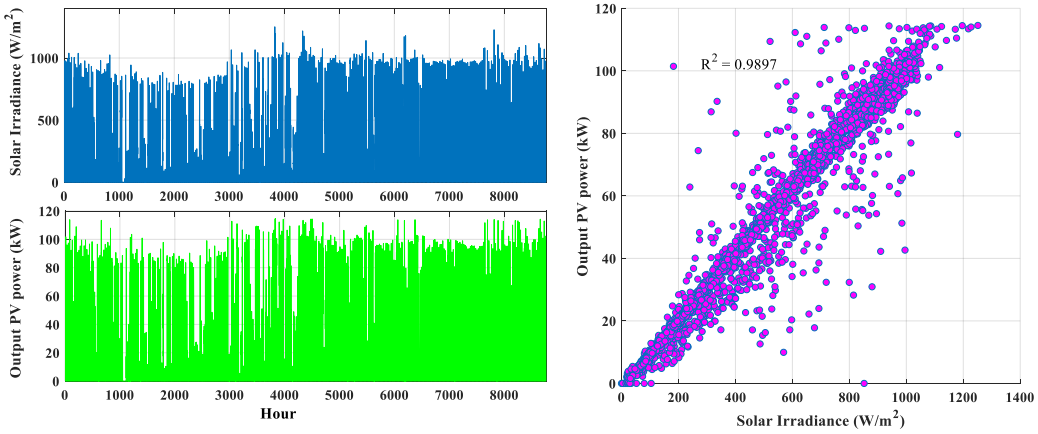


Figure 7. Hourly solar irradiance and the output power of the photovoltaic (PV) array.

The methods of Ishaque et al. [4] and Chaibi et al. [16] are implemented respectively to extract the parameters for SDM and DDM by MATLAB code. For each measurement of irradiance and ambient temperature, the PV output voltage and current at the maximum power point are computed using Equations (6) and (7) for SDM and DDM, respectively.

An estimation of global MPP coordinates is found considering that the PV plant consists of 23 strings of 18 modules 275W (N_s = 23, N_m = 18). Thus, the whole output PV power is computed as $N_s * N_m * P_{MPP}$, where the P_{MPP} is computed for each couple of the hourly monitored data of irradiance and ambient temperature by using the PV output voltage and current at the maximum power point in accordance to Equations (6) and (7) for the SDM and DDM, respectively.

The obtained power output is compared to the actual data by the normalized mean bias error (NMBE) as:

$$\mathrm{NMBE}(\%) = \frac{1}{N}\sum_{1}^{N}\frac{P_{\mathrm{model}} - P_{\mathrm{actual}}}{\max(P_{\mathrm{actual}})} * 100 \; N = 1 \ldots 8760, \tag{10}$$

where, the P_{model} can be P_{SDM} or P_{DDM} and represents the power calculated from SDM and DDM.

Six classification algorithms, as shown in Table 1, are chosen to identify which model between the SDM and DDM provide the best performance for a given solar irradiance and temperature. Therefore, each classification algorithm is based on two predictors: irradiance and temperature.

Table 1. Machine learning classifiers.

Algorithm	Comments/Details
Classification Trees	Coarse Tree
k-Nearest Neighbors	Cubic
Discriminant Analysis	Quadratic
Naïve Bayes	Kernel
Support Vector Machine (SVM)	Gaussian
Classification Ensembles	Boosted Trees

Figure 8 depicts the adopted approach to classify the equivalent-circuit models for a given solar irradiance and temperature and to provide the PV output power with the highest accuracy.

In order to assess the performance of the classification algorithms based on ML and the proposed approach, we introduce the accuracy index, the confusion matrix, the receiver operating characteristic (ROC) curve, and the normalized mean absolute error (NMAE).

The accuracy index of the classification algorithms can be evaluated as:

$$\mathrm{Accuracy} \; = \frac{\sum_{i=1}^{q} \mathrm{matched} \; (K(x_i), c_i)}{q}, \tag{11}$$

where $K(x_i)$ is the predicted class by the classifier and c_i is the i-th class. In other terms, it represents the number of the case in which the predicted class matches the expected class. In the dataset, if the classes are not equally distributed, the classifier cannot be accurate. In order to overcome this limitation, the "cross-validation (CV)" method is applied. It divides the dataset into k equal partitions (k folders), by generating k testing sets and using the remain data as the training set. Then, a classifier is evaluated for k iterations. In the present study, k is set to 5.

A further tool to present the classification algorithm performance is the confusion matrix, noted also as an error matrix. The matrix includes the predicted true/false values and actual true/false values as shown in Figure 9. The prediction is correct for true positive (TP) and true negative (TN) and prediction fails for false negative (FN) and false positive (FP).

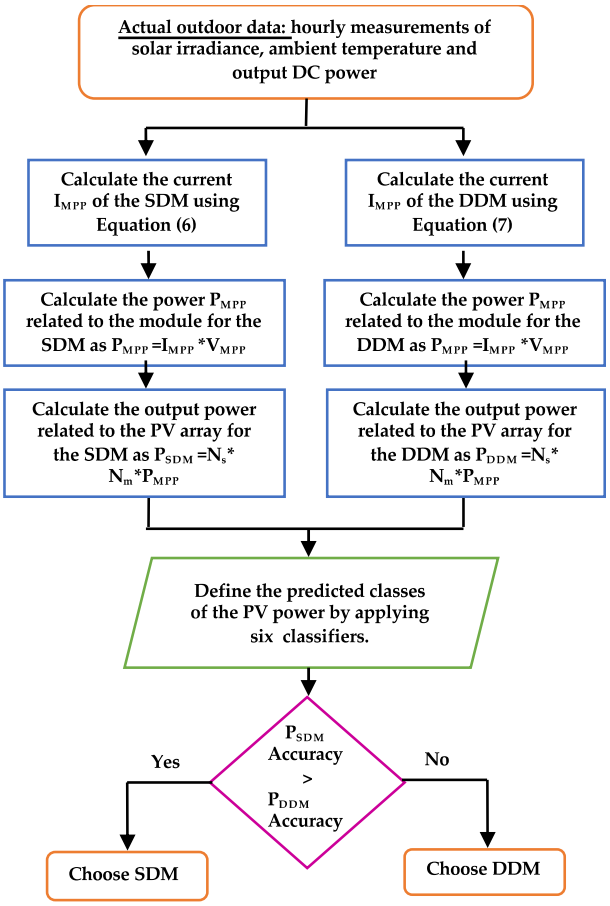


Figure 8. Adopted strategy to classify the adequate equivalent-circuit model according to climatic variations. DC: direct current, SDM: single-diode model, DDM: double-diode model; MPP: Maximum power point.

	Predicted =TRUE	Predicted =FALSE
Actual = TRUE	TP (True Positive)	FN (False Negative)
Actual = FALSE	FP (False Positive)	TN (True Negative)

Figure 9. Predicted true/false values and actual true/false values.

Furthermore, it is possible to define some indexes such as the true positive rate (TPR) and true negative rate (TNR), given by:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})}, \mathrm{TNR} = \frac{\mathrm{TN}}{(\mathrm{TN} + \mathrm{FP})}, \tag{12}$$

where, TPR represents the proportion of TRUE values that are correctly predicted as TRUE and the TNR is defined as the proportion of FALSE observations that are correctly predicted as FALSE. Therefore, the overall accuracy is given by:

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})}, \tag{13}$$

The ROC curve shows a true positive rate versus false positive for different thresholds of the classifier output. It can be used to find the threshold that maximizes the classification accuracy.

In order to evaluate the performance of the implemented classification algorithms in term of PV output power, the predicted and experimental data are used to compute the percentage value of the normalized mean absolute error (NMAE) for each case, as follows:

$$\mathrm{NMAE}(\%) = \frac{1}{N}\sum_{1}^{N}\left|\frac{\mathrm{Predicted}(i) - \mathrm{Actual}(i)}{\mathrm{Max}_1^N(\mathrm{Actual}(i))}\right| * 100 \; i = 1 \ldots 2880, \tag{14}$$

where, N is the number of samples used for the testing step.

4. Results and Discussion

In order to address the performance of both SDMs and DDMs under different levels of irradiance and temperature, we identify the low values class when the irradiance is below 400 W/m^2, the medium is between 400 W/m^2 and 800 W/m^2 and the high values class is above 800 W/m^2. For the temperature changes, the low variations are below 20 °C, the medium are between 20 °C and 40 °C, and the last class is for temperatures above 40 °C.

Table 2 includes the normalized mean bias error of PV output power, as defined by Equation (10), using the SDM and the DDM for low, medium, and high classes of solar irradiance and temperature.

Table 2. Normalized mean bias error for SDM and DDM.

		SDM	DDM
Temperature	Low	0.54%	0.49%
	Medium	1.98%	1.49%
	High	0.60%	–0.79%
Irradiance	Low	0.81%	0.69%
	Medium	3.04%	1.92%
	High	0.05%	–1.36%

At low and medium changes of solar irradiance, the DDM exhibits more accuracy with a low value of bias error which explains an overestimation of output power that does not exceed 1.92% compared to SDM (overestimation up to 3.04%). For high irradiance levels, the SDM shows a positive error that means an overestimation of the PV power output, unlike the DDM that shows a negative error (underestimation). In terms of temperature, both models present close behaviors for low and medium temperatures, but SDM shows higher error than DDM for medium temperature only. For high-temperature level, SDM shows positive error (overestimation), unlike DDM, which shows a negative error (underestimation). Therefore, the equivalent-circuit models perform in a different manner under various levels of irradiance and temperature, demonstrating a high influence of climatic conditions on the accuracy of the SDMs and DDMs.

In the next step, six classification algorithms are implemented using 5880 samples, about 70% of the data related to the whole year, for the training and the remaining (about 30%) (2880 samples) for the validation. In particular, the months of February, May, August, and November were chosen to test the models.

In order to validate what was claimed previously about equivalent-circuit models classifications according to climatic variations, the predicted power of the SDM and DDM are plotted and ranged in Figure 10. The PV power values are classified using different algorithms and this is for a large variation of solar irradiance and temperature. As seen in this figure, most predictions are correct, and it is clear the power increases linearly with irradiance and temperature.

Performance of the Classification Algorithms during the Training

The performance of the classification algorithms was investigated by using the classification confusion matrix and receiver operating characteristic (ROC) curve.

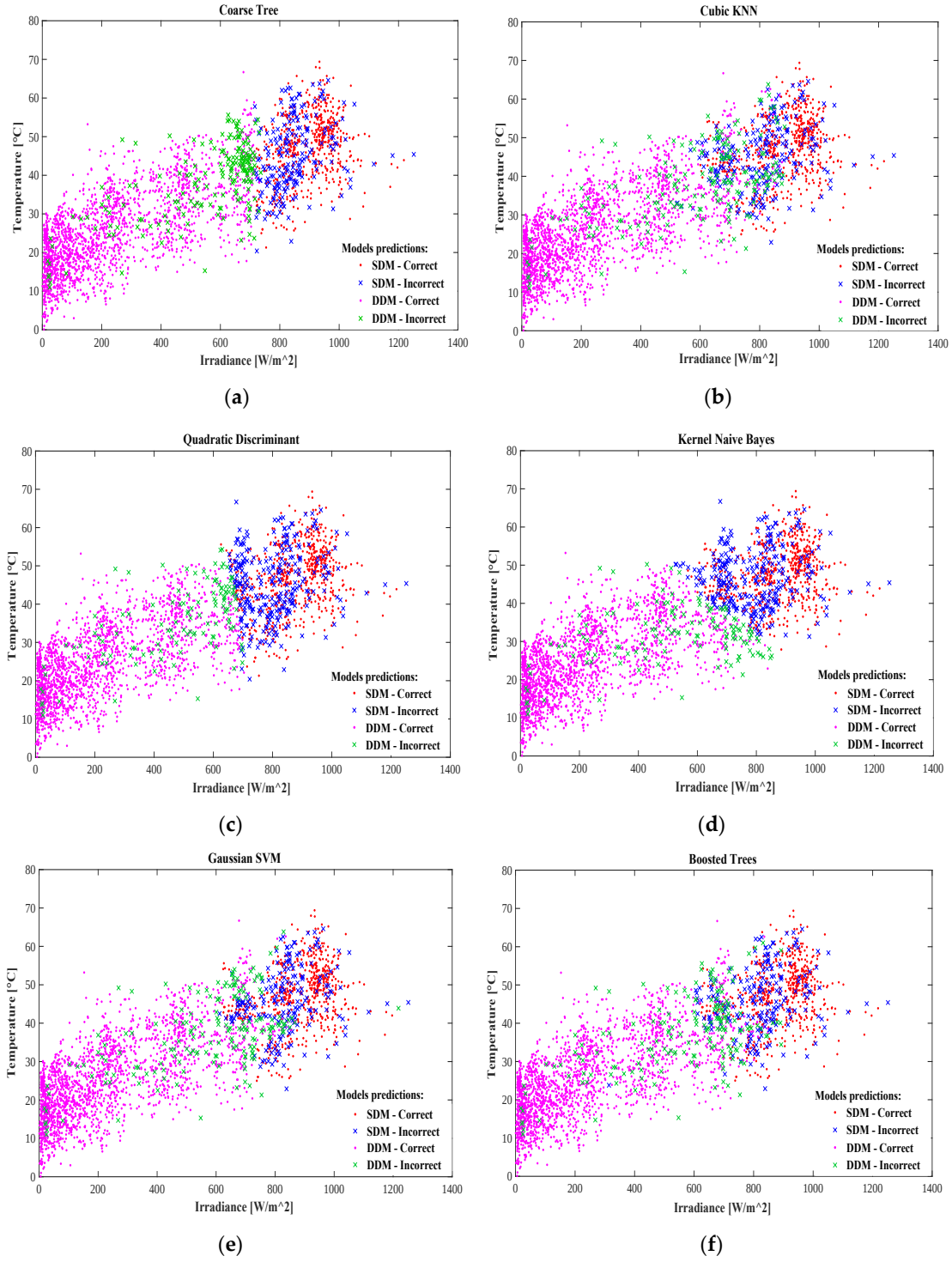


Figure 10. Predicted powers using SDM and DDM under a variation of solar irradiance and temperature. (**a**) Coarse Tree; (**b**) Cubic KNN; (**c**) Quadratic Discriminant; (**d**) Kernel Naïve Bayes; (**e**) Gaussian SVM; (**f**) Boosted trees.

In the confusion matrix, the rows correspond to the predicted class (output class) and the columns correspond to the true class (target class). The cm (*i*,*j*) is the number of samples (or percentage of samples) whose target is the *i*-th class that is classified as *j*. It represents the percentages of all the examples predicted to belong to each class that is correctly and incorrectly classified. These metrics are often called the precision (or positive predictive value) and false discovery rate, respectively.

In Figure 11, the cells show the percentage of correct classifications by the trained network. In the case of course tree, 91% of samples are correctly classified as the DDM and similarly, 75% of samples are correctly classified as the SDM. Cubic KNN, Gaussian SVM, and boosted trees show the same trend as the course tree algorithm for TPR. Quadratic discriminant and kernel Naïve Bayes present lower TPR (87%) for thr DDM which means that 13% of samples are incorrectly predicted as a SDM. In the case of SDM class the TPR is higher than the corresponding one of course tree, cubic KNN, Gaussian SVM, and boosted trees (84% for quadratic discriminant and 82% for kernel Naïve Bayes). Therefore, the last two models fail for 16% and 18% of samples, respectively.

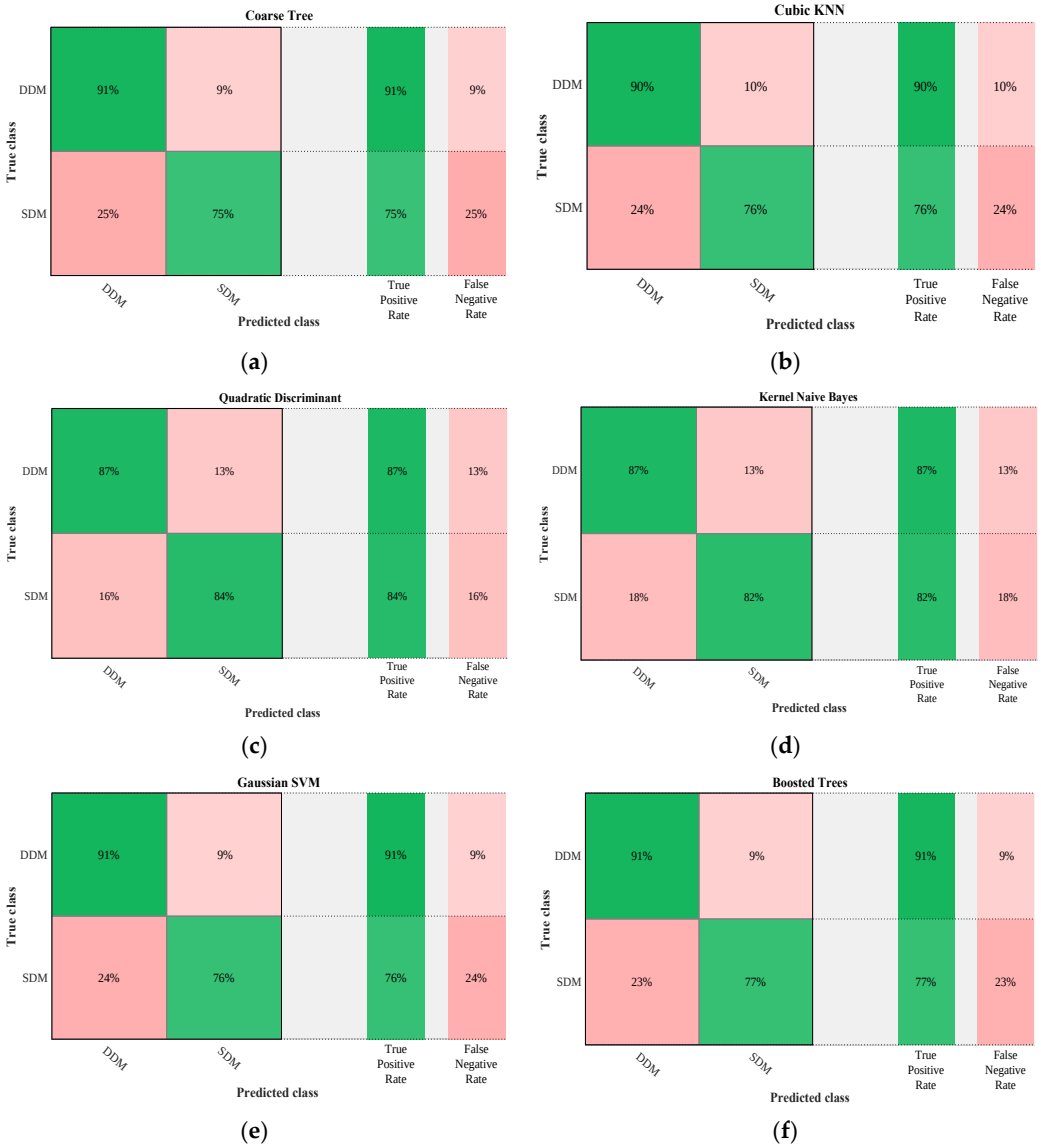


Figure 11. Confusion matrix related to six classification algorithms. (**a**) Coarse Tree; (**b**) Cubic KNN; (**c**) Quadratic Discriminant; (**d**) Kernel Naïve Bayes; (**e**) Gaussian SVM; (**f**) Boosted trees.

The optimal operating points on the ROC curve for each classification model are plotted in Figure 12. The ROC curve for Naïve Bayes is generally lower than the other two ROC curves, which indicates worse in-sample performance than the other two classifier methods. By comparison of the area under the curve (AUC) for all classifiers, classification tree and SVM have the lowest AUC measure, meanwhile Naïve Bayes and quadratic discriminant have the highest AUC value. Therefore, the classification tree and SVM present high performance for the considered sample data.

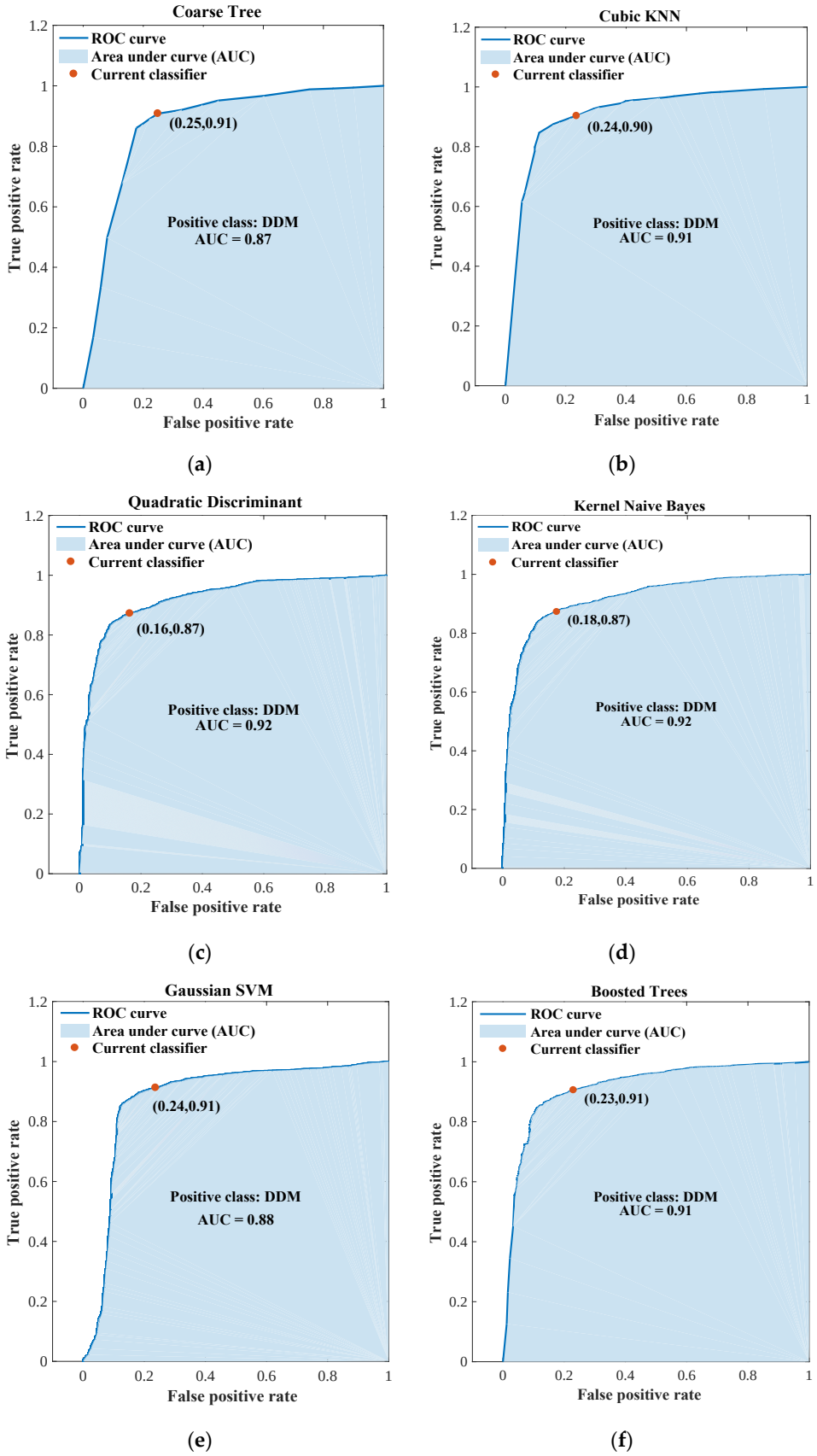


Figure 12. Receiver operating characteristics (ROC) curve for each classification model. (**a**) Coarse Tree; (**b**) Cubic KNN; (**c**) Quadratic Discriminant; (**d**) Kernel Naïve Bayes; (**e**) Gaussian SVM; (**f**) Boosted trees.

Table 3 summarizes the performance of the classification algorithms during the training in terms of TPR, AUC, and accuracy as defined by Equation (11).

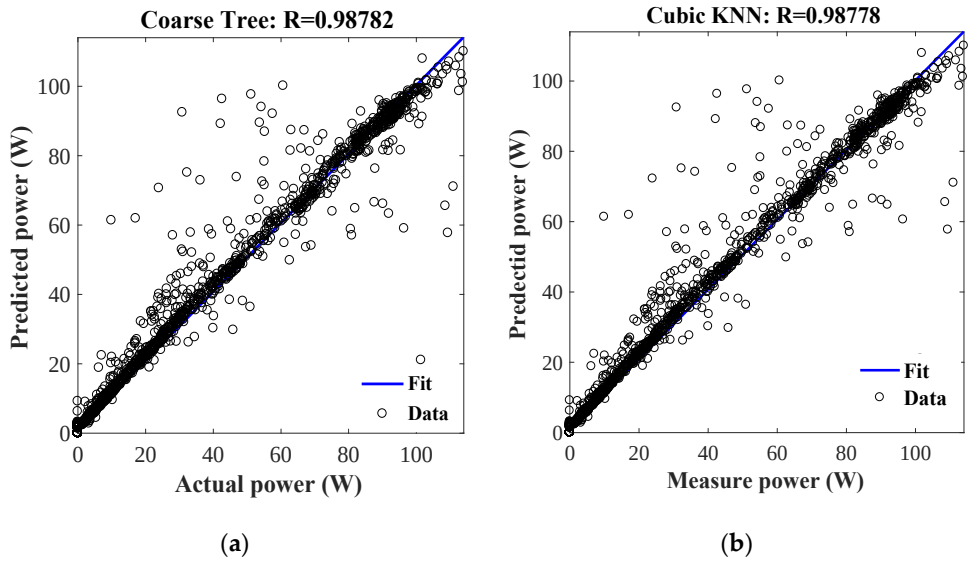
Table 3. TPR, area under the curve (AUC), and accuracy for each classification model.

Classification Models	TPR (%)		AUC	Accuracy (%)
	SDM	DDM		
Classification Trees	75	91	0.87	86.8%
k-Nearest Neighbors	76	90	0.91	86.8%
Discriminant Analysis	84	87	0.92	86.3%
Naïve Bayes	82	87	0.92	86.0%
SVM	76	91	0.88	87.5%
Classification Ensembles	77	91	0.91	87.1%

In Table 3, it is clear that the SVM classifier presents the highest accuracy with a value of 87.5%. However, the Naïve Bayes provides the lowest accuracy with a mean value of 86%.

In order to assess the performance of the classification algorithms in terms of PV power output predicted, a test dataset related to the months of February, May, August, and November was chosen for a total of 2880 samples. Figure 13 shows the linear regression of actual power (targets) relative to predicted power (outputs). High R values demonstrate that the ML classification algorithms are very suitable to predict the output power based on the hybrid modeling between SDM and DDM.

The NMAE for the SDM and DDM related to the testing dataset of 2880 samples was computed of 1.634% and 1.523% respectively, as Table 4 shows. In the same table can also be observed that NMAEs average value for the classification algorithms is 1.48%. Therefore, the ML classification algorithms can improve the accuracy of the PV modeling based on the traditional SDM and DDM models to the different solar irradiance and temperature. The potential of error reduction is estimated between 0.04% and 0.15%



(**a**) (**b**)

Figure 13. *Cont.*

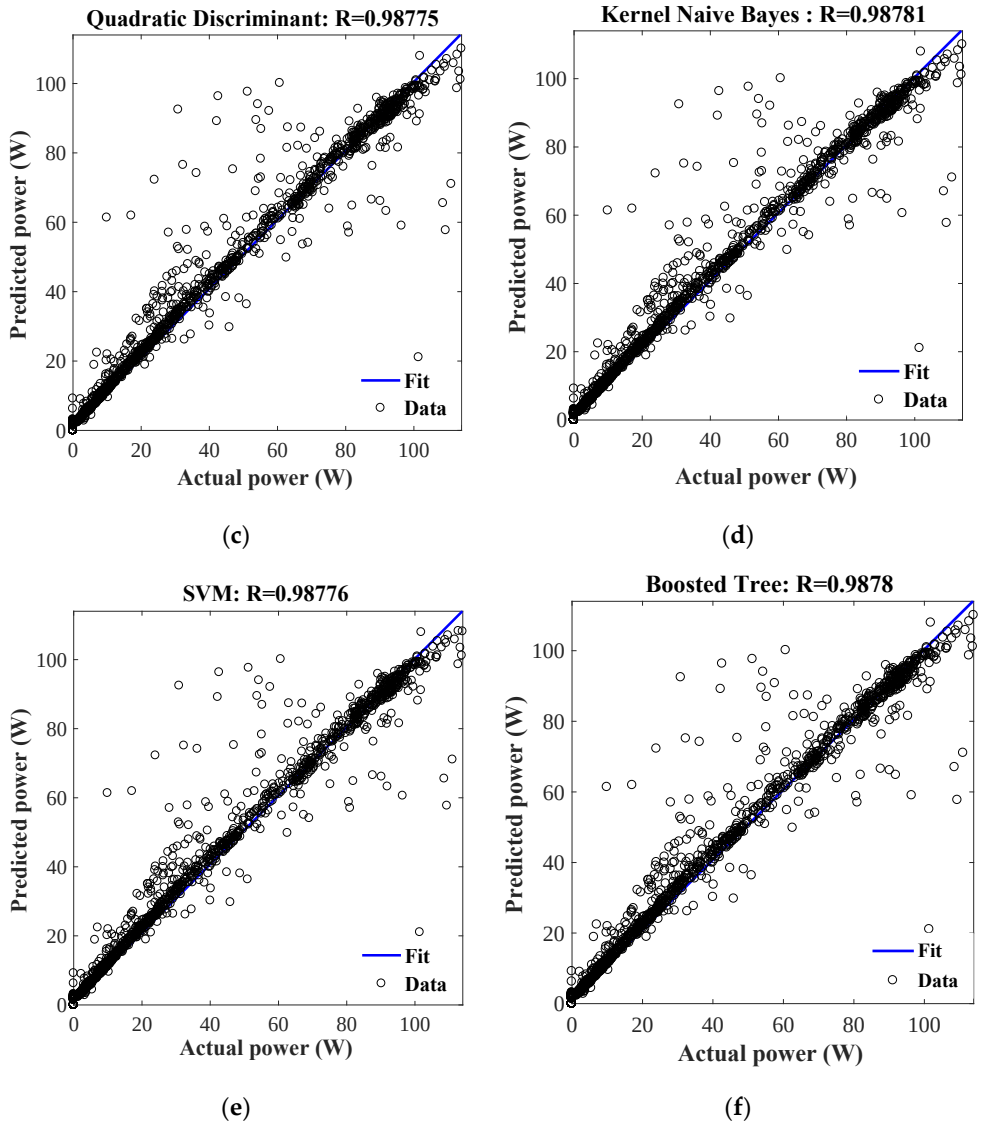


Figure 13. Linear regression of targets and outputs for each classification algorithm. (**a**) Coarse Tree; (**b**) Cubic KNN; (**c**) Quadratic Discriminant; (**d**) Kernel Naïve Bayes; (**e**) Gaussian SVM; (**f**) Boosted trees.

Table 4. R and normalized mean absolute error (NMAE) related to six classification algorithms.

Algorithms	R	NMAE (%)
Classification Trees	0.98782	1.476
k-Nearest Neighbors	0.98778	1.469
Discriminant Analysis	0.98775	1.483
Naïve Bayes	0.98781	1,483
SVM	0.98776	1.472
Classification Ensembles	0.9878	1.473
SDM	-	1.634
DDM	-	1.523

5. Conclusions

Prediction of PV module performances becomes an important task in order to anticipate the long-term functioning of PV systems. In literature, the PV modeling techniques adopt the equivalent-circuit models whose performances are influenced by climatic conditions. This paper

presents a classification method of the single-diode and double-diode equivalent-circuit models under real operating conditions of irradiance and temperature.

A hybrid approach based on the ML classification algorithms is proposed to combine SDM and DDM according with the corresponding accuracy. Six classification algorithms, such as classification trees, k-nearest neighbors, discriminant analysis, Naïve Bayes, support vector machines, and classification ensembles were implemented in order to identify which model between the SDM and DDM provides an estimation of the output power of a PV array with higher accuracy for a given solar irradiance and temperature. The algorithms were fitted using the hourly measurements of solar irradiance on the plane of the array and ambient temperature over one year and related to a Poly-Si 113.85 kWp grid-connected PV plant located in southeast Italy, characterized by the Mediterranean climate. High accuracy demonstrates the high potential of six classification algorithms in the PV power predicting. During the training process, the support vector machines classifier presents the highest TPR of 91% for DDM and an accuracy with a value of 87.5%. However, the Naïve Bayes provides the lowest values of TPR (87%) and accuracy (86%). In the validation phase, the performance assessment in terms of NMAE demonstrates that the hybrid approach using ML classifiers presents lower errors compared to the use of only SDMs or DDMs with an error reduction up to 0.15%. This error achieved the lowest value for the k-nearest neighbors algorithm with a value of 1.469%.

Author Contributions: M.M. and Y.C. contributed to the design and implementation of the research, to the analysis of the results, to the writing and the reviewing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The research work of the School of Electrical and Computer Engineering, National Technical University of Athens, Greece for the presented study received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 799835.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

The appendix contains the analytical expressions of the classification methods.

Appendix A.1 k-Nearest Neighbors

Let x and x' be a training and test sample respectively, and c and c' be the true class of a training sample and the predicted class for a test sample, the Euclidean distance between a test sample and the training samples.

$$d(x', x_i) = \sqrt{(x' - x_{i1})^2 + (x' - x_{i2})^2 \ldots\ldots + \left(x' - x_{ij}\right)^2} \quad \text{(A1)}$$

n is the total number of input samples (i = 1,2, … ,n) and j is the total number of features (j = 1,2, … ,p).

In the kNN classification for k = 1 the predicted class of test sample x' is set equal to the true class c of its nearest neighbor, where m_i is the nearest neighbor to x if the distance is:

$$d(m_i, x) = \min_j \{d(m_i, x)\} \quad \text{(A2)}$$

For k-nearest neighbors, the predicted class of test sample x is set equal to the most frequent true class among k nearest training samples [31].

Appendix A.2 Discriminant Analysis

Given p variables, K classes, and N_k, the total number of observations for each class S_t, S_w, and Sa, is defined as follows [32]:

$$S_T = \sum_{k=1}^{K} \sum_{i=1}^{N_k} (X_{ki} - M)(X_{ki} - M)' \quad \text{(A3)}$$

$$S_W = \sum_{k=1}^{K} \sum_{i=1}^{N_k} (X_{ki} - M_k)(X_{ki} - M_k)' \tag{A4}$$

$$S_A = S_T - S_W \tag{A5}$$

where, X_{ki} represents the i-th observation in the k-th class, M is a vector of the mean value for each class, and M_k is the vector of means of observations in the k-th class. The discriminant function is defined as the weighted average of the independent variables. The weights can be found by solving the eigenvectors V as

$$V = {S_W}^{-1} S_A \tag{A6}$$

where, the elements of eigenvectors are the canonical coefficients and the correlations between the independent variables and the canonical variates are given by:

$$Corr_{jk} = \frac{1}{\sqrt{w_{jj}}} \sum_{i=1}^{p} v_{tk} w_{ji} \tag{A7}$$

where, Vj are the elements of V and Wj are the elements of W. The within-group covariance matrix, W, is given by:

$$W = \left(\frac{1}{N-K}\right) S_W \tag{A8}$$

Appendix A.3 Naïve Bayes

Given an instance and its occurring probability $p(d)$, the Bayes theorem says:

$$P(c_i|d) = \frac{P(d|c_i) * P(c_i)}{P(d)} \tag{A9}$$

where:

$P(c_i|d)$ is the probability of the instance d being in the class c_i;
$P(d|c_i)$ is the probability of observing d in a domain where c holds;
$P(c_i)$ is the prior probability of c_i.

It is assumed that all instances show an independent distribution and all classes occur with the same probability $P(c_i) = P(c_j)$, $P(d|c_i)$ can be simplified as follows:

$$P(d|c_i) = P(d_1|c_i) * P(d_2|c_i) \ldots\ldots P(d_n|c_i) \tag{A10}$$

where each $P(d_n|c_i)$ and $P(c_i)$ can be estimated by statistical analysis of features and classes of the training dataset. The Naïve Bayes classifier computes the probability of each instance for all classes in C and select the class cj with the highest probability $P(c_i|d)$, denoted as c_{MAP} and noted as "maximum a posteriori (MAP)" class:

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c_i|d) \tag{A11}$$

Generally, the features are assumed to have a Gaussian probability distribution as follows:

$$P_{Gaussian}(x|c) = \frac{1}{\sigma_c \sqrt{2\pi}} e^{-(x-\mu_c)^2/2{\sigma_c}^2} \tag{A12}$$

The mean μ_c is the average of all values of the feature found in the dataset D. When the features do not follow a Gaussian distribution, the kernel density method [34] is applied to estimate the probability distribution, as follows:

$$P_{kernel}(x|c) = \frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{\sigma_c\sqrt{2\pi}}e^{-(x-\mu_{cj})^2/2{\sigma_c}^2}\right) \tag{A13}$$

where, j is the j-th element of the dataset $D^m \subset D$ given by m samples. The kernel method performs m estimation of the Gaussian probability, unlike the probability which is evaluated only once using Equation (A12). In the present study, the kernel-based Naïve Bayes method is used to estimate the probability distribution.

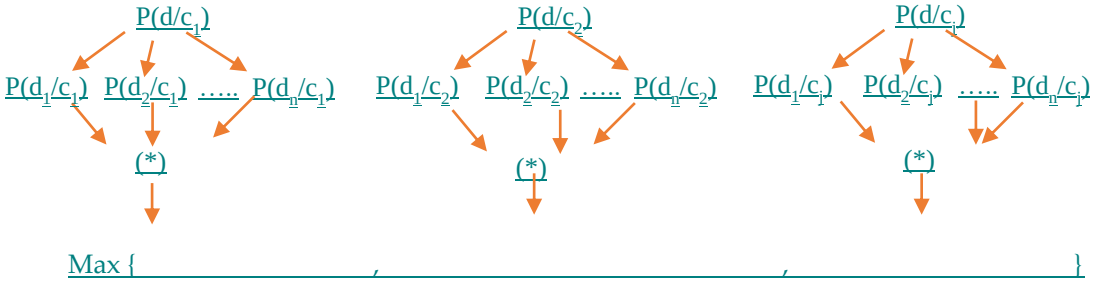


Figure A1. "Maximum a posteriori" (MAP) class [34].

Appendix A.4 Support Vector Machines

Given a training set of N data points, $\mathcal{D}_N = \{x_k,\ y_k\}_{k=1}^N$ where, $x_k\ \epsilon\ \mathbb{R}^d$ is the k-th input data and $y_k\ \epsilon\ \mathbb{R}$ is the k-the output data, the support vector method constructs a classifier as follows [35]:

$$y(x) = sign\left[\sum_{k=1}^{N}\alpha_k y_k \Psi(x,\ x_k) + b\right] \tag{A14}$$

where, $\alpha_k\ \epsilon\ \mathbb{R}$ are positive constant and $b\ \epsilon\ \mathbb{R}$ are a constant. The term $\Psi(x,\ x_k)$ can be a linear, polynomial, exponential function. The classifier is constructed as:

$$y_k[\omega^T\varphi(x_k) + b] \geq 1 - e_k \tag{A15}$$

e_k is a positive artificial variable. The formulation of the classification problem is as follows:

$$\underset{w,e_k}{\text{mix}}\mathcal{J}(w,e_k) = \frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{k}^{N}e_k\ k = 1\ldots N \tag{A16}$$

where, γ is the regularization factor. In order to solve the optimization problem, the Lagrange function is defined as:

$$L(w,b,e,\alpha) = \mathcal{J}(w,b,e) - \sum_{k=1}^{N}\alpha_k\Big\{y_k\Big[w^T\varphi(x_k) + b\Big] - 1 + e_k\Big\}k = 1\ldots N \tag{A17}$$

where, $\alpha_k\ \epsilon\ \mathbb{R}$ are the Lagrange multipliers. The optimal conditions are:

$$\begin{cases} \frac{\partial L}{\partial w} = 0 & \rightarrow w = \sum\limits_{k=1}^{N}\alpha_k\varphi(x_k) \\ \frac{\partial L}{\partial b} = 0 & \rightarrow \sum\limits_{k=1}^{N}\alpha_k y_k = 0 \qquad k = 1\ldots N \\ \frac{\partial L}{\partial e_k} = 0 & \rightarrow \alpha_k = \gamma e_k \\ \frac{\partial L}{\partial \alpha_k} = 0 & \rightarrow y_k\{w^T\varphi(x_k) + b\Big] - 1 + e_k \end{cases} \tag{A18}$$

So, the solution in matrix notation is:

$$\begin{bmatrix} \Omega + \frac{1}{\gamma} I & 1 \\ I^T & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ b \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{A19}$$

Applying the Mercer's theorem

$$\Omega_{kj} = y_k y_j \, \varphi^T(x_k)\varphi\left(x_j\right) = K\left(x_k, \, x_j\right) k, \; \mathrm{j} = 1 \ldots N \tag{A20}$$

where, $K\left(x_k, \, x_j\right)$ is the kernel matrix. Hence, the classifier in Equation (A14) is found by solving the linear set of Equations (A19) and (A20) instead of quadratic programming.

In the present study the radial basis function (RBF) kernel is used and defined as:

$$K\left(x_k, \, x_j\right) = \exp\left(-\frac{\|x_k - \, x_j\|_2^2}{\sigma^2}\right) \tag{A21}$$

where, σ is a tuning parameter.

References

1. Al-Majidi, S.D.; Abbod, M.F.; Al-Raweshidy, H.S. Design of an Efficient Maximum Power Point Tracker Based on ANFIS Using an Experimental Photovoltaic System Data. *Electronics* **2019**, *8*, 858. [CrossRef]
2. Chaibi, Y.; Allouhi, A.; Salhi, M. Annual performance analysis of different maximum power point tracking techniques used in photovoltaic systems. *Prot. Control Mod. Power Syst.* **2019**, *1*, 1–10. [CrossRef]
3. Pindado, S.; Cubas, J.; Roibás-Millán, E.; Bugallo-Siegel, F.; Sorribes-Palmer, F. Assessment of explicit models for different photovoltaic technologies. *Energies* **2018**, *11*, 1353. [CrossRef]
4. Ishaque, K.; Salam, Z.; Taheri, H. Simple, fast and accurate two-diode model for photovoltaic modules. *Sol. Energy Mater. Sol. Cells* **2011**, *95*, 586–594. [CrossRef]
5. Chaibi, Y.; Allouhi, A.; Malvoni, M.; Salhi, M.; Saadani, R. Solar irradiance and temperature influence on the photovoltaic cell equivalent-circuit models. *Sol. Energy* **2019**, *188*, 1102–1110. [CrossRef]
6. Ishaque, K.; Salam, Z.; Shamsudin, A.; Amjad, M. A direct control based maximum power point tracking method for photovoltaic system under partial shading conditions using particle swarm optimization algorithm. *Appl. Energy* **2012**, *99*, 414–422. [CrossRef]
7. Et-torabi, K.; Nassar-eddine, I.; Obbadi, A.; Errami, Y.; Rmaily, R.; Sahnoun, S.; El fajri, A.; Agunaou, M. Parameters estimation of the single and double diode photovoltaic models using a Gauss–Seidel algorithm and analytical method: A comparative study. *Energy Convers. Manag.* **2017**, *148*, 1041–1054. [CrossRef]
8. Villalva, M.G.; Gazoli, J.R.; Filho, E.R. Comprehensive Approach to Modeling and Simulation of Photovoltaic Arrays. *IEEE Trans. Power Electron.* **2009**, *24*, 1198–1208. [CrossRef]
9. Khandakar, A.; EH Chowdhury, M.; Khoda Kazi, M.; Benhmed, K.; Touati, F.; Al-Hitmi, M.; Gonzales, A. Machine Learning Based Photovoltaics (PV) Power Prediction Using Different Environmental Parameters of Qatar. *Energies* **2019**, *12*, 2782. [CrossRef]
10. Zhu, R.; Guo, W.; Gong, X. Short-term photovoltaic power output prediction based on k-fold cross-validation and an ensemble model. *Energies* **2019**, *12*, 1220. [CrossRef]
11. Theocharides, S.; Makrides, G.; Georghiou, G.E.; Kyprianou, A. Machine learning algorithms for photovoltaic system power output prediction. In Proceedings of the 2018 IEEE International Energy Conference (ENERGYCON), Limassol, Cyprus, 3–7 June 2018; pp. 1–6.
12. Shi, J.; Lee, W.J.; Liu, Y.; Yang, Y.; Wang, P. Forecasting power output of photovoltaic systems based on weather classification and support vector machines. *IEEE Trans. Ind. Appl.* **2012**, *48*, 1064–1069. [CrossRef]
13. Malvoni, M.; De Giorgi, M.G.; Congedo, P.M. Forecasting of PV Power Generation using weather input data-preprocessing techniques. *Energy Procedia* **2017**, *126*, 651–658. [CrossRef]
14. Malvoni, M.; Hatziargyriou, N. One-day ahead PV power forecasts using 3D Wavelet Decomposition. In Proceedings of the 2019 International Conference on Smart Energy Systems and Technologies (SEST), Porto, Portugal, 9–11 September 2019; pp. 1–6.

15. Wang, J.; Li, P.; Ran, R.; Che, Y.; Zhou, Y. A short-term photovoltaic power prediction model based on the Gradient Boost Decision Tree. *Appl. Sci.* **2018**, *8*, 689. [CrossRef]
16. Chaibi, Y.; Salhi, M.; El-jouni, A.; Essadki, A. A new method to determine the Parameters of a photovoltaic Panel equivalent circuit. *Sol. Energy* **2018**, *163*, 376–386. [CrossRef]
17. Bana, S.; Saini, R.P. A mathematical modeling framework to evaluate the performance of single diode and double diode based SPV systems. *Energy Rep.* **2016**, *2*, 171–187. [CrossRef]
18. Chin, V.J.; Salam, Z.; Ishaque, K. Cell modelling and model parameters estimation techniques for photovoltaic simulator application: A review. *Appl. Energy* **2015**, *154*, 500–519. [CrossRef]
19. Abbassi, R.; Abbassi, A.; Jemli, M.; Chebbi, S. Identification of unknown parameters of solar cell models: A comprehensive overview of available approaches. *Renew. Sustain. Energy Rev.* **2018**, *90*, 453–474. [CrossRef]
20. Jordehi, A.R. Parameter estimation of solar photovoltaic (PV) cells: A review. *Renew. Sustain. Energy Rev.* **2016**, *61*, 354–371. [CrossRef]
21. Pillai, D.S.; Rajasekar, N. Metaheuristic algorithms for PV parameter identification: A comprehensive review with an application to threshold setting for fault detection in PV systems. *Renew. Sustain. Energy Rev.* **2018**, *82*, 3503–3525. [CrossRef]
22. Chaibi, Y.; Malvoni, M.; Chouder, A.; Boussetta, M.; Salhi, M. Simple and efficient approach to detect and diagnose electrical faults and partial shading in photovoltaic systems. *Energy Convers. Manag.* **2019**, *196*, 330–343. [CrossRef]
23. Chaibi, Y.; Salhi, M.; El-jouni, A.; Essadki, A. A new method to extract the equivalent circuit parameters of a photovoltaic panel. *Sol. Energy* **2018**, *163*, 376–386. [CrossRef]
24. James, M. *Classification Algorithms*; Wiley-Interscience: Hoboken, NJ, USA, 1932.
25. Kotsiantis, S.B. Supervised Machine Learning: A Review of Classification Techniques. In *Frontiers in Artificial Intelligence and Applications*; IOS Press: Amsterdam, The Netherlands, 2007; pp. 249–268.
26. Tan, L. *Code Comment Analysis for Improving Software Quality*; Elsevier Inc.: Amsterdam, The Netherlands, 2015; ISBN 9780124115439.
27. Nisbet, R.; Elder, J.; Miner, G. *Handbook of Statistical Analysis & Data Mining*; Academic Press: Cambridge, MA, USA, 2009; ISBN 9780123747655.
28. Hallinan, J.S. *Data Mining for Microbiologists*, 1st ed.; Elsevier Ltd.: Amsterdam, The Netherlands, 2012; Volume 39, ISBN 9780080993874.
29. Cover, T.M.; Hart, P.E. Nearest Neighbor Pattern Classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
30. Weinberger, K.Q.; Saul, L.K. Distance Metric Learning for Large Margin Nearest Neighbor Classification Kilian. *J. Mach. Learn. Res.* **2009**, *10*, 207–244.
31. Peterson, L.E. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]
32. Katholieke, J.A.K.S.; Vandewalle, J. Least Squares Support Vector Machine Classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300.
33. Rayens, W.S. Discriminant Analysis and Statistical Pattern Recognition. *Technometrics* **1993**, *35*, 324–326. [CrossRef]
34. John, G.H.; Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. In *UAI'95: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2013.
35. Vapnik, V.N. Statistical Learning Theory. *N. Y. Manag. Sci.* **1998**, *3*, 113–129.
36. Gunn, S. Support Vector Machines for Classification and Regression. Available online: https://www.semanticscholar.org/paper/Support-Vector-Machines-for-Classification-and-Gunn/ceb5e9c07f2d95a700c1ed0813dfbae8c3901c18 (accessed on 27 December 2019).
37. Drakos, G. Support Vector Machine vs Logistic Regression. Available online: https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f (accessed on 12 August 2018).
38. Bramer, M. *Principles of Data Mining*; Springer: Berlin/Heidelberg, Germany, 2013. [CrossRef]
39. Freund, Y.; Schapire, R.E. A Short Introduction to Boosting. *J. Jpn. Soc. Artif. Intell.* **1999**, *14*, 771–780.