

Article

Not peer-reviewed version

Euclidean Distance-Based Tree Algorithm for Fault Detection and Diagnosis in Photovoltaic Systems

Youssef Mouleloued , Kamel Kara , [Aissa Chouder](#) , [Abdelhadi Aouaichia](#) , [Santiago Silvestre Berges](#) *

Posted Date: 3 March 2025

doi: 10.20944/preprints202503.0083.v1

Keywords: fault detection and diagnosis; FDD; supervision algorithm; binary classification; short circuits; partial shading; PV systems



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Euclidean Distance-Based Tree Algorithm for Fault Detection and Diagnosis in Photovoltaic Systems

Youssef Mouleloued ¹, Kamel Kara ¹, Aissa Chouder ², Abdelhadi Aouaichia ¹
and Santiago Silvestre ^{3,*}

¹ Laboratoire des systèmes électriques et télécommande, Faculté de Technologie, Université Blida 1, BP 270, Route de Soumaa, 09000, Blida, Algeria; moulelouedyoussef@gmail.com; km kara@yahoo.fr; aouaichia.abdelhadi@gmail.com

² Electrical Engineering Laboratory (LGE), University Mohamed Boudiaf of M'sila, BP 166, 28000, M'sila, Algeria; aissa.chouder@univ-msila.dz

³ Department of Electronic Engineering, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain

* Correspondence: santiago.silvestre@upc.edu

Abstract: In this paper, a new methodology for fault detection and diagnosis in photovoltaic systems is proposed. This method employs a novel Euclidean distance-based tree algorithm to classify various considered faults. Unlike the decision tree, which requires the use of the Gini index to split the data, this algorithm mainly relies on computing distances between an arbitrary point in the space and the entire dataset. Then, the minimum and the maximum distances of each class are extracted and ordered in ascending order. The proposed methodology requires four attributes: Solar irradiance, temperature, and coordinates of maximum power point (I_{mpp} , V_{mpp}). The developed procedure for fault detection and diagnosis is implemented and applied to classify a dataset comprising seven distinct classes: normal operation, string disconnection, short circuit of three modules, short circuit of ten modules, and three cases of string disconnection with 25%, 50%, and 75% of partial shading. The obtained results demonstrate the high efficiency and effectiveness of the proposed methodology, with a classification accuracy reaching 97.33%. A comparison study between the developed fault detection and diagnosis methodology and Support Vector Machine, Decision Tree, Random Forest, and K-Nearest Neighbors is conducted. The proposed procedure shows high performance against the other algorithms in terms of accuracy, precision, and recall.

Keywords: fault detection and diagnosis; FDD; supervision algorithm; binary classification; short circuits; partial shading; PV systems

1. Introduction

The worldwide demand for electrical energy continues to increase and the governments of different nations have to face several challenges to effectively respond to this demand. The first challenge is to provide energy for the growing proportion of the world's population [1–3]. While the second challenge lies in the production of this energy without causing environmental pollution, as well as causing climatic problems such as global warming [4–6].

One of the ways to reduce greenhouse gas emissions is to use renewable energy, such as wind and solar energies. The wind and the sun provide infinite amounts of energy without generating greenhouse gases, unlike burning the fossil fuels electric power stations. Solar energy allows producing electricity from photovoltaic panels or solar thermal power stations, thanks to sunlight captured by solar panels. Solar energy is clean, does not emit any greenhouse gases and its source the sun, is free, inexhaustible and available everywhere in the world. Several countries around the globe are already at the forefront of renewable energy technologies and generate a large part of their electricity from photovoltaic systems (PVS).

Like all other industrial process, a PVS can be subjected, during its operation, to various faults and anomalies leading to a drop in the performance of the system and even to the total unavailability of the system. These faults will obviously reduce the productivity of the installation [7] and generate an additional cost of maintenance to restore the system to normal conditions. Hence, the importance of having a detecting and diagnosing faults in system photovoltaic installation, which contributes to raising production efficiency and reducing maintenance time and cost [8].

There are many research contributions over the past decades in developing methods and algorithms for detecting and diagnosing faults in PV systems [9–11]. According to references [12,13], these algorithms can be classified into three distinct categories, whereas in reference [14], they are grouped into six categories. In this quick narration, the first categorization is adopted.

The first category encompasses all algorithms that use mathematical analysis and signal processing. Methods within this category heavily rely on the information extracted solely from the I-V characteristic, whether it pertains to a photovoltaic module (PVM), a PV string, or a photovoltaic array (PVA). Time Domain Reflectometry (TDR) is a technique utilized for identifying faulty photovoltaic modules within a photovoltaic array [15]. It has been employed to detect open circuits in grid-connected photovoltaic systems (GCPV) [16]. Earth Capacitance Measurement (ECM) has been added to the TDR to identify the PV module disconnected from the PV string [17]. Reference [18] has demonstrated the applicability of the ECM algorithm in PV strings made of both silicon and amorphous silicon.

The second category comprises several algorithms characterized by two main phases: the detection phase utilizing a PV model and the subsequent diagnosis phase employing various methods, such as artificial intelligence [19–22]. These algorithms detect faults by comparing the measured values extracted from the considered PV generator with the simulated values from the PV model. The residual signal derived from this comparison can be utilized to detect degradation faults [23] as well as various cases of line-line faults [24].

The third category encompasses artificial intelligence and machine learning algorithms, including support vector machine (SVM) [25–29], decision tree (DT) [7,30], random forest (RF) [31–33], K-nearest neighbors (KNN) [34–36], and artificial neural network (ANN) [37–39]. Reference [25] provides a comparison of efficiency and execution time among various multi-class strategies—such as one vs. all (OVA), adaptive directed acyclic graph (ADAG), and decision directed acyclic graph (DDAG)—utilizing SVM. The goal of SVM classification is to categorize data into four classes: module short circuit, inverse bypass diode module, shunted bypass module, and shadowing effect in a module. The OVA strategy has demonstrated significant superiority over others in terms of efficiency, achieving an 88.33% accuracy rate. In reference [26], the SVM algorithm was employed to detect series faults of 10%, 50%, 70%, and 90% under sunny, cloudy, and rainy weather conditions. The recorded accuracies were 88.3%, 91.5%, and 75.3%, respectively. In reference [27], both the CPA and SVM algorithms were utilized to identify four operating states: normal, open circuit, short circuit, and partial shading. The authors concluded that with $k = 6$ (number of dimension), the algorithm achieved an accuracy rate of 100%. In reference [28], the authors used the SVM algorithm to detect faults such as open circuit, short circuit, and lack of solar radiation. The algorithm requires four inputs: short circuit current I_{sc} , open circuit voltage V_{oc} , and coordinates of maximum power point $Impp$ and $Vmpp$. The algorithm's efficiency and accuracy were enhanced by employing k-fold cross-validation. The drawback with the mentioned algorithms is that their authors solely relied on accuracy as the criterion for evaluation, whereas employing different metrics like precision and recall could offer a more comprehensive assessment of the algorithms.

In [7], a novel approach based on DT algorithm is presented. This approach comprises two models: the first model detects faults, while the second model diagnoses four different fault types: short circuit, string, line-line, and free faults. The accuracy rates for the first and second models are 99.86% and 99.80%, respectively. Notably, although a confusion matrix was calculated, precision and recall metrics were not evaluated. The utilization of the random forest algorithm for fault detection and classification in PV systems is highlighted in [31]. The method introduced in this study necessitates

the current from each string in the PV array, along with the PV array voltage, as features. Successfully, the algorithm detects and diagnoses four different faults: degradation, partial shading, line-to-line, and short circuit faults. Authors have employed the grid-search method to optimize the random forest parameters. In order to evaluate this method, experimental and simulation samples were used.

The accuracy of this method reached 99%. Another method was developed based on RF to detect and diagnose faults in photovoltaic systems [32]. A set of criteria were used to evaluate the method, which are computation time, accuracy, and F1 score. In [34], a modified KNN algorithm was proposed and applied to photovoltaic systems for fault detection and diagnosis. The main modification made by the researchers in this work is to facilitate the selection of the appropriate K value in addition to the distance function. This modification greatly contributed to the increase in the classification speed. Moreover, in [35], an interesting technique, that uses the KNN algorithm, has been developed to detect multiple faults, including line to line and partial shading faults. Remarkably, this method only relies on data from the datasheet and achieves an accuracy rate of 99%. Another model developed is based on the combination of KNN and Exponential Weighted Moving Average (EWMA) [36]. The KNN aims to detect faults on the DC side of the PV system, while the EMWA works to diagnose those faults. Researchers in [38] built a two-stage classifier: the first stage is a model of the PV system to detect faults, while the second stage devoted for diagnosis purposes where two artificial neural networks were used to identify eight different faults. In [39], another approach based on artificial neural networks for fault detection and diagnosis in PV systems is introduced. The authors utilized an ANN with radial basis function (RBF) architecture, relying on two features: power generated and solar irradiance. The achieved accuracy in this study was 97.9% for the 2.2 KW PV system and 97% for the 4.16 KW PV system.

In this work, a novel fault detection and diagnosis algorithm is developed and designed in the DC of PV systems. This methodology is based on an innovative tree algorithm that mainly depends on calculating Euclidean distances to detect faults when they occur, effectively. At first, the algorithm classifies the data into two classes so that all the distances between a random point in space and the entire data set are calculated. Then, in each class, the minimum and maximum distances are extracted. After that, all the distances are arranged in ascending order to show one case out of 5 possible cases. Based on the apparent case, the data are classified. The algorithm needs 4 features to function properly: solar radiation, temperature, and current and voltage at the maximum power point. This algorithm has been implemented to seven different classes: normal operation, series disconnection, short circuit for 3 and 10 modules and three other classes for series disconnection with partial shading of 25%, 50% and 75%. The efficiency and effectiveness of the methodology is clearly demonstrated by the accuracy rate achieved which exceeds 97%. In order to further evaluate the algorithm, a comparative study was conducted between the proposed algorithm and several well-known algorithms (support vector machine, K nearest neighbor, decision tree and random forest). The comparison results show a clear superiority of the developed algorithm in terms of accuracy, precision and recall.

This paper is organized as follows: Section 2 is dedicated to introducing the developed algorithm, while Section 3 explains the database used and its different categories. Section 4 reveals the classification strategy followed in this work. In Section Five, the results obtained are presented and discussed. The last section of this paper presents a summary of the work done in this research paper

2. Proposed Euclidean-Based Decision Tree Classification Algorithm

Despite the similarities between the proposed algorithm and the decision trees in their data splitting approach, the key distinction lies in using the Euclidean distance for partitioning data instead of the Gini index. Initially, a training dataset, comprising values for N features for each of the two classes (class 0 and class 1), is created. Then, the following steps are performed:

1. Choose an arbitrary point $(x_1, x_2, \dots, x_N)$ in an N-dimensional space.
2. Using equations 4.1 and 4.2, compute the Euclidean distances between the chosen point and all samples within the training dataset for each respective class:

$$\text{dist}_i^0 = \sqrt{(x_1 - x_{i1}^0)^2 + (x_2 - x_{i2}^0)^2 + \dots + (x_N - x_{iN}^0)^2} \quad i = 1, 2, \dots, n \quad (1)$$

$$\text{dist}_i^1 = \sqrt{(x_1 - x_{i1}^1)^2 + (x_2 - x_{i2}^1)^2 + \dots + (x_N - x_{iN}^1)^2} \quad i = 1, 2, \dots, m \quad (2)$$

where:

$(x_{i1}^0, x_{i2}^0, \dots, x_{iN}^0)$ and $(x_{i1}^1, x_{i2}^1, \dots, x_{iN}^1)$ represent the i^{th} samples of class 0 and class 1, $i_1, i_2, \dots, i_m$ respectively. n denotes the number of samples in class 0, while m denotes the number of samples in class 1.

3. Determine the minimum and maximum distances for each class:

$$\min_0 = \min_{i=1,2,\dots,n} (\text{dist}_i^0) \quad (3)$$

$$\max_0 = \max_{i=1,2,\dots,n} (\text{dist}_i^0) \quad (4)$$

$$\min_1 = \min_{i=1,2,\dots,m} (\text{dist}_i^1) \quad (5)$$

$$\max_1 = \max_{i=1,2,\dots,m} (\text{dist}_i^1) \quad (6)$$

4. Among the following five cases, one may arise:

- case 1: $\min_0 < \min_1 < \max_0 < \max_1$
 - Training samples having distances within the interval $[\min_0, \min_1[$ belong to class 0 (pure data in class 0).
 - Training samples having distances within the interval $] \max_0, \max_1]$ belong to class 1 (pure data in class 1).
 - Training samples having distances within the interval $[\min_1, \max_0]$ can not be classified, therefore another random point must be chosen for their classification. 164
- case 2: $\min_1 < \min_0 < \max_1 < \max_0$
 - Training samples having distances within the interval $[\min_1, \min_0[$ belong to class 1 (pure data in class 1).
 - Training samples having distances within the interval $] \max_1, \max_0]$ belong to class 0 (pure data in class 0).
 - Training samples having distances within the interval $[\min_0, \max_1]$ can not be classified, therefore another random point must be chosen for their classification. 172
- case 3: $\min_0 < \min_1 < \max_1 < \max_0$
 - Training samples having distances within the interval $[\min_0, \min_1[$ or $] \max_1,$

- $max_0]$ belong to class 0.
- Training samples having distances within the interval $[min_1, max_1]$ can not be classified, therefore another random point must be chosen for their classification.
 - case 4: $min_1 < min_0 < max_0 < max_1$
 - Training samples having distances within the interval $[min_1, min_0[$ or $]max_0, max_1]$ belong to class 1.
 - Training samples having distances within the interval $[min_0, max_0]$ can not be classified, therefore another random point must be chosen for their classification.
 - case 5: $min_0 < max_0 < min_1 < max_1$ or $min_1 < max_1 < min_0 < max_0$
 - Training samples having distances within the interval $[min_0, max_0]$ belong to class 0.
 - Training samples having distances within the interval $[min_1, max_1]$ belong to class 1.
5. If the case that occurred in the previous step is case 1, 2, 3, or 4:
- Choose another random point $(x_1, x_2, \dots, x_N)$.
 - Using equations 4.1 and 4.2, compute the Euclidean distances between the chosen point and the unclassified samples within the training dataset for each respective class.
 - Go to step 3.
6. The algorithm iterates through steps (c) to (e) until all data is classified (case 5) or the stopping criterion is met. It employs early stopping as its stopping criterion to effectively mitigate overfitting without compromising the accuracy of the algorithm [40–42].
- To address the overfitting issue, the difference between test accuracy and training accuracy is calculated. This difference should be minimal (less than 3% for example). If it exceeds this threshold, the training process is halted.

Figure 1 provides a graphical illustration of the proposed algorithm depicting a given possible situation.

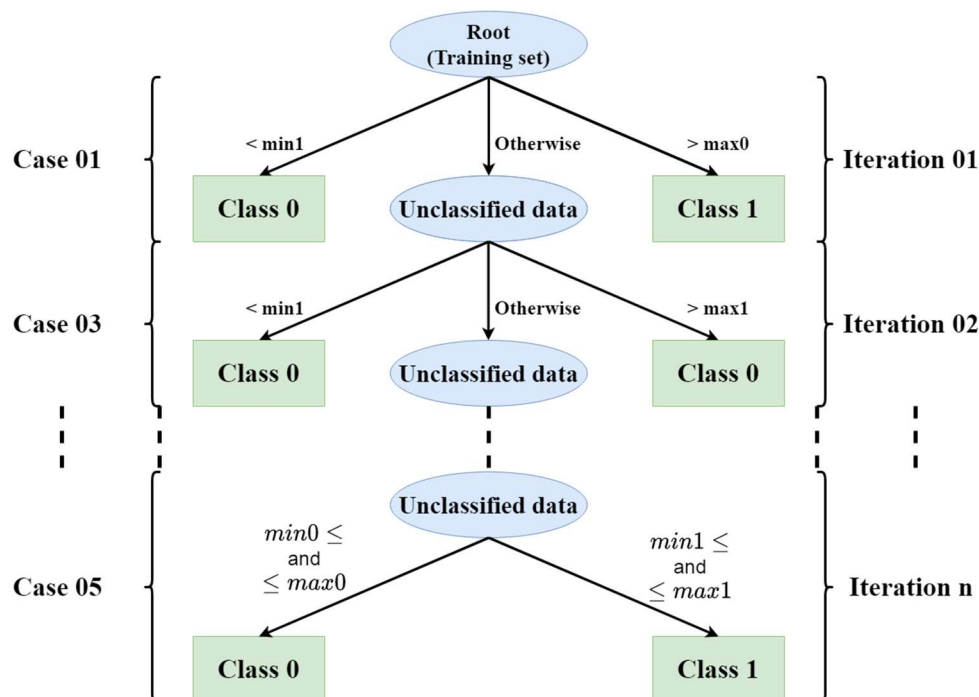


Figure 1. Flowchart of the proposed algorithm.

The flowchart of the algorithm is given in Figure 2.

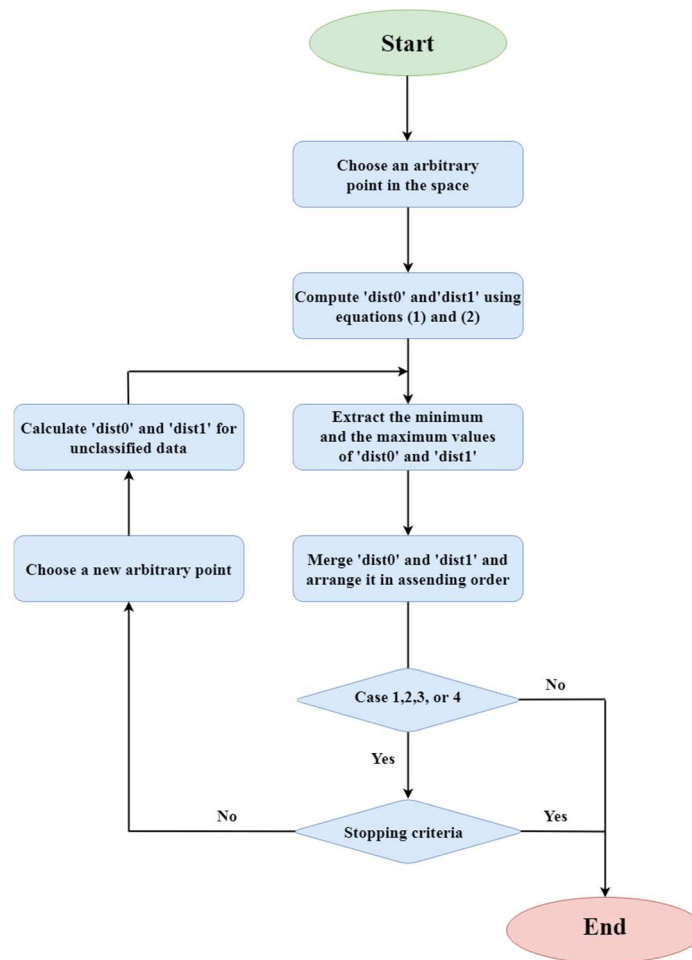


Figure 2. Flowchart of the proposed algorithm.

The pseudo code of the proposed algorithm is given below:

Algorithm 1 Pseudo code of the Euclidean Distance-Based Tree

Algorithm

- 1: **STEP (a):** Generate a random point.
- 2: **STEP (b):** Using (eq.1) and (eq.2), calculate the distances $dist^0$ and $dist^1, (i = 1, 2, \dots, n, j = 1, 2, \dots, m)$.
- 3: **STEP (c):** Find $min_0, max_0, min_1, max_1$, the minimal and maximal distances of each class.
- 4: **STEP (d):** Store the computed distances into a vector named $dist$ and organize it in ascending order.
- 5: **STEP (e):**
- 6: $c = 1$ (the counter for unclassified data).
- 7: **if** $min_0 < min_1 < max_0 < max_1$ (case 1) **then**
- 8: **for** $i = 1$ to k ($k = n + m$) **do**
- 9: **if** $min_0 \leq dist(i) < min_1$ **then**
- 10: The point associated to $dist(i)$ belongs to class 0.
- 11: **else if** $max_0 \leq dist(i) < max_1$ **then**
- 12: The point associated to $dist(i)$ belongs to class 1.
- 13: **else if** $(min_1 \leq dist(i) < max_0)$ **then**
- 14: $Unclassified(c :) = trainingset(i :)$

```

15:         Increment c
16:     end if
17: end for
18: trainingset = Unclassified.
19: if the stopping criteria is not verified then
20:     Choose a new arbitrary point.
21:     Calculate the distances  $dist^0_i$  and  $dist^1_j$  for unclassified data.
22:     Go to step (c).
23: else
24:     Go to step (f).
25: end if
26: end if
27: if  $min_1 < min_0 < max_1 < max_0$  (case 2) then
28:     for  $i = 1$  to  $k$  do
29:         if  $min_1 \leq dist(i) < min_0$  then
30:             The point associated to  $dist(i)$  belongs to class 1.
31:         else if  $max_1 \leq dist(i) < max_0$  then
32:             The point associated to  $dist(i)$  belongs to class 0.
33:         else if  $(min_0 \leq dist(i) < max_1)$  then
34:             Unclassified( $c :$ ) = trainingset( $i :$ )
35:             Increment c
36:         end if
37:     end for
38: trainingset = Unclassified.
39: if the stopping criteria is not verified then
40:     Choose a new arbitrary point.
41:     Calculate the distances  $dist^0_i$  and  $dist^1_j$  for unclassified data.
42:     Go to step (c).
43: else
44:     Go to step (f).
45: end if
46: end if
47: if  $min_0 < min_1 < max_1 < max_0$  (case 3) then
48:     for  $i = 1$  to  $k$  do
49:         if  $min_0 \leq dist(i) < min_1$  or  $max_1 \leq dist(i) < max_0$  then
50:             The point associated to  $dist(i)$  belongs to class 0.
51:         else if  $(min_1 \leq dist(i) < max_1)$  then
52:             Unclassified( $c :$ ) = trainingset( $i :$ )
53:             Increment c
54:         end if
55:     end for
56: trainingset = Unclassified.
57: if the stopping criteria is not verified then
58:     Choose a new arbitrary point.
59:     Calculate the distances  $dist^0_i$  and  $dist^1_j$  for unclassified data.
60:     Go to step (c).
61: else
62:     Go to step (f).
63: end if
64: end if
65: if  $min_1 < min_0 < max_0 < max_1$  (case 4) then
66:     for  $doi = 1$  to  $k$ .
67:         if  $min_1 \leq dist(i) < min_0$  or  $max_0 \leq dist(i) < max_1$  then
68:             The point associated to  $dist(i)$  belongs to class 1.
69:         else if  $(min_0 \leq dist(i) < max_0)$  then
70:             Unclassified( $c :$ ) = trainingset( $i :$ )
71:             Increment c
72:         end if
73:     end for
74: trainingset = Unclassified.

```

```
75:   if the stopping criteria is not verified then
76:       Choose a new arbitrary point.
77:       Calculate the distances  $dist^0_i$  and  $dist^1_j$  for unclassified data.
78:       Go to step (c).
79:   else
80:       Go to step (f).
81:   end if
82: end if
83: if  $min_0 < max_0 < min_1 < max_1$  or  $min_1 < max_1 < min_0 < max_0$  (case 5) then
84:     for  $doi = 1$  to  $k$ .
85:         if  $min \leq dist(i) \leq max_0$  then
86:             The point associated to  $dist(i)$  belongs to class 0.
87:         end if
88:         if  $min \leq dist(i) \leq max_1$  then
89:             The point associated to  $dist(i)$  belongs to class 1.
90:         end if
91:     end for
92:     Go to step (f).
93: end if
94: Step (f): End(all data is classified or the stopping criterion is met).
```

3. Dataset Description

The PV array used to generate the dataset, for both healthy and faulty states, consists of two parallel strings. Each string comprises fifteen series-connected Isofoton PVM (106W-12V).The Simulink/MATLAB platform is utilized to simulate the current (Imp_{pp}) and voltage (Vm_{pp}) at the maximum power point of this PV array under both healthy and faulty states, considering various values of cell temperature (T) and irradiance (G). Through this simulation, 753 samples are generated for each of the considered classes, consisting of the four physical quantities (T, G, Imp_{pp}, Vm_{pp}). In this study, besides the normal operating state, six faulty states are considered. These states and their corresponding labels are given in Table 1.

Table 1. Operating states and their labels.

Class name	Label
Normal operation	Class 0
Short circuit of three modules	Class 1
Short circuit of ten modules	Class 2
String disconnection	Class 3
String disconnection with 25% of partial shading	Class 4
String disconnection with 50% of partial shading	Class 5
String disconnection with 75% of partial shading	Class 6

As shown in Figure 3, utilizing the Imp_{pp} as a feature makes it possible to distinguish ²¹⁸ between three faults: string disconnection, string disconnection with 50% shading, and ²¹⁹ string disconnection with 75% shading. Meanwhile, in Figure 4, it appears that the Vm_{pp} ²²⁰ feature can be used to classify faults such as string disconnection with 25% shading, short ²²¹ circuits of three modules, and short circuits of ten modules. To detect the healthy state class, ²²² both Imp_{pp} and Vm_{pp} features must be used simultaneously.

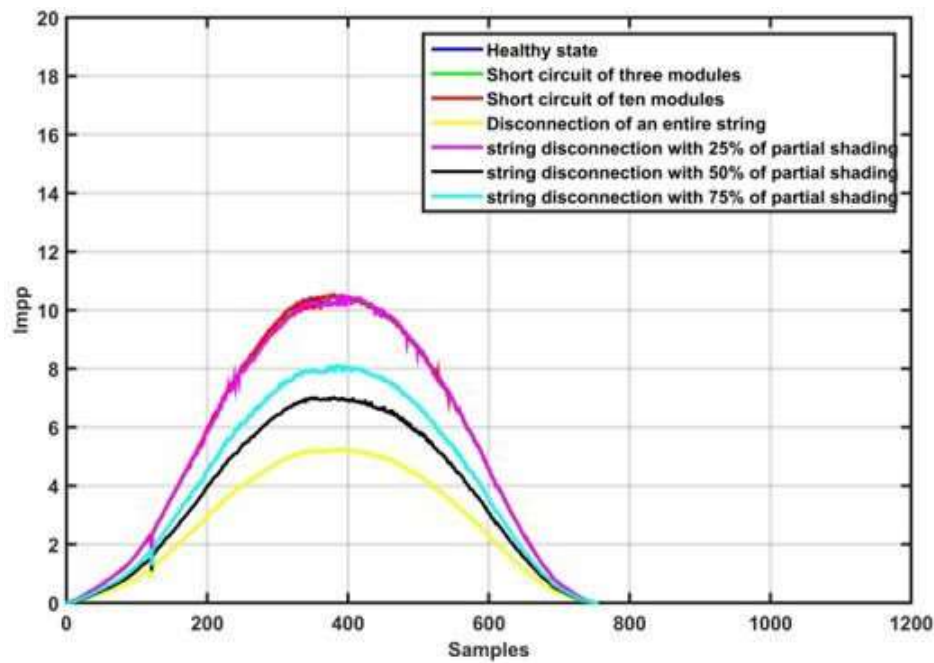


Figure 3. I_{mpp} for various operating states of the PVA.

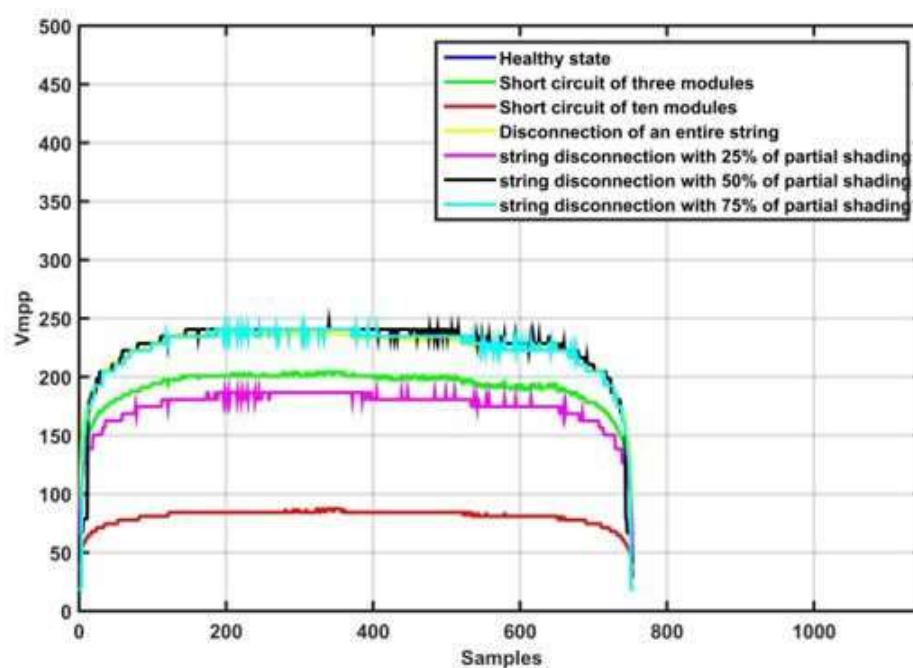


Figure 4. V_{mpp} for various operating states of the PVA.

4. Fault detection and diagnosis methodology

The flowchart of the classification strategy used is shown in Figure 5. In order for the algorithm to function effectively, the multi-class dataset must be adapted to a bi-class dataset by isolating one class at a time, starting from class 0 and going up to class 6. Therefore, six classifiers must be designed for this purpose. The first, the second and the third classifiers isolate classes 0, 1 and 2 from the rest of the classes, respectively. Then, the fourth classifier isolates class 4 from classes 3, 5, and 6. The fifth classifier separates class 3 from classes 5 and 6. Finally, the sixth classifier distinguishes between classes 5 and 6. Each classifier is designed based on the classification algorithm described previously and uses the four specified features (T , G , I_{mpp} , and V_{mpp}).

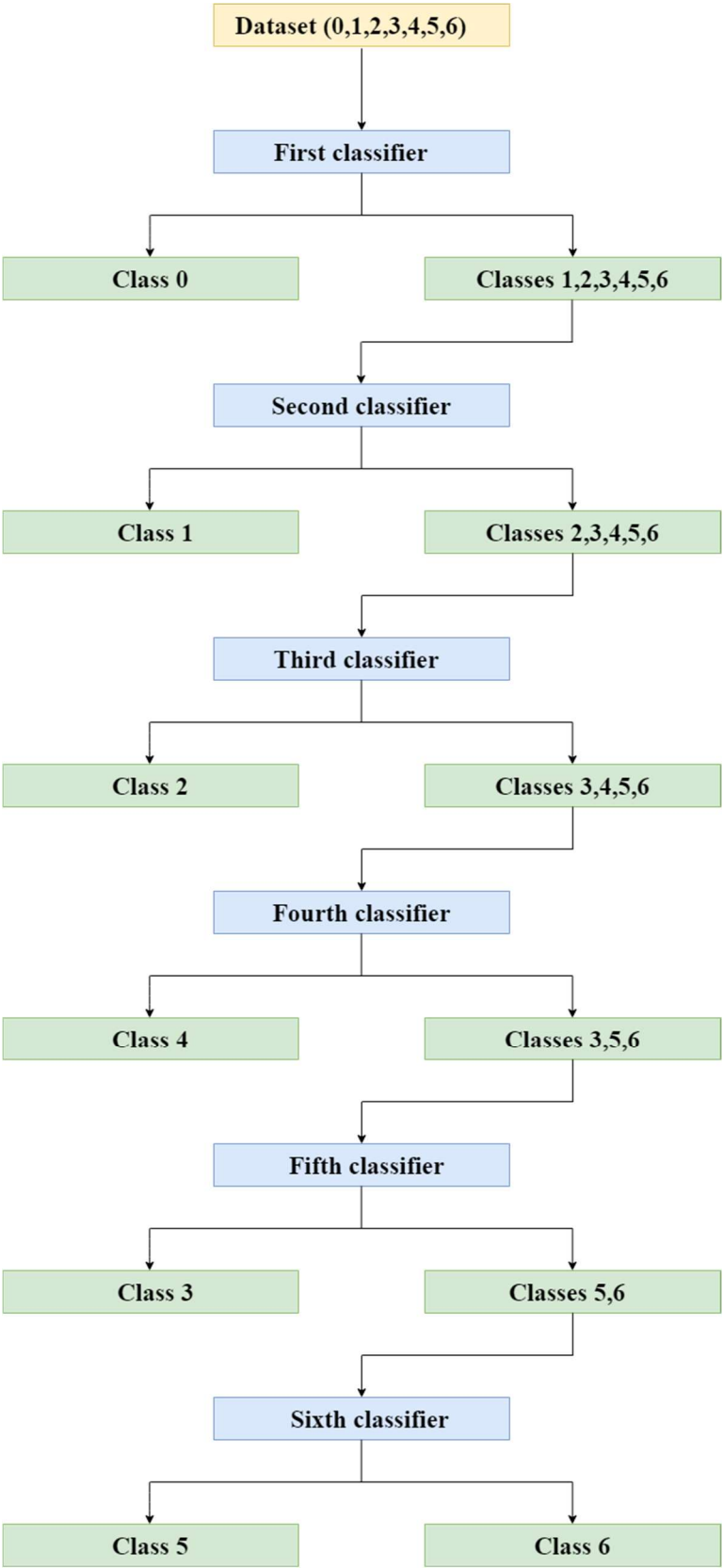


Figure 5. Fault detection and diagnosis flowchart.

5. Results and discussion

The confusion matrix is a well-known and important mathematical tool in the field of machine learning for evaluating algorithms. The elements of this matrix play a role in calculating the

accuracy, precision and recall metrics. This matrix has two rows and two columns, as illustrated in Table 2.

Table 2. Confusion matrix.

Real class labels	Predicted label: A	Predicted label: B
Class A	TP	FN
Class B	FP	TN

TP: stands for True Positive, which means the number of data that are in class 0 in the dataset and the algorithm successfully considered them in class 0. **FN:** stands for False Negative, which means the number of data that are in class 0 in the dataset and the algorithm considered them in class 1. **FP:** stands for False Positive, which means the number of data that are in class 1 in the dataset and the algorithm considered them in class 0. **TN:** stands for True Negative, which means the number of data that are in class 1 in the dataset and the algorithm successfully considered them in class 1. The three metrics are computed as follow:.

- The accuracy: is a metric that shows how many data, the algorithm correctly classifies them and it's given by equation

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \times 100 \quad (7)$$

- The precision: It measures the proportion of correctly predicted positive data to the total data predicted as positive. It is given by:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100 \quad (8)$$

- The recall: is a metric that shows how many data that are really in class 1 and the classifier correctly predicted them in class 1. It is given by:

$$\text{Recall} = \frac{TP}{TP + FN} \times 100 \quad (9)$$

5.1. Training the Fault Detection and Diagnosis Model Using the Proposed Algorithm

Like any other statistical learning algorithm, the proposed algorithm firstly needs to be trained using a training dataset. Following training, its performance is evaluated using a separate testing set. The dataset is partitioned into two subsets: the training set comprises 87% of the global dataset, while the testing set encompasses 13% of the global dataset. As mentioned earlier, six classifiers are necessary to detect and diagnose the specified faults. To mitigate overfitting effectively without compromising the algorithm's accuracy, the early stopping criterion is employed to stop the training process of each classifier.

The accuracy metric for each classifier is calculated at every iteration and illustrated in Figures 6 through 11. As can be seen, for all classifiers the accuracy value increases over iterations. Classifiers 1 to 6 of the trained model require 23, 9, 4, 6, 16, and 17 steps, respectively, to separate a class from the other classes.

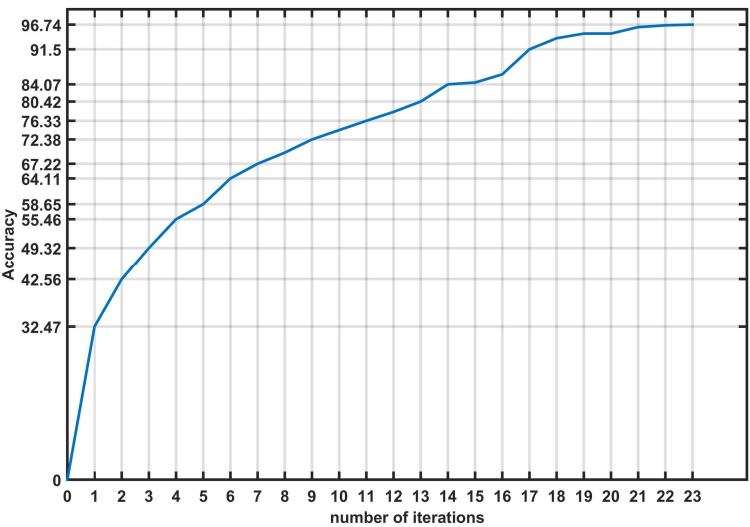


Figure 6. Evolution of accuracy for the first classifier.

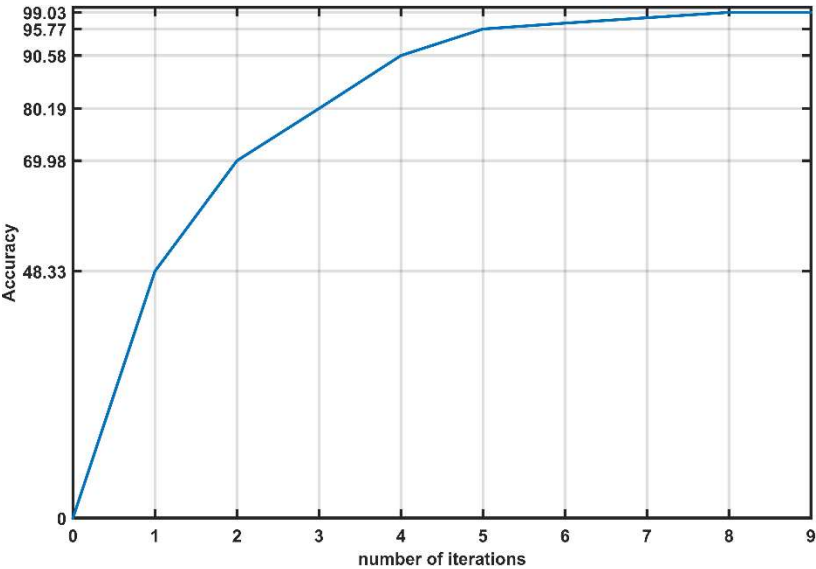


Figure 7. Evolution of accuracy for the second classifier.

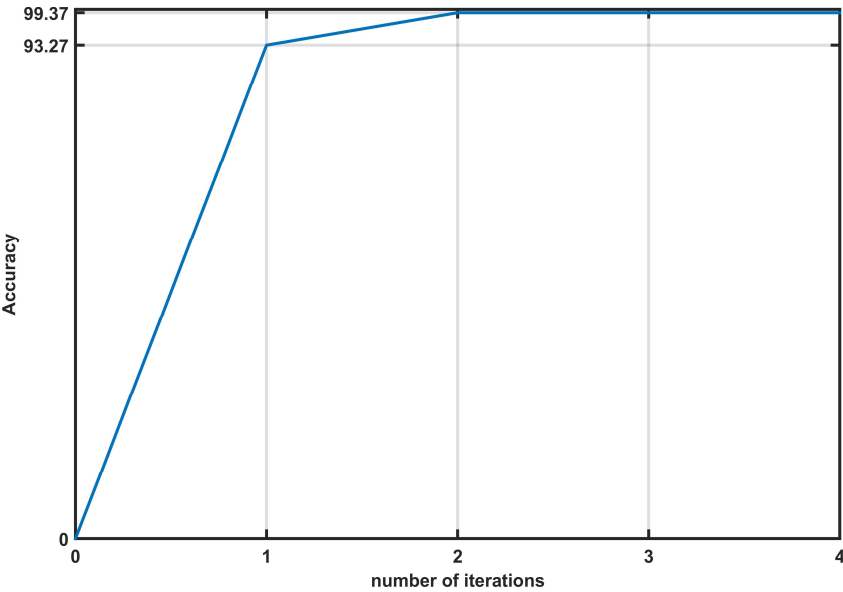


Figure 8. Evolution of accuracy for the third classifier.

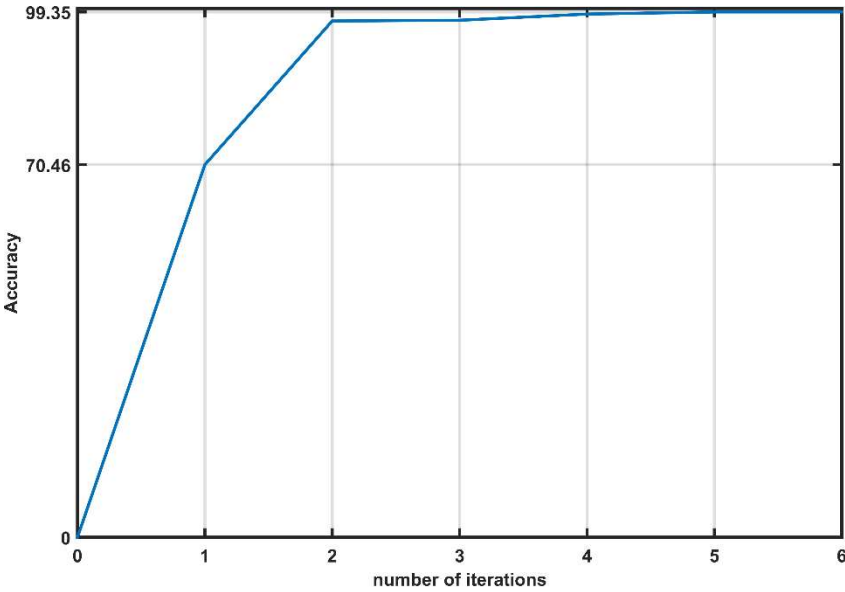


Figure 9. Evolution of accuracy for the forthclassifier.

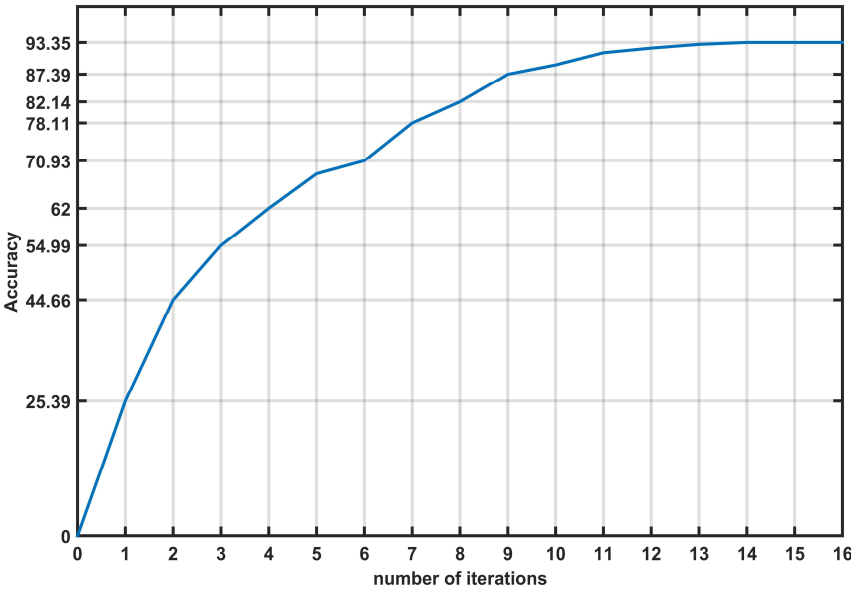


Figure 10. Evolution of accuracy for the fifth classifier.

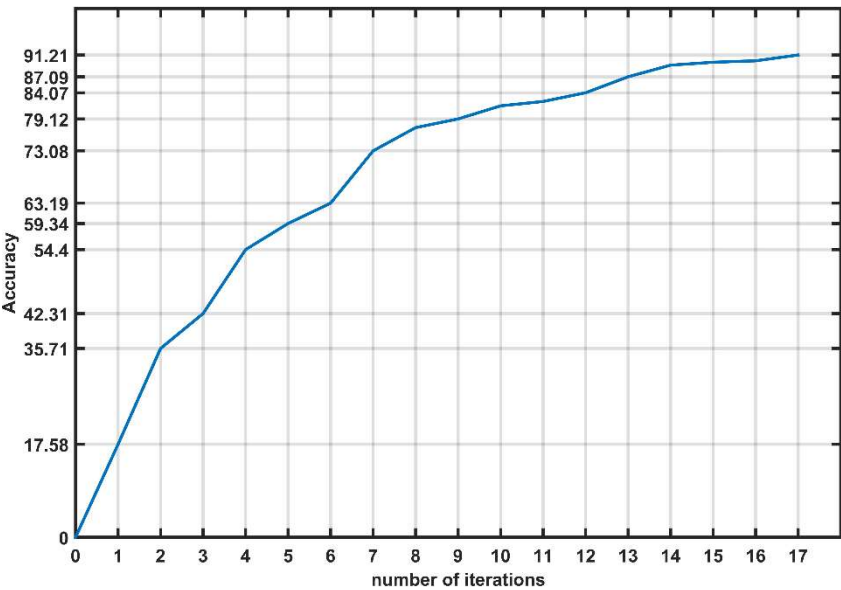


Figure 11. Evolution of accuracy for the sixth classifier.

5.2. Evaluating the Performance of the Obtained Model Using the Proposed Algorithm

The performance of the proposed approach is evaluated using the average values of precision, precision, and recall. The higher these values, the better the performance of the proposed approach is and vice versa. The confusion matrices and values of the three metrics are calculated from the test dataset. The results are presented in Table 3 and Table 4, respectively.

Table 3 presents the confusion matrix values for each classifier within the resulting model. These values were used to calculate the precision, accuracy, and recall measures for all six classifiers and are presented in Table 4. The last row of Table 4 shows the average values of the three measures, which represent the measures of the resulting model.

Table 3. Confusion matrices for the obtained model.

	TP	FN	FP	TN
Classifier 1	1107	19	29	168
Classifier 2	947	3	4	178
Classifier 3	762	1	3	183
Classifier 4	570	3	1	190
Classifier 5	354	13	10	179
Classifier 6	177	20	5	155

Table 4. Metrics values for the obtained model.

	Accuracy (%)	Precision (%)	Recall (%)
Classifier 1	97	97	98
Classifier 2	99	100	100
Classifier 3	100	100	100
Classifier 4	99	100	99
Classifier 5	96	97	99
Classifier 6	93	97	90
Average values	97.33	99	97

5.3. Comparative Study using Various Machine Learning Algorithms

In this comparative study, the fault detection and diagnosis model depicted in the flowchart of Figure 5 is constructed using various statistical methods, namely the SVM algorithm [26], the DT algorithm [7,30], the RF algorithm [31–33], and the KNN algorithm [34–36].

The confusion matrices for the obtained model using the aforementioned algorithms are provided in Table 5, while Table 6 presents the values for accuracy, precision, and recall, along with the average values of these metrics.

Table 5. Confusion matrices for the obtained model using the four algorithms.

		SVM	DT	RF	KNN
Classifier 1	TP	1104	113	1119	1107
	FN	15	6	8	12
	FP	165	23	15	155
	TN	34	176	176	44
Classifier 2	TP	1086	950	950	1078
	FN	0	3	1	1
	FP	122	3	1	1
	TN	61	180	180	182
Classifier 3	TP	1021	765	764	892
	FN	0	2	0	0
	FP	103	0	1	0
	TN	84	186	186	187
Classifier 4	TP	927	566	568	696
	FN	0	3	1	0
	FP	107	0	0	0
	TN	90	196	196	196
Classifier 5	TP	796	364	371	508
	FN	57	10	4	1
	FP	131	7	3	167
	TN	50	185	183	157
Classifier 6	TP	631	29	36	105
	FN	112	172	175	73
	FP	84	202	202	201
	TN	90		102	

Table 6. Metrics values (in percentage) for the obtained model using the four algorithms.

		SVM	DT	RF	KNN
Classifier 1	Accuracy	86	98	99	87(K=7)
	Precision	87	98	99	88
	Recall	99	99	100	99
Classifier 2	Accuracy	90	99	100	100(K=1)
	Precision	90	100	100	100
	Recall	100	100	100	100
Classifier 3	Accuracy	91	100	100	100(K=3)
	Precision	91	100	100	100
	Recall	100	100	100	100
Classifier 4	Accuracy	90	100	100	100(K=2)
	Precision	90	100	100	100
	Recall	100	99	100	100
Classifier 5	Accuracy	82	97	99	76(K=35)
	Precision	86	98	99	75

	Recall	93	97	99	100
Classifier 6	Accuracy	78	64	63	59(K=1)
	Precision	88	87	85	80
	Recall	84	54	53	60
Average values Accuracy		85.33	97.16	93.50	87
Precision		88.65	91.50	97.16	90.50
Recall		96	93.50	92	93.16

Table 7 collects all the average values of the three metrics for each of the proposed ²⁸⁴ algorithm as well as the SVM, the DT, the RF and the KNN algorithms. From the table, it ²⁸⁵ can be seen that the performance of the proposed algorithm is superior to the rest of the ²⁸⁶ algorithms in term of accuracy, precision and recall.

Table 7. Metrics average values.

	Accuracy (%)	Precision (%)	Recall (%)
The proposed algorithm	97.33	98.66	97.5
SVM	85.33	88.65	96
DT	93	97.16	91.50
RF	93.50	97.16	92
KNN	87	90.50	93.16

Figures 12 to 16 display the fault detection and diagnosis results using the proposed algorithm-based model and those based on the SVM, DT, RF, and KNN algorithms, respectively. It can be seen from these figures that the smallest number of incorrectly classified data is obtained in the case of both the RF algorithm-based model and the proposed algorithm-based model. The models fail to correctly classify all data due to data overlap and overfitting issues.

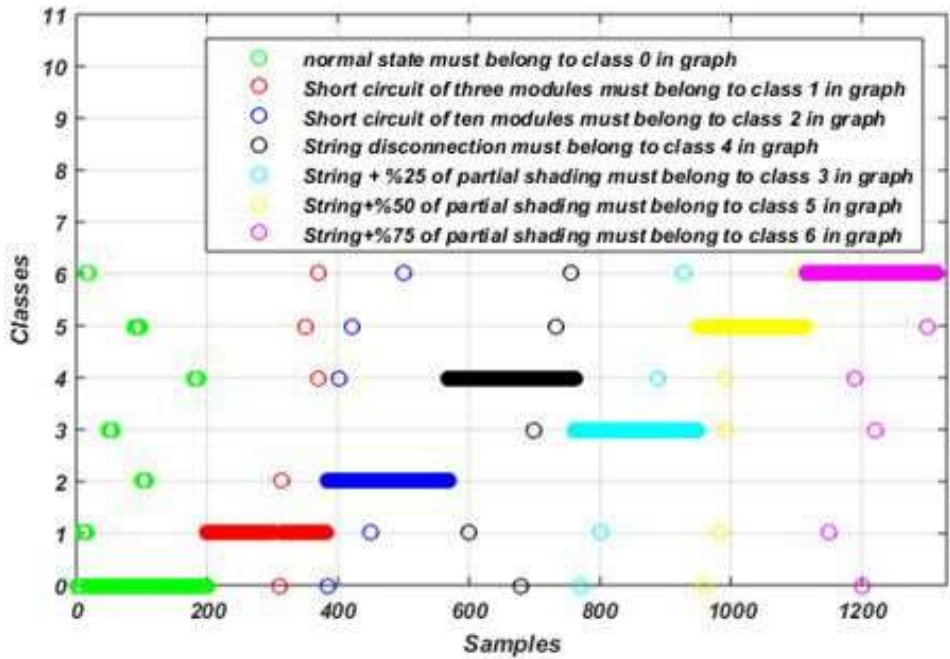


Figure 12. Fault detection and diagnosis results using the proposed algorithm-based model.

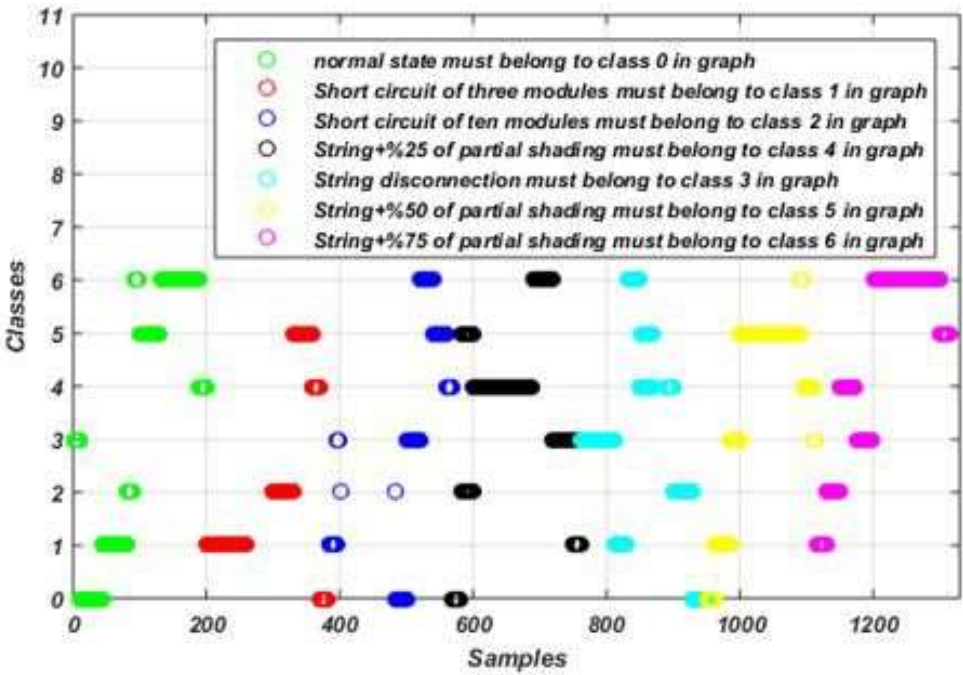


Figure 13. Fault detection and diagnosis results using the SVM algorithm-based model.

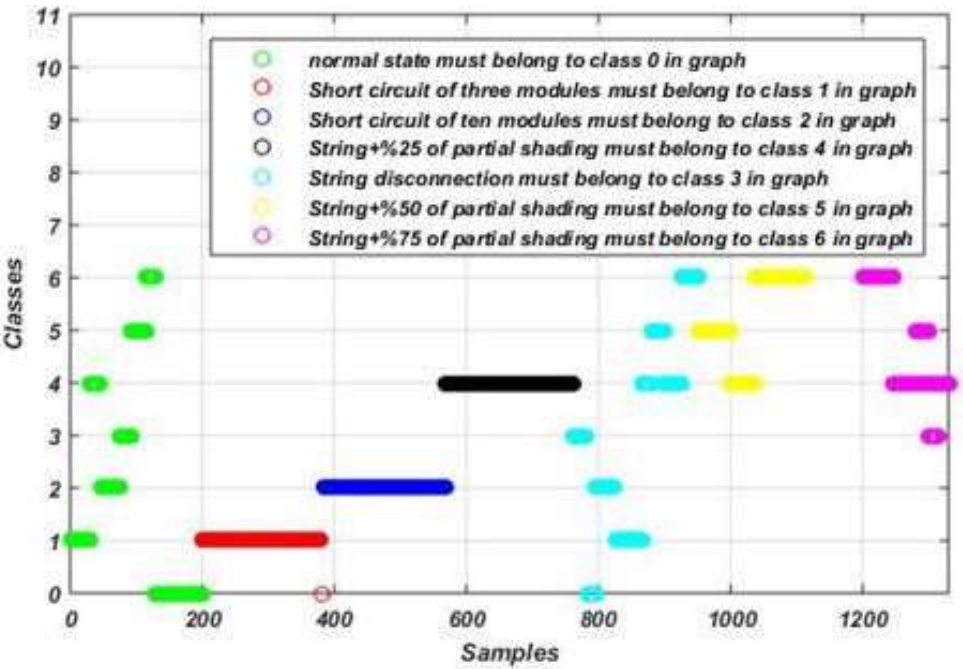


Figure 14. Fault detection and diagnosis results using the DT algorithm-based model.

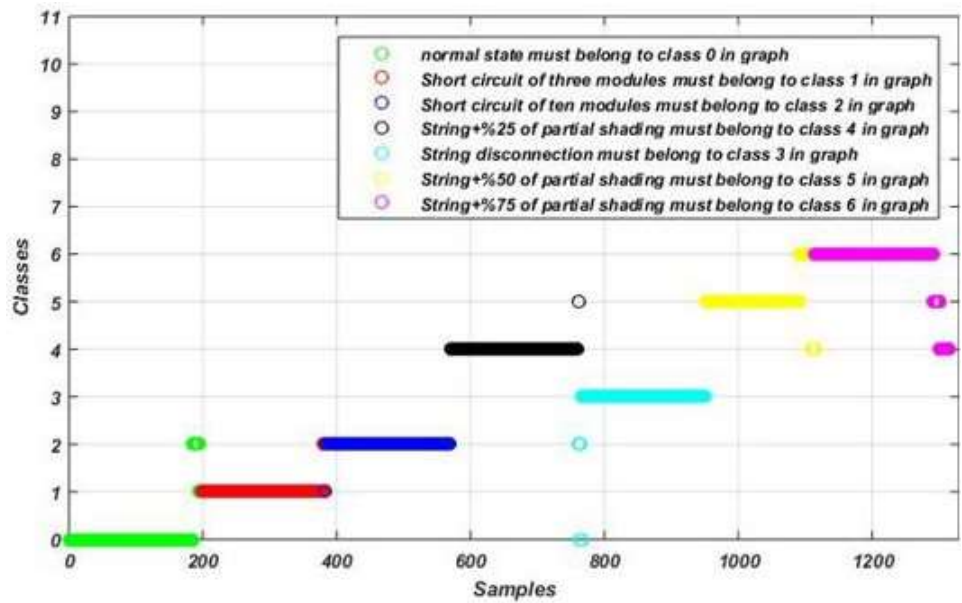


Figure 15. Fault detection and diagnosis results using the RF algorithm-based model.

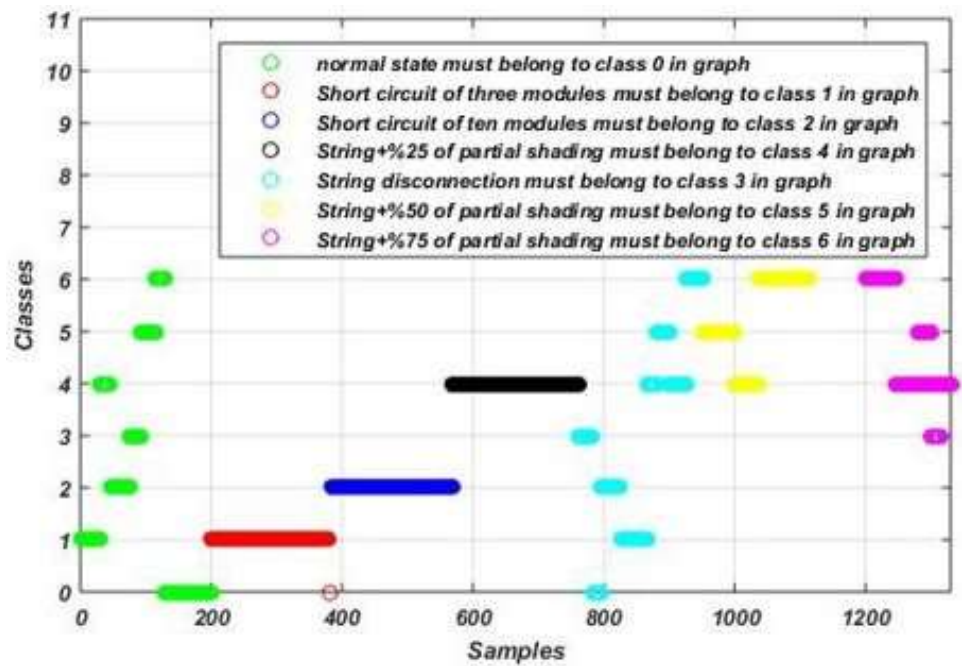


Figure 16. Fault detection and diagnosis results using the KNN algorithm-based model.

6. Conclusion

In this work, an enhanced approach was proposed for identifying and diagnosing PV array faults. A comparative study was conducted between the proposed algorithm based model and models based on four statistical learning algorithms: SVM, DT, RF, and KNN. Unlike the decision tree algorithm, which uses the Gini index to split the data onto two classes, the proposed algorithm calculates Euclidean distances between an arbitrary point and the dataset samples. It then utilizes the minimal and maximal distances to separate the samples belonging to each class.

In this study, four features, namely: cell temperature, irradiance, current and voltage of the maximum power point were utilized. The proposed methodology effectively distinguishes the normal operating condition from other abnormal states, achieving a classification accuracy of 97%. The comparative investigation demonstrated that the proposed approach outperformed the other

methods considered in this work in terms of accuracy, precision, and recall. By increasing the number of classifiers, the proposed technique can be easily extended to encompass additional faults.

Author Contributions: Conceptualization, Y.M.; methodology, K.K and A.C.; software, Y.M and A.A.; validation, Y.M, K.K. and A.C.; formal analysis, K.K.; investigation, K.K. and A.A.; resources, Y.M., K.K and S.S.; data curation, Y.M.; writing—original draft preparation, Y.M.; writing—review and editing, K.K., A.C. and S.S.; visualization, Y.M. and A.A.; supervision, K.K., A.C. and S.S.; project administration, K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data is not available.

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

Abbreviations

The following abbreviations are used in this manuscript:

ADAG	Adaptive directed acyclic graph
ANN	Artificial neural network
DDAG	Decision directed acyclic graph
DT	Decision Tree
FDD	Fault Detection and Diagnosis
FN	False Negative
FP	False Positive
G	Irradiance
GCPV	Grid connected photovoltaic
Imp _p	Current at the maximum power point
I _{sc}	Current of short circuit
KNN	K Nearest Neighbors
OVA	One vs.all
PV	Photovoltaic
PVM	Photovoltaic module
PVS	Photovoltaic system
RF	Random Forest
SVM	Support Vector Machine
T	Temperature
TDR	Time domain reflectometry
TN	True Negative
TP	True Positive
V _{mp_p}	Voltage at the maximum power point
V _{oc}	Voltage of open circuit

321

References

1. Sohani, A.; Sayyaadi, H.; Cornaro, C.; Shahverdian, M.; Pierro, M.; Moser, D.; Karimi, N.; Doranehgard, M.; Li, L.K. Using machine learning in photovoltaics to create smarter and cleaner energy generation systems: A comprehensive review. *Journal of Cleaner Production* **2022**, p. 132701.
2. Mughal, S.; Sood, Y.R.; Jarial, R. A review on solar photovoltaic technology and future trends. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* **2018**, *4*, 227–235.

3. Madeti, S.R.; Singh, S. A comprehensive study on different types of faults and detection techniques for solar photovoltaic system. *327 Solar Energy* **2017**, *158*, 161–185.
4. Hernandez, J.; Velasco, D.; Trujillo, C. Analysis of the effect of the implementation of photovoltaic systems like option of distributed generation in Colombia. *Renewable and sustainable energy reviews* **2011**, *15*, 2290–2298.
5. Qais, M.H.; Hasanien, H.M.; Alghuwainem, S.; Nouh, A.S. Coyote optimization algorithm for parameters extraction of three-diode photovoltaic models of photovoltaic modules. *Energy* **2019**, *187*, 116001.
6. Kumar, B.P.; Ilango, G.S.; Reddy, M.J.B.; Chilakapati, N. Online fault detection and diagnosis in photovoltaic systems using wavelet packets. *IEEE Journal of Photovoltaics* **2017**, *8*, 257–265.
7. Benkercha, R.; Moulahoum, S. Fault detection and diagnosis based on C4. 5 decision tree algorithm for grid connected PV system. *335 Solar Energy* **2018**, *173*, 610–634.
8. Villarini, M.; Cesarotti, V.; Alfonsi, L.; Introna, V. Optimization of photovoltaic maintenance plan by means of a FMEA approach based on real data. *Energy Conversion and Management* **2017**, *152*, 1–12.
9. Pillai, D.S.; Rajasekar, N. A comprehensive review on protection challenges and fault diagnosis in PV systems. *Renewable and Sustainable Energy Reviews* **2018**, *91*, 18–40.
10. Zhao, Q.; Shao, S.; Lu, L.; Liu, X.; Zhu, H. A new PV array fault diagnosis method using fuzzy C-mean clustering and fuzzy membership algorithm. *Energies* **2018**, *11*, 238.
11. Hazra, A.; Das, S.; Basu, M. An efficient fault diagnosis method for PV systems following string current. *Journal of Cleaner Production* **2017**, *154*, 220–232.
12. Khelil, C.K.M.; Amrouche, B.; soufiane Benyoucef, A.; Kara, K.; Chouder, A. New intelligent fault diagnosis (IFD) approach for grid-connected photovoltaic systems. *Energy* **2020**, *211*, 118591.
13. Khelil, C.K.M.; Amrouche, B.; Kara, K.; Chouder, A. The impact of the ANN's choice on PV systems diagnosis quality. *Energy Conversion and Management* **2021**, *240*, 114278.
14. Mellit, A.; Tina, G.M.; Kalogirou, S.A. Fault detection and diagnosis methods for photovoltaic systems: A review. *Renewable and Sustainable Energy Reviews* **2018**, *91*, 1–17.
15. Takashima, T.; Yamaguchi, J.; Otani, K.; Kato, K.; Ishida, M. Experimental studies of failure detection methods in PV module strings. In Proceedings of the 2006 IEEE 4th World Conference on Photovoltaic Energy Conference. IEEE, 2006, Vol. 2, pp. 2227–2230.
16. Takashima, T.; Yamaguchi, J.; Ishida, M. Fault detection by signal response in PV module strings. In Proceedings of the 2008 33rd IEEE Photovoltaic Specialists Conference. IEEE, 2008, pp. 1–5.
17. Takashima, T.; Yamaguchi, J.; Otani, K.; Oozeki, T.; Kato, K.; Ishida, M. Experimental studies of fault location in PV module strings. *Solar Energy Materials and Solar Cells* **2009**, *93*, 1079–1082.
18. Takashima, T.; Yamaguchi, J.; Ishida, M. Disconnection detection using earth capacitance measurement in photovoltaic module string. *Progress in Photovoltaics: Research and Applications* **2008**, *16*, 669–677.
19. Chouder, A.; Silvestre, S. Automatic supervision and fault detection of PV systems based on power losses analysis. *Energy conversion and Management* **2010**, *51*, 1929–1937.
20. Silvestre, S.; Chouder, A.; Karatepe, E. Automatic fault detection in grid connected PV systems. *Solar energy* **2013**, *94*, 119–127.
21. Spataru, S.; Sera, D.; Kerekes, T.; Teodorescu, R. Photovoltaic array condition monitoring based on online regression of performance model. In Proceedings of the 2013 IEEE 39th Photovoltaic Specialists Conference (PVSC). IEEE, 2013, pp. 0815–0820.
22. Drews, A.; De Keizer, A.; Beyer, H.G.; Lorenz, E.; Betcke, J.; Van Sark, W.; Heydenreich, W.; Wiemken, E.; Stettler, S.; Toggweiler, P.; et al. Monitoring and remote failure detection of grid-connected PV systems based on satellite observations. *Solar energy* **2007**, *81*, 548–564.
23. Bastidas-Rodriguez, J.D.; Franco, E.; Petrone, G.; Ramos-Paja, C.A.; Spagnuolo, G. Quantification of photovoltaic module degradation using model based indicators. *Mathematics and Computers in Simulation* **2017**, *131*, 101–113.
24. Dhoke, A.; Sharma, R.; Saha, T.K. An approach for fault detection and location in solar PV systems. *Solar Energy* **2019**, *194*, 197–208. 370

25. Mandal, R.K.; Kale, P.G. Assessment of different multiclass SVM strategies for fault classification in a PV system. In Proceedings of the Proceedings of the 7th International Conference on Advances in Energy Research. Springer, 2021, pp. 747–756.
26. Cho, K.H.; Jo, H.C.; Kim, E.S.; Park, H.A.; Park, J.H. Failure diagnosis method of photovoltaic generator using support vector machine. *Journal of Electrical Engineering & Technology* **2020**, *15*, 1669–1680.
27. Chen, L.; Lin, P.; Zhang, J.; Chen, Z.; Lin, Y.; Wu, L.; Cheng, S. Fault diagnosis and classification for photovoltaic arrays based on principal component analysis and support vector machine. In Proceedings of the IOP Conference series: Earth and Environmental science. IOP Publishing, 2018, Vol. 188, p. 012089.
28. Wang, J.; Gao, D.; Zhu, S.; Wang, S.; Liu, H. Fault diagnosis method of photovoltaic array based on support vector machine. *Energy sources, part a: recovery, utilization, and environmental effects* **2019**, pp. 1–16.
29. Yi, Z.; Etemadi, A.H. Line-to-line fault detection for photovoltaic arrays based on multiresolution signal decomposition and two-stage support vector machine. *IEEE Transactions on Industrial Electronics* **2017**, *64*, 8546–8556.
30. Dhibi, K.; Mansouri, M.; Bouzrara, K.; Nounou, H.; Nounou, M. An enhanced ensemble learning-based fault detection and diagnosis for grid-connected PV systems. *IEEE Access* **2021**, *9*, 155622–155633.
31. Chen, Z.; Han, F.; Wu, L.; Yu, J.; Cheng, S.; Lin, P.; Chen, H. Random forest based intelligent fault diagnosis for PV arrays using array voltage and string currents. *Energy conversion and management* **2018**, *178*, 250–264.
32. Dhibi, K.; Fezai, R.; Mansouri, M.; Trabelsi, M.; Kouadri, A.; Bouzara, K.; Nounou, H.; Nounou, M. Reduced kernel random forest technique for fault detection and classification in grid-tied PV systems. *IEEE Journal of Photovoltaics* **2020**, *10*, 1864–1871.
33. Dhibi, K.; Fezai, R.; Bouzrara, K.; Mansouri, M.; Nounou, H.; Nounou, M.; Trabelsi, M. Enhanced RF for Fault Detection and Diagnosis of Uncertain PV systems. In Proceedings of the 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD). IEEE, 2021, pp. 103–108.
34. Wang, L.; Qiu, H.; Yang, P.; Gao, J. Fault Diagnosis Method Based on An Improved KNN Algorithm for PV strings. In Proceedings of the 2021 4th Asia Conference on Energy and Electrical Engineering (ACEEE). IEEE, 2021, pp. 91–98.
35. Madeti, S.R.; Singh, S. Modeling of PV system based on experimental data for fault detection using kNN method. *Solar Energy* **2018**, *173*, 139–151.
36. Harrou, F.; Taghezouit, B.; Sun, Y. Improved k NN-based monitoring schemes for detecting faults in PV systems. *IEEE Journal of Photovoltaics* **2019**, *9*, 811–821.
37. Karatepe, E.; Hiyama, T.; et al. Controlling of artificial neural network for fault diagnosis of photovoltaic array. In Proceedings of the 2011 16th International Conference on Intelligent System Applications to Power Systems. IEEE, 2011, pp. 1–6.
38. Chine, W.; Mellit, A.; Lughi, V.; Malek, A.; Sulligoi, G.; Pavan, A.M. A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks. *Renewable Energy* **2016**, *90*, 501–512.
39. Hussain, M.; Dhimish, M.; Titarenko, S.; Mather, P. Artificial neural network based photovoltaic fault detection algorithm integrating two bi-directional input parameters. *Renewable Energy* **2020**, *155*, 1272–1292.
40. Bai, Y.; Yang, E.; Han, B.; Yang, Y.; Li, J.; Mao, Y.; Niu, G.; Liu, T. Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems* **2021**, *34*, 24392–24403.
41. Prechelt, L. Automatic early stopping using cross validation: quantifying the criteria. *Neural networks* **1998**, *11*, 761–767.
42. Zhang, T.; Yu, B. Boosting with early stopping: Convergence and consistency **2005**.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.