Mismatching and partial shading identification in photovoltaic arrays by an artificial neural network ensemble

Michel Piliougine*, Giovanni Spagnuolo

Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata – Università degli Studi di Salerno. Via Giovanni Paolo II, 132. 84084 Fisciano (SA), Italy

ARTICLE INFO

Keywords:
artificial neural network
ensemble neural network
mismatched condition
multilayer perceptron
partial shading
photovoltaic fault detection

ABSTRACT

Photovoltaic arrays may suffer from a number of temporary and permanent faults. Partial shading and soiling belong to the former group, while cell cracking and delamination fall within the latter one. In these cases, the shape of the current vs voltage curve around the maximum power point shows features that are different from those ones of an array operating in normal conditions. The shape change should allow triggering, in case of a temporary fault, control actions that might improve the electrical power production. Instead, if the fault is permanent, the identification of the modified shape should activate a procedure for a more in–depth analysis of the problem and a maintenance action. In this paper, the conditions leading to a change in the array behavior during its delivering of the maximum power are examined. The change of curvature of the current vs voltage curve around its maximum power point is suitably detected to trigger the fault mitigation action. The feature is caught through an ensemble of artificial neural networks, which analyzes the current vs voltage curve and classifies the module as healthy or faulty. It is demonstrated that few samples around the maximum power point are required, this meaning that the proposed approach is compatible with the operation of any perturbative maximum power point tracking algorithm and its application does not lead to any power production drop. In addition, the approach does not require neither temperature nor irradiance measurements as inputs. The neural networks are trained through synthetic data, so that their application is not limited to arrays including a specific photovoltaic module. The method is also validated through experimental data.

*Corresponding author
Email addresses: mpiliouginerocha@unisa.it (M. Piliougine); gspagnuolo@unisa.it (G. Spagnuolo)

1. Introduction

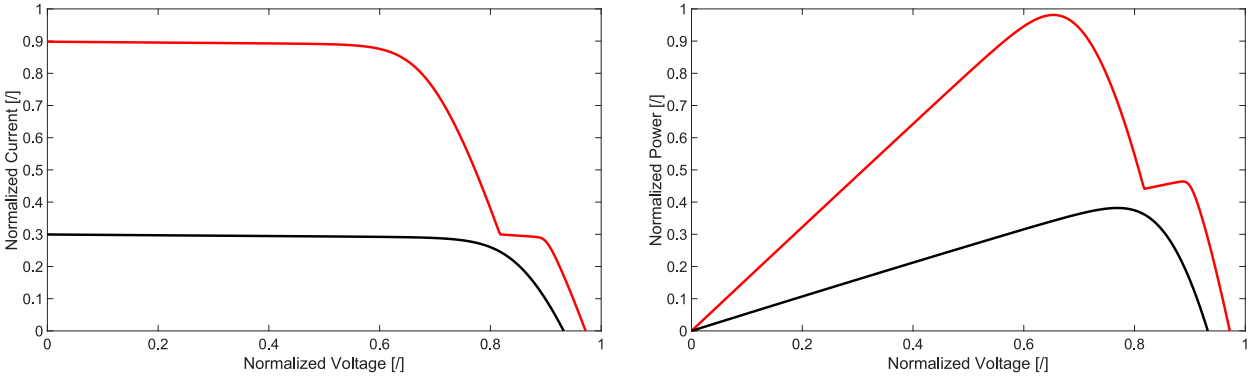
Photovoltaic (PV) technology is a key one for ensuring electrical energy at a low environmental impact. Its wider adoption will be crucial to meet the objectives of the EU Green Deal [1] and Recovery Plan [2]. In terms of global PV capacity, the amount of 708 GW was reached at the end of the year 2020 [3]. Moreover, the worldwide solar electrical energy produced in 2021 has grown around 145 TWh, which is close to the +18% with respect to 2020. Consequently, the final value at the end of 2021 was around to 1 000 TWh [4].

The maximization of the PV power production requires suitable algorithms, circuits and systems aimed at monitoring the plant and performing diagnostic and maintenance actions. Possible faults might be classified into *temporary* and *permanent* ones. Those ones belonging to the first category determine a reduction of the module power for a limited amount of time. Partial shading and soiling fall into this group: a proper control action might minimize their detrimental effect on the electrical production. Permanent faults, instead, must be detected promptly in order to trigger a more in–depth investigation of the source of the fault and suitable maintenance operations.

Some of the faults falling into both categories determine a change of the shape of the PV generator Current vs. Voltage (*I–V*) and Power vs. Voltage (*P–V*) curves around the Maximum Power Point (MPP). The shape of the *I–V* curve around its MPP, especially the one at a high voltage, carries an information that, unfortunately, cannot be deduced by measuring the MPP voltage and current coordinates only. Nevertheless, decrypting this information is useful to detect the occurrence of a partial shading phenomenon or of some types of faults. In the former case, the identification of the curvature of the high voltage MPP is useful to perform the so–called Global MPPT (GMPPT). Indeed, the presence of another MPP, occurring at a completely different voltage value and allowing the string delivers more power, is discovered by commercial inverters by triggering a voltage sweep over their whole operating range. Unfortunately, to determine the optimal time rate at which the sweep has to be run is a very hard task. A too frequent voltage sweep may lead to a significant reduction of the produced energy if the mismatching occurs for a limited time during the day. Instead, a long time between two consecutive sweeps might leave the PV array working on a relative MPP, while a higher power should be delivered if the array is driven towards a different operating voltage.

Many papers afford the GMPPT problem, thus the tracking of the absolute MPP. They are based on algorithms that are much more complicated than classical MPPT, e.g., the Perturb and Observe (P&O) and the Incremental Conductance (INC) ones, and often exploit global optimization capabilities of stochastic methods. These approaches have a good performance when the PV array operates at mismatched conditions, but they are too much involved for uniform operating conditions and also require a hardware that is much more expensive than the one needed for running P&O or INC methods. Unfortunately, papers introducing GMPPT methods do not afford the key issue of detecting the occurrence of the mismatched PV operating conditions. The novel algorithms are often tested by assuming that the instant at which the mismatched conditions take place is known and the GMPPT algorithm starts its operation exactly in that moment. Some approaches trigger the GMPPT operation when the PV array shows a significant power drop. This is not a sufficient condition for stating that the mismatch occurs, because the same drop might be due to a sudden reduction of the irradiance received by the whole PV array, not only by a part of it. The identification of a regular or irregular *I–V* curve shape around the MPP is also useful for the detection of some types of temporary and permanent faults, as a trigger for running fault mitigation strategies and further maintenance actions.

In this paper, the identification of the normal or abnormal local curvature of the *I–V* characteristic around the MPP where the PV generator works is afforded. To the best of the authors' knowledge, the detailed analysis of the mechanism leading to the change of curvature of the *I–V* characteristic around the MPP in case of partial shading has been never proposed in the literature. Instead of concentrating on the shape of the curve in the region around the high voltage MPP, as it is done in this paper, many studies are focused on the analysis of the mechanisms leading to multiple MPPs in *I-V* curves and of the simulation thereof. For instance, in [5], the prediction and assessment of shading on PV systems is performed by accurately simulating the shape of the shadow affecting the array. In [6] a simple method to infer the cloud shadow motion vector from small scale irradiance sensor networks data is proposed. A cloud speed sensor, which is useful to predict the occurrence of shadowing phenomena and their effect on the grid through the abrupt changes of the PV power is also analyzed in [7]. In some studies, e.g. in [8] and [9], the shadowing effect is detected by an ANN based or Machine Learning based analysis of the evolution of the PV string operating voltage, thus not by working on the *I-V* curve as it is done in this paper. Some other approaches, e.g. [10], are based on the elaboration of images showing the PV modules and acquired in the visible spectrum through a camera. A recent approach that is based on the *I-V* curve analysis, as it is for the one presented in this manuscript, has been recently published [11]. The method proposed therein is based on sample-by-sample scan of the curve and it is dedicated to a specific PV module type.



(a) *I–V* curves (b) *P–V* curves

Figure 1: Uniform and mismatched curve comparison

Indeed, its results mainly depend on the noise level affecting the *I-V* curve data and on the values fixed by the user for a number of thresholds used in the algorithm. Instead, the approach presented in this paper is much more general, does not use any PV module specific parameter and processes only few samples of the *I-V* curve located around its MPP occurring at high voltage.

The conclusions drawn from Section 2 are used to design an Artificial Neural Network (ANN) that is able to perform an *on-site* identification of the curvature change, without any prior knowledge of the temperature and irradiance conditions the PV array is working at. The procedure used to prepare the data for the ANN training and allowing to achieve accurate classification results has not been analyzed before in literature. In Section 3, it is shown that the ANN allows to perform this task effectively by employing few samples of the *I–V* curve around the MPP only. This means that the ANN is able to work on the current and voltage measurements that are ordinarily done by any perturbative MPPT algorithm, thus without requiring that the PV generator operating point is moved too far away from the current MPP, with an additional power drop. This feature of the approach proposed in this paper appears to be innovative with respect to the current literature and of great interest in the applications.

In Section 4 the classification results are shown. The ANN validation is done through synthetic data and by using some thousand of experimental *I–V* curves. Finally, taking the best trained networks, an ensemble of ANNs is proposed. Section 5 shows how to apply the ensemble of ANNs to a selection of some challenging curves. The main conclusions end the paper.

2. Causes of MPP region curvature change

Some mechanisms may lead to a macroscopic change of curvature of the *I–V* curve around its MPP. Here below they have been classified into two categories, i.e., partial shading and faults, both temporary and permanent ones.

2.1. Partial shading and mismatching in photovoltaic strings

PV strings are made of modules connected in series, so that the string voltage meets the DC input range of the inverter and, especially, the smaller range in which the inverter performs the MPPT. In presence of non–uniform operating conditions, e.g., shading, some of the bypass diodes each PV module is equipped with turn on and a deep change of the *I–V* and *P–V* curves is produced [12]. At uniform operating conditions, the unique MPP is located at a voltage that is quite close to V_{oc}, usually at the 80% of that V_{oc}. For instance, cases like the one sketched in Fig. 1a may occur. Whereas the black *I–V* curve shows a normalized curve referring to a PV string working under uniform conditions of low irradiance, the red curve represents a case where the 90% of the modules in the string are receiving a high irradiance and the 10% are shadowed, thus having the lower photo–induced current of all the modules in the former string. The red *I–V* curve in Fig. 1a corresponds to the red *P–V* one in Fig. 1b. These plots show the worst case: at a voltage value that is close to the one at which the absolute MPP occurs in the uniform case, i.e., at about the 80% of V_{oc}, a local MPP appears. The latter implies a power that is much lower than the one the PV array might deliver by working in the absolute MPP. A weaker mismatched condition would give an absolute MPP, or a relative MPP having a power

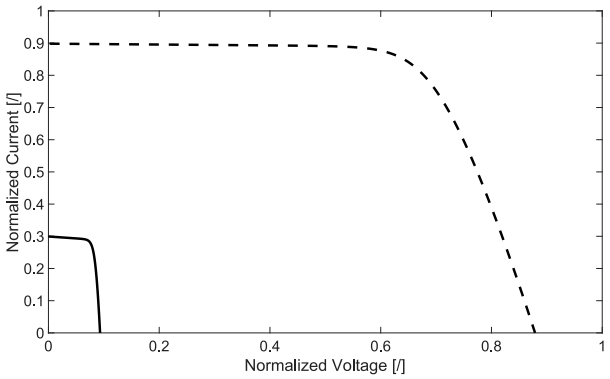


Figure 2: *I–V* curves of the substrings

that is comparable with the one of the absolute MPP, at a similar voltage value. A voltage scan or a more sophisticated algorithm can drive the MPPT controller towards the absolute MPP in the low voltage range. The comparison proposed in Fig. 1a reveals that there is a significant difference in the curvature of the two *I–V* characteristics around the rightmost MPP.

The source of this difference comes out by analyzing Fig. 2, which refers to the mismatched operating conditions of the array. It shows in dashed line the *I–V* curve of nine non–shadowed modules and in solid the same curve for one module receiving the lower irradiance level. When the whole array works at uniform conditions and under a low irradiance, the black curve in Fig. 1a is obtained by adding ten curves as the solid one in Fig. 2 along the voltage axis, because of the series connection of the ten modules in the string. Consequently, the mild curvature of each module *I–V* curve around the MPP is preserved in the *I–V* curve of the whole string. Instead, when the voltage values of the solid curve of Fig. 2 (shadowed substring) are summed to those ones of the dashed curve (sunny substring), the high slope of the latter one prevails over the mild shape of the former one. The greater the difference between the short–circuit currents of the solid and of the dashed curves, the more true this assertion is. The red curve in Fig. 1a is obtained by adding, for the same value of current in Fig. 2, the voltage values for both solid and dashed curves. For values of the normalized current that are lower than 0.3 A, the curvature of the red characteristic is emphasized because it is stretched towards the right side. Fig. 1a clearly shows that the rightmost knee is harsher than one resulting from a superposition of identical curves, e.g., the leftmost knee of the red curve or the one of the black curve. The difference in the curvature of the *I–V* curve around different MPPs is also evident by looking at the experimental measurements shown in many papers, e.g., in Dolara et al. [13].

2.2. Some temporary and permanent faults

In Köntges et al. [14] a useful correlation between the main PV module faults and the modifications suffered from the *I–V* curve for each one of them is presented. The faults are classified into six different categories depending on the effect each one of them has on the *I–V* curve. The non–homogeneous degradation, due to corrosion, heterogeneous delamination and loss of transparency, of modules belonging to the same string determines the switch–on of some bypass diodes, so that what is described in the previous subsection may happen as well in this case.

The mismatching due to cell cracks is not only related to a discrepancy among the current levels of different cells, but even to reverse biasing of some cells. Broken cells or disconnection of the tabs between them may lead again to a steeped *I–V* array curve, combined with a significant reduction of the short–circuit current [15]. This is confirmed in [16] for some modules through the use of electroluminescence imaging.

Irregular soiling, partial snow coverage, bird depositions or higher accumulation of dust on the bottom part of low–tilted modules (due to night moisture or a short and soft rain) lead to the same stepped shape of the *I–V* curve that has been discussed in the previous subsection [17, 18].

It is worth to note that a modified module behavior around the MPP is also associated to other types of faults. For instance, in [19] it is shown that a PV module affected by many snail trails exhibits a larger radius of curvature in its MPP region with respect to a new module. The voltage and the current values at the MPP are both reduced, but it is impossible to detect the occurrence of snail trails by measuring these values only, because they are also compatible with a healthy module working at a lower irradiance and a higher temperature. Instead, the *I–V* curve shape around the

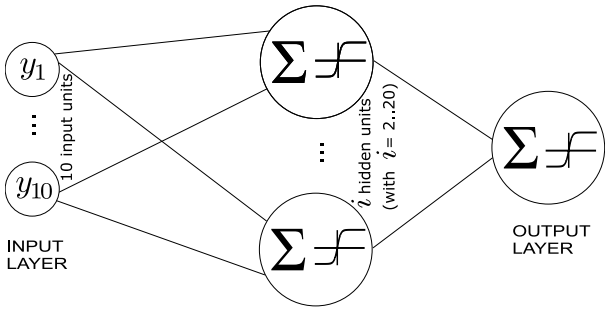


Figure 3: Multilayer perceptron with a unique hidden layer

MPP unambiguously reveals the effect of the snail trails. The same holds for the Potential Induced Degradation (PID), which becomes evident from the *I–V* curve flattening in the MPP area [14].

3. Methodology

3.1. Multilayer perceptron ANN

An ANN is a learning technique inspired on real biological neural systems composed of many simple elements, which are the neurons, each one performing a very simple operation. The connection of the output of several neurons to the input of others gives rise to an ANN that is able to model complex behaviors. The Multilayer Perceptron (MLP) is an ANN that is able to classify every input sample into different already known classes, thus implementing a supervised learning process. Although the main MLP underlying idea was introduced in the 60's, in 1986 it was presented in its current structure for the first time Rumelhart et al. [20].

The MLP neurons are structured in a layered architecture [21]: the input layer allows to enter the information from the input samples. One or more intermediate or hidden layers process the information and an output layer returns the result, e.g., the class the input sample belongs to. The output responses generated by the neurons of one layer serve as input stimulus only to the neurons of the next layer. No feedback from one layer to the previous ones and no lateral connections between neurons of the same layer occur. In Fig. 3 the MLP architecture with a single layer of hidden neurons is presented. It suits a large number of problems [22]. In the multi layer structures, the neurons of one layer are connected to each neuron of the previous layer: the value of the i^{th} neuron is multiplied by the weight w_i. Each receiving neuron is computed by summing all the inputs $w_i \cdot x_i$ and by giving the result to the input of a transfer function f to produce an output. This is described in Fig. 4. Several transfer functions can be used, depending on the specific problem.

The values of the weights w_i are determined during the training process. They are initially set as random values. Then, a number of samples belonging to known classes, i.e. the *training set*, is given at the MLP input. An optimization procedure is applied to determine the values of the weights allowing to have the known outputs for the corresponding inputs belonging to the training set. Depending on the error made by the network during the inputs processing, a correction of the weights of the last layer is determined, which, in turn, affects the weights of the previous layers up to the input one. This correction mechanism, which is iteratively applied to the elements of the training set, is known as *back–propagation* [20, 23]. This steepest descent optimization algorithm may be replaced, as it is done in this paper, with a Levenberg–Marquardt approach [24]. Once the overall error, including all the inputs belonging to the training, falls below a given threshold, the training process ends and the MLP allows to classify new inputs that were not included in the training set. The MLP classification accuracy is evaluated by using a *test set*, which includes further inputs belonging to known classes and not being included in the training set.

3.2. The ANN training data

The ANN input is a set of *I–V* samples around the MPP, which are assumed to be acquired during the normal operation of any perturbative MPPT algorithm, e.g., P&O or INC. No additional information is required at the input, such as irradiance or cell temperature readings. The ANN output is a value representing the estimated category associated to the sample presented as input, that in our case could be *healthy operation* or *faulty operation*. Finally, to improve the overall performance, several MLPs can be combined into an ensemble, so that the strengths of each

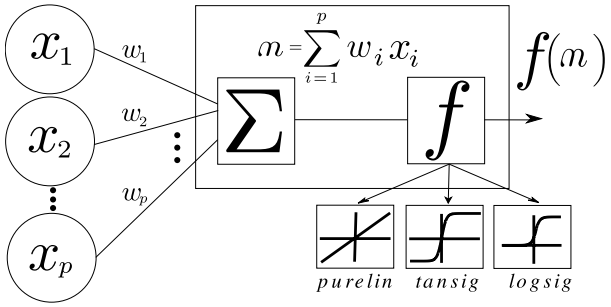


Figure 4: Estimating the output of a single neuron

Table 1
Tuning parameters for the MLP training procedure

Parameter	Value	Meaning
trainFcn	trainlm	Levenberg–Marquardt
transferFcn	tansig	Hyperbolic tangent
divideFcn	divideind	Division of samples by the user
epochs	100 000	Maximum number of epochs
max_fail	3 000	Validation checks
performFcn	mse	mean squared error

Table 2
Ranges used to generate the *I*–*V* curves using the model

	Minimum value	Maximum value
I_{ph} [A]	1	12
I_s [A]	1e−12	1e−5
η [/]	1	2
$R_{s,cell}$ [Ω]	0.001	0.01
$R_{sh,cell}$ [Ω]	1	50
T [K]	273	343
cells/bypass diode	1	30
active bypass diodes	1	5
total number of cells	6	900

individual MLP are complemented in the whole [25]. The tuning parameters required by the *Depp Learning Toolbox of Matlab* [26] and used to train the MLP are summarized in Table 1.

In order to guarantee a general applicability of the approach to a wide set of PV strings, the MLP has been trained through synthetic data generated by using the well–known Single Diode Model (SDM) by fixing sets of its parameters that are randomly picked from Table 2. It is worth to note that also the number of PV cells protected by a bypass diode is varied as well as the number of bypass diodes per module. Each module in the string has been simulated by using (1):

$$I = I_{ph} - I_s \left[\exp\left(\frac{V + I R_s}{N_s \eta V_T} \right) - 1 \right] - \frac{V + I R_s}{R_{sh}} \tag{1}$$

where V and I are voltage and current at the module terminals, I_{ph} is the photo–induced current, I_s is the saturation current, η is the ideality factor, N_s is the total number of cells of the string, R_s and R_{sh} are the series and shunt resistance, $V_T = kT/q$ is the thermal voltage, k =1.381e−23 J/K is the Boltzmann constant, q =1.602e−19 C is the electron charge, and T in K is the module operating temperature.

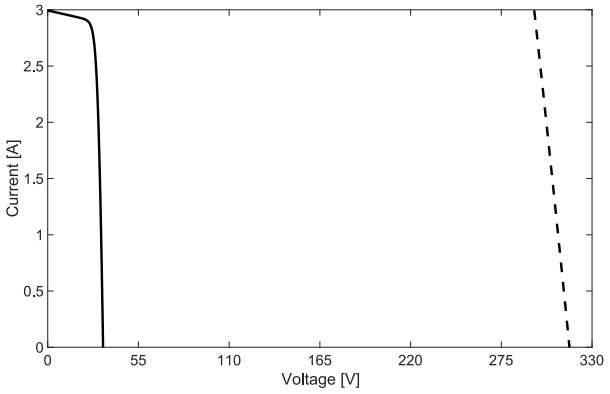


Figure 5: Synthetic data: example of *I–V* curve generation

It is worth to note that this paper focuses on the analysis of the *I–V* curve that is around the rightmost MPP, which occurs at a high voltage, not far from the open–circuit operating condition. In that region, all the bypass diodes are turned off, so that their behavior does not need to be modeled. Of course, the different photo–induced currents of the modules in the PV string have to be kept into account and the maximum operating current to simulate is the smallest module short–circuit current in the string. Each individual module can be modeled with (1) and the resulting equations can be combined together with the Kirchhoff voltage and current laws, allow to model the PV string in the high voltage range.

Only a reduced number of points of these curves have been used for the ANN training. In case of uniform conditions, the points around the unique MPP have been considered. For the *I–V* curves related to mismatched conditions, the points closest to the rightmost MPP of the curve have been the selected ones.

As an example, a case falling into the ranges listed in Table 2 is described. It refers to a PV string including 540 cells working at $I_{ph} = 9$ A and 60 cells at $I_{ph} = 3$ A. The values of the other parameters have been settled as: $I_s = 1.35\text{e}{-10}$ A, $\eta = 1$, $R_{s,cell} = 0.01\ \Omega$, $R_{sh,cell} = 5\ \Omega$ and $T = 273$ K.

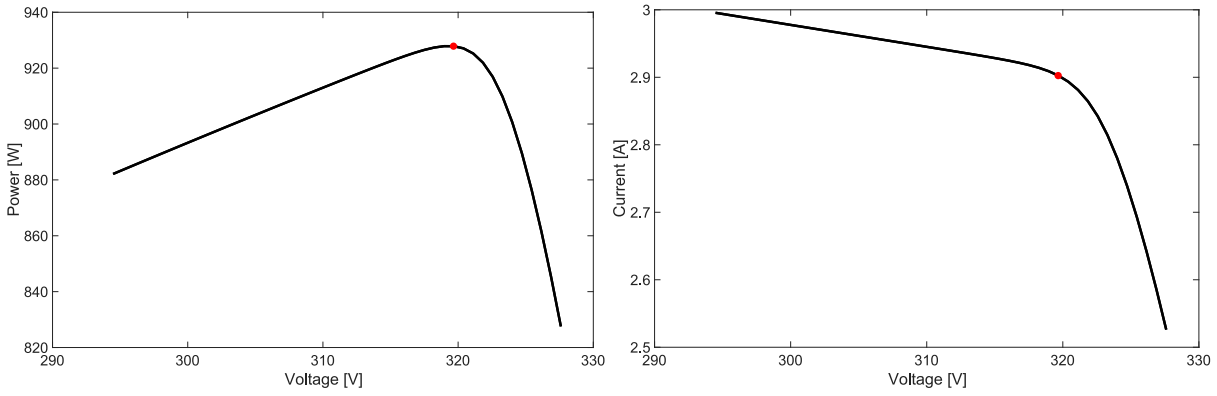
Fig. 5 shows the *I–V* curves of the two subsets of cells in the current range going from the lower I_{ph} downward to 0 A. Evidently, in this range every bypass diode is in the off state and the equation (1) is enough to plot the *I–V* curves of both the cell subsets. Thus, the model to run is very simple because the bypass diodes modeling is not required. The *P–V* curve shown in Fig. 6a is then obtained by adding the voltages of the two cell subsets shown in Fig. 5: it is enough to limit the range to $I \leqslant 3$ A and $P \geqslant 90\%\ P_{mpp}$ (P_{mpp} is the power in the MPP that is marked with a red dot in Fig. 6a and Fig. 6b). Curve portions generated in this way form the training set of the ANN, so that the ANN will be able to classify the real curve on the basis of samples distributed in a range that is very close to the MPP, where the MPPT algorithm forces the string to work in real conditions. This allows to affirm that the procedure proposed in this paper does not have any impact to the regular operation of the array.

Initially, a synthetic set of simulated *I–V* curves is available. Two subsets consisting of 5 000 curves each have been simulated, thus a total of 10 000 *I–V* curves have been obtained according to the procedure just described:

- A first subset with 5 000 *I–V* curves that are referred to PV strings operating under uniform conditions.
- A second subset of 5 000 *I–V* curves related to PV strings under different mismatched conditions.

Each one of the above sets has been divided into three groups:

- *Training set*, which consists of 3 500 curves under healthy condition and 3 500 under faulty condition, is used to train the ANN and to optimize its internal weights to fulfill a given convergence threshold.
- *Validation set*, consisting of 500 healthy curves plus 500 faulty curves, is a small set of curves that is used to avoid over–fitting: if for a consecutive number epochs of the training process, which is defined by the user as *validation checks*, the error over the training set is improved without also improving the error over this validation set, the training process is stopped.

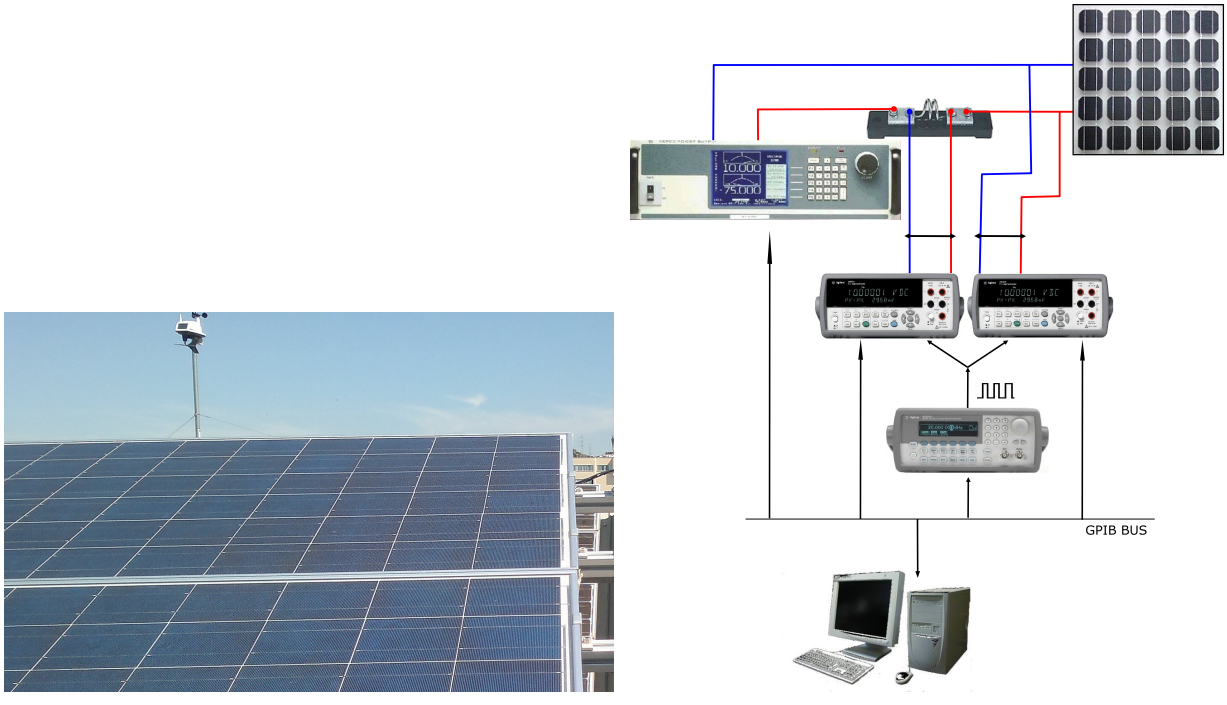


(a) P–V curve of the series (b) I–V curve of the series

Figure 6: Synthetic data: curves for $I \leqslant 3$ A and $P \geqslant 90\%\, P_{mpp}$

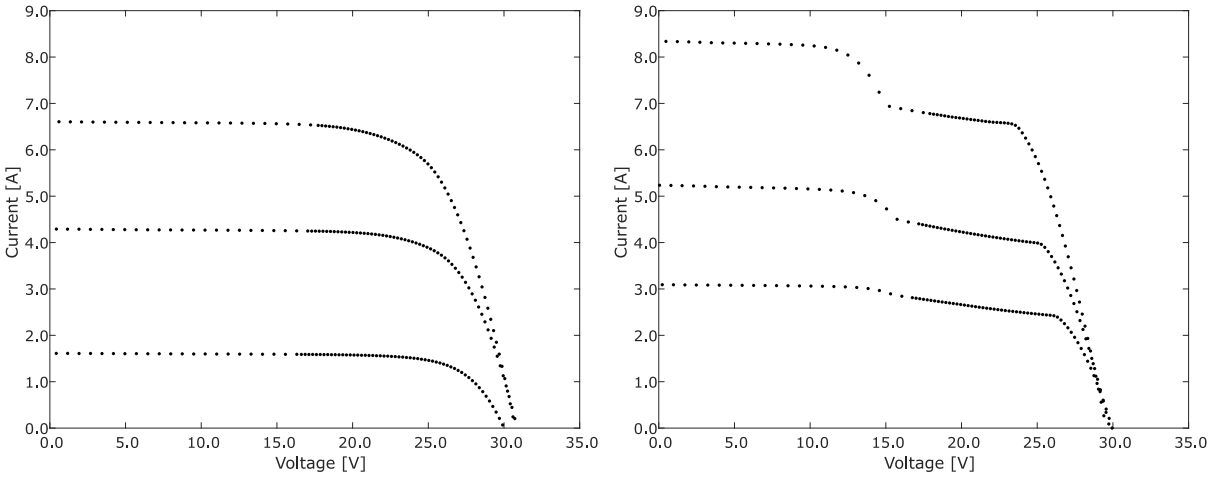
- *Test set* of synthetic samples, i.e., 1 000 healthy curves and 1 000 faulty curves, is used to measure the performance of the model with data not used during the training. In the following, this test set including synthetic curves is referred to as *TEST#1*.

The training procedure implemented in the *Matlab Deep Learning Toolbox* [26] allows to randomly assign each curve of a given set of I–V curves into *training*, *validation*, and *test* sets. Alternatively, it allows the user to provide a fixed splitting for all the executions (divideind), that is the alternative used in this work (see Table 1).



(a) PV module photo (b) system to measure the I–V curves

Figure 7: Experimental setup



(a) under healthy conditions

(b) under faulty conditions

Figure 8: Experimental *I–V* curves

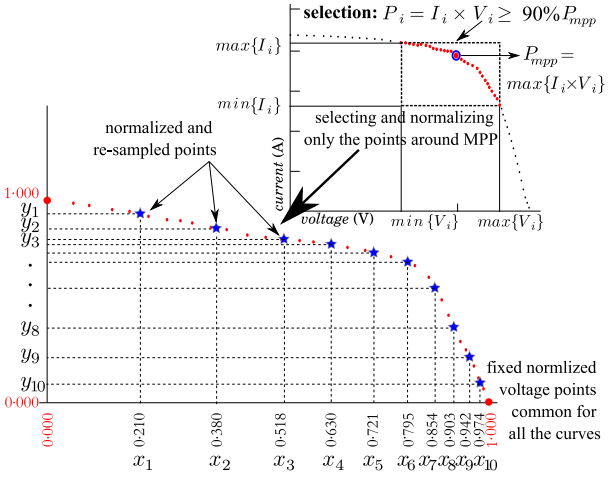


Figure 9: Selection, normalization and re–sampling of the *I–V* curve around its rightmost MPP

3.3. Experimental test set

In addition to the previous synthetic test set, a further test set of experimental curves acquired from a real PV generator has been used. The multi–crystalline silicon PV module shown in Fig. 7a) has been considered: its main nominal data are summarized in Table 3. The specifications of the hardware and software used to acquire the measurements are described in [27] and [28] respectively. A schematic draw with the main elements of the measurement system is shown in Fig. 7b.

On the one hand, Fig. 8a shows some of the *I–V* curves acquired at an early stage of use of the PV module, thus corresponding to its healthy state. On the other hand, after several years of outdoor use, the PV module was characterized by steeped *I–V* curves with a very hard shape around the rightmost knee, as it can be seen in Fig. 8b. These curves are an example of the curves corresponding to the faulty state. Two thousand of experimental curves, the half corresponding to the healthy state of the module and thus of the same type shown in Fig. 8a, and the half representing faulty conditions of the type shown in Fig. 8b, have been included in a further test set, which is named in the following *TEST#2*. It is worth to note that the experimental data have not been used to train the ANN in order to avoid that this process is strictly related to a specific module or PV generator. Instead, the use of synthetic data generated as explained in the previous subsection guarantees the generality of the ANN training over a wider scenario.

Table 3
Specifications of the PV module (at STC)

Magnitude	Unit	Value
Maximum Power $P_{mpp,\mathrm{STC}}$	W	195
Voltage at MPP $V_{mpp,\mathrm{STC}}$	V	26.0
Current at MPP $I_{mpp,\mathrm{STC}}$	A	7.50
Short–circuit current $I_{sc,\mathrm{STC}}$	A	8.27
Open–circuit voltage $V_{oc,\mathrm{STC}}$	V	32.5
Current temperature coefficient α	%/°C	+0.05
Voltage temperature coefficient β	%/°C	−0.36
Power temperature coefficient γ	%/°C	−0.46
Nominal operating cell temperature NOCT	°C	47

Although the experimental curves were available in their whole current and voltage range, they have been processed to emulate the availability of their samples in a restricted range around their rightmost MPP as it should be in the real case. The right upper corner of Fig. 9 puts into evidence the samples of one of the available experimental curves in the range comprised between P_{mpp} and the $90\%P_{mpp}$. This subset of the *I–V* samples is then normalized within the range $[0,1]\times[0,1]$ to improve the performance [29]. This has been done by means of (2) and (3), where $min\{V_i\}$ and $max\{V_i\}$ define the range of voltage values for the selected points, $min\{I_i\}$ and $max\{I_i\}$ the range of current, and x_i and y_i are the normalized values of voltage and current respectively:

$$\forall i \,:\, P_i \geq 90\%P_{mpp} \,:\, x_i = \frac{V_i - min\{V_i\}}{max\{V_i\} - min\{V_i\}} \tag{2}$$

$$\forall i \,:\, P_i \geq 90\%P_{mpp} \,:\, y_i = \frac{I_i - min\{I_i\}}{max\{I_i\} - min\{I_i\}} \tag{3}$$

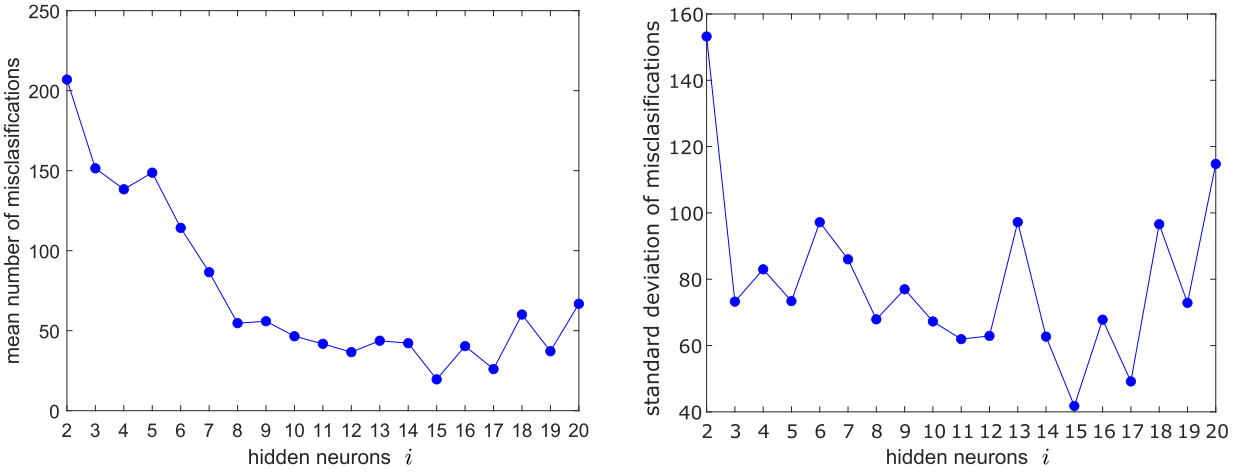
In order to have the same number of samples for every *I–V* curve, the normalized experimental samples have been given as inputs to a trivial linear interpolation procedure generating a fixed number of samples, which is 10 in the following, logarithmically spaced. Distributing the $\{x_i\}$ values in that way allows to have more points to describe the portion of the curve with higher slope. The normalized voltage values at which all the curves are newly sampled are the same for all the curves, so that the ANN does not need this information to perform the curve classification. Fig. 9 shows an example of the re–sampling procedure, which can be always performed regardless of the initial distribution of the experimental samples.

3.4. The ANN architecture

Healthy and faulty curve classification can be performed through a MLP with one hidden layer which is depicted in Fig. 3. This MLP is composed of an input layer, a unique hidden layer, and an output layer. There is one neuron for each input quantity, thus ten neurons in the input layer, one for each $\{y_i\}$ value of the normalized and re–sampled curve. The categories the curves may belong to are two, thus only one neuron in the output layer taking a binary value has been used. The adoption of the *hyperbolic tangent* as transfer function does not allow to have at the MLP output exactly -1 or $+1$. Therefore, to the value at the MLP output, the *sign* function has been applied to know the predicted category.

The number neurons in the hidden layer has to be optimized. A low number should not allow to capture the complexity of the underlying function, and a very high number leads to over–fitting the set used for training [30]. Although there are some heuristics available in the literature, a general rule to perform this task is not well assessed. In some other applications, e.g., in [31–33], the selection of the optimal number of neurons is performed starting with only a few hidden units, repeating the training procedure for higher number of neurons until the classification error over the test set does not decrease anymore.

The ANN performance depends on the random initialization of its weights [30]. Therefore, for each number of hidden neurons, several repeated executions of the training process have to be performed, so that differently trained MLPs with also different classification errors over the test set are generated. The best MLPs among all the trained ones with the same number of hidden neurons are thus selected. In addition, an ensemble of MLPs including the best



(a) mean of misclassified curves

(b) standard deviation of misclassified curves

Figure 10: Mean and standard deviation of misclassifications among 50 runs

candidates [25] can be created. Thus, the final category assigned to a new *I–V* curve comes from pooling together all the MLPs of the ensemble and getting the winner category.

4. Results of the training

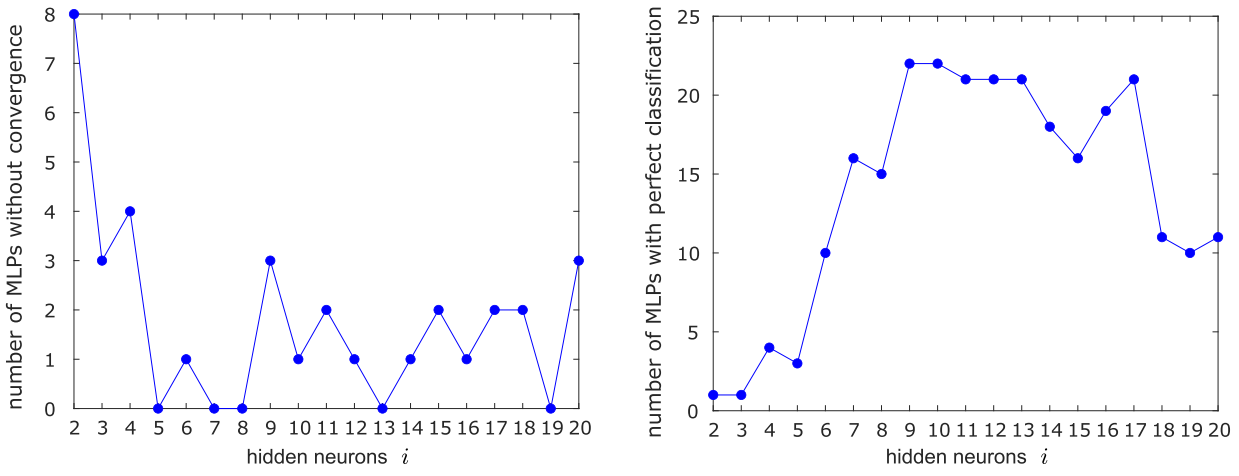
The *Matlab Deep Learning Toolbox* [26] (ver. R2021a) has been run on a laptop computer *MSI Creator 17 A10SE*, equipped with an Intel Core i7–10875H (8 cores/16 threads) at 2.30 GHz with 32 GB of DDR4 RAM memory at 2 666 MHz. The installed operating system is *Microsoft Windows 10 Pro* 64 bit (ver. 20H2). This configuration has been used for the training phase. It will be shown that the resulting ANN ensemble does not need a so powerful hardware to be used on–line to analyze in real–time the status of a PV string, because a low–cost embedded system, even integrating the MPPT algorithm, would be sufficient.

The architecture shown in Fig. 3 having one hidden layer has been tested, with an increasing number of neurons i from 2 up to 20. In order to achieve the best trained MLPs for each architecture, the training process has been repeated 50 times ($k = 1, \dots, 50$) for each value of i, and those MLPs with the minimum classification error among the 50 runs are the best ones for that number of neurons. The classification error for comparing the performance of different trained MLPs should be estimated by taking into account a test set that does not include curves used during the training. Once the MLPs have been trained, by means of a total of $20 \times 50 = 1\,000$ executions, each classification error considering *TEST#1* is estimated to select the best MLPs among the 50 runs for each value i of number of hidden neurons.

With the exception of a few cases in which the training process failed to converge and other cases with 2 hidden neurons only, all the trained MLPs are able to correctly classify all the 2 000 synthetic *I–V* curves included in the set *TEST#1*. Thus, a further and more challenging comparison among the different MLP architectures has been run on *TEST#2*, which includes 2 000 experimental curves.

For each architecture, for i from 2 until 20, the number of curves belonging to *TEST#2* that have been wrongly classified has been computed over the 50 trained MLPs (with the same value of i). Then, the mean and the standard deviation of these results among the 50 repetitions have been calculated. Fig. 10a shows a clear trend of the mean number of misclassifications, allowing to state that $i = 15$ is the best MLP architecture, also according to the standard deviation values that are shown in Fig. 10b. The latter figure highlights that for $i = 15$ the training process has been less sensitive to the initial random choice of the ANN weights, so that similar results are achieved along the 50 runs.

Fig. 11a shows, for each number i of hidden neurons, the number of runs that resulted in a failed convergence. Besides the very basic architecture including only 2 hidden neurons, this undesirable behavior takes place only 2 times for $i = 15$, that is the 4%. The number of MLPs achieving the classification of all the curves in *TEST#2*, among the 50 runs is plotted in Fig. 11b as a function of the number of hidden neurons i. The best results are given when $i = 9$ and $i = 10$, while for $i = 15$ the 32% of the 50 runs were completely successful. The corresponding 16 MLPs appear to be the best candidates for an ensemble.



(a) MLPs failed to converge

(b) MLPs with perfect classification

Figure 11: MLPs without convergence and with perfect classification among 50 runs

Table 4
Training information recorded for the 15 selected MLPs

MLP	k	Epochs	Time (s)	MSE training	MSE validation	MSE *Test#1*
M1	1	5 802	110	3.9e−11	8.2e−10	1.0e−9
M2	4	1 601	31	3.4e−11	1.4e−9	2.7e−10
M3	6	146	3	4.1e−11	6.0e−9	2.2e−7
M4	9	2 535	46	1.5e−11	6.4e−10	1.3e−9
M5	12	1 404	19	5.7e−11	1.1e−9	4.5e−9
M6	14	4 140	57	9.2e−11	3.6e−10	7.2e−10
M7	21	2 723	38	2.8e−11	2.8e−10	6.3e−10
M8	22	3 213	45	1.3e−10	4.7e−9	7.7e−9
M9	25	3 544	50	1.3e−10	9.7e−10	5.2e−10
M10	27	3 427	48	6.0e−11	2.0e−10	7.3e−10
M11	32	1 430	21	5.2e−11	5.2e−11	1.4e−9
M12	38	3 918	60	9.7e−11	1.2e−9	1.2e−10
M13	42	1 329	20	1.1e−10	2.0e−9	1.3e−9
M14	43	3 884	58	7.5e−11	2.4e−9	3.9e−9
M15	47	1851	28	3.5e−10	6.2e−9	1.3e−8

Among these 16 MLPs with $i = 15$, there is one that, accordingly to its training record (information about the training process), has achieved the worst results and has finished after very few epochs. In addition, excluding this one, the ensemble will have an odd number of MLPs and we are avoiding a tie when pooling the networks for a new curve to classify. Table 4 summarizes the recorded information relative to the training process of selected 15 MLPs, indicating the index of the repetition k, from a total of 50 runs, that has generated each network. The performance reported in Table 4 is the mean squared error (MSE) over the training, validation, and test (*TEST#1*) sets, taking into account the value at the output of the MLP without applying the *sign* function to the obtained result. As an example, Fig. 12 details the evolution of these performance indicators as the epochs progress for the MLP M12 trained in the repetition $k = 38$.

Instead of selecting one of these MLPs as the best one to use, all them can be arranged to have as the final decision the winner category among the 15 outputs. Therefore, the final model consists in an ensemble of 15 MLPs, specifically those ones listed in Table 4. Fig. 13 illustrates the proposed model. In addition, the procedure allowing to determine the occurrence of healthy or faulty conditions is described through the sequence of steps in Fig. 14. The normalized and re–sampled current values are the common input vector for each one of the 15 MLPs in the ensemble. Then, each individual MLP processes the input producing its own output, to which the *sign* function is applied, obtaining as a

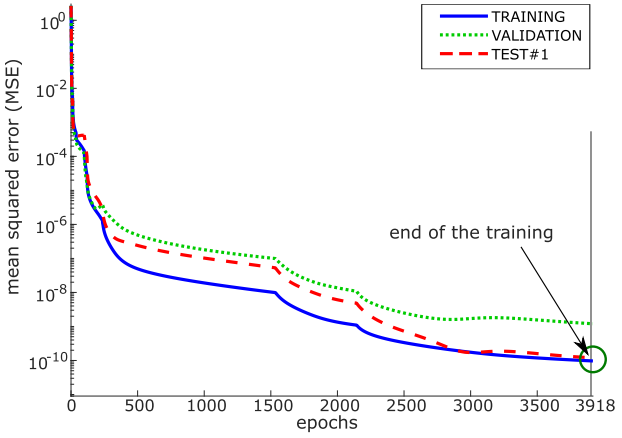


Figure 12: Training process of the MLP M12 obtained at the run $k = 38$

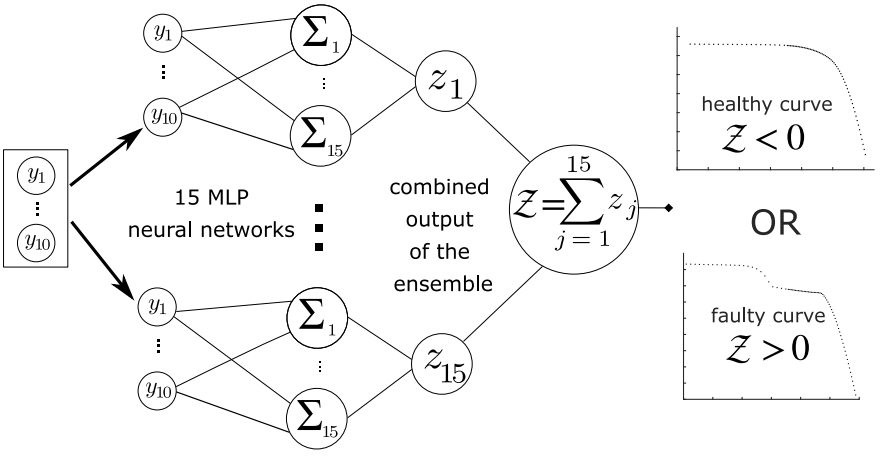


Figure 13: Diagram of operation of the ensemble of 15 MLPs

result either -1 (*healthy conditions*), or $+1$ (*failure conditions*). Once these values from z_1 to z_{15} have been obtained, all them are added up by (4) to obtain $\mathcal{Z}$. If the pooling is won by the MLPs that say *healthy*, the result will be $\mathcal{Z} < 0$. On the contrary, if the MLPs that say *faulty* win, we achieve $\mathcal{Z} > 0$.

$$\mathcal{Z} = \sum_{j=1}^{15} z_j \begin{cases} < 0 \Longrightarrow \textit{healthy curve} \\ > 0 \Longrightarrow \textit{faulty curve} \end{cases} \tag{4}$$

The ensemble appears to require a computational burden that is lower than a more complicated ANN including more than one hidden layer. Each ANN in the ensemble may run independently from each other. This means that the ensemble can be even implemented in modern embedded systems, e.g. Systems On Chip and Field Programmable Gate Arrays, which allow parallel programming, so that the computation time is dramatically shortened. Such a solution may also profit from the cloud-edge computing paradigm.

5. Application of the ensemble of MLPs

Some experimental curves there were not included in the *TEST#2* set have been used to show the behavior of the proposed ANN ensemble. These curves have been selected because they represent some challenging cases: they are shown in Fig. 15. The ensemble has been applied to each of them, by obtaining the results that can be seen in Table 5. Every following comment refers to each of these curves:

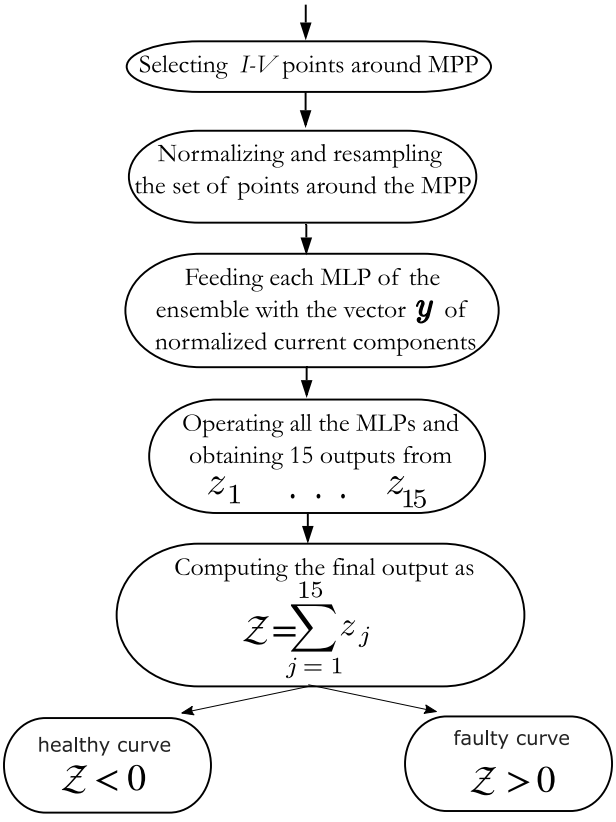


Figure 14: Sequence of steps to determine the state of a PV module or array using the ensemble of 15 MLPs

- Fig. 15a: this experimental *I–V* curve corresponds to a weak mismatching. As it comes out from Table 5, all the MLPs of the ensemble, except the one labeled as M14, classify it as a healthy curve because the shape of the portion of the curve around the MPP is very rounded, as all the cases of healthy curves used to train the MLPs.
- Fig. 15b: this experimental curve evidences a problem with the *I–V* tracer system and noise affects a few points around the MPP. This curve is correctly classified as a healthy one, although 4 MLPs of the ensemble (see Table 5) perform a wrong classification (M8, M9, M12, and M13), which should be due to the noise.
- Fig. 15c: this experimental curve shows a slight increase of the current around the MPP with respect to the short circuit current. This might be due to the increased irradiance during the acquisition of the *I–V* curve. Due to this behavior, which was not included in the training phase, 5 MLPs of the ensemble are not able to achieve a correct output (M4, M7, M10, M13, and M14).
- Fig. 15d: this *I–V* curve shows a behavior around the MPP that did not characterize and sample in the training set. There are 6 MLPs of the ensemble that classify it as a faulty curve, and 9 MLPs as a healthy one, thus confirming the fact that this curve represents a challenging issue for the classifier.
- Fig. 15e: although this curve refers to a mismatched case, the rightmost part of it shows a curvature that is typical of a curve of a uniformly irradiated module. In this case, 8 MLPs classify the curve as a healthy one and 7 MLPs as a faulty one. This confirms how challenging this curve is.
- Fig. 15f: this experimental *I–V* curve exhibits a distortion in the MPP region. Whereas there are 7 MLPs that say that the module is in healthy conditions, there are even 8 MLPs stating the contrary.
- Fig. 15g: in this curve, the classifier takes few points, that are in red color, on the right–hand side of the MPP. Thus, an important portion of the rightmost MPP has been excluded, precisely the part with the highest slope.

Table 5
Results of the application of the ensemble to the selected curves

MLP	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
M1	−1	−1	−1	+1	−1	+1	+1	+1	+1
M2	−1	−1	−1	−1	+1	+1	+1	+1	+1
M3	−1	−1	−1	+1	−1	−1	−1	+1	+1
M4	−1	−1	+1	−1	−1	+1	+1	+1	+1
M5	−1	−1	−1	−1	+1	−1	+1	+1	−1
M6	−1	−1	−1	−1	−1	−1	+1	+1	+1
M7	−1	−1	+1	+1	−1	+1	+1	+1	+1
M8	−1	+1	−1	−1	+1	−1	−1	−1	+1
M9	−1	+1	−1	−1	−1	−1	−1	−1	−1
M10	−1	−1	+1	+1	+1	+1	+1	+1	+1
M11	−1	−1	−1	−1	−1	−1	−1	−1	−1
M12	−1	+1	−1	+1	+1	+1	+1	+1	+1
M13	−1	+1	+1	+1	−1	+1	+1	+1	+1
M14	+1	−1	+1	−1	+1	+1	−1	−1	+1
M15	−1	−1	−1	−1	+1	−1	−1	−1	−1
$\mathcal{Z}$	**−13**	**−7**	**−5**	**−3**	**−1**	**+1**	**+3**	**+5**	**+7**

Without those points, the MLPs of the ensemble have more difficult in classifying this curve as a faulty one. Even so, there are more MLPs making a correct classification than those who are wrong (9 correct vs. 6 wrong).

- Fig. 15h: this *I*–*V* curve shows three MPPs. As in the previous case, the distortion of the rightmost one is evident, but again the part of the curve selected around the MPP does not include the most interesting part of the curve. Therefore, some of the MLPs of the ensemble (5 from a total of 15 MLPs) have problems in correctly classifying this example. Nevertheless, the global output of the ensemble is that the PV module is in a faulty condition, which is a correct classification.

- Fig. 15i: the behavior of this curve around the MPP might be due to a problem with the acquisition system or to a fault, the latter because the curve was measured few days before the module went out of service. The majority of the MLPs of the ensemble have classified this curve as taken under faulty conditions (11 MLPs vs 4 MLPs). This could be enough to repeat the measurement and, if the results are the same, further study should be required over that PV module.

6. Conclusions

An ensemble of multilayer perceptron neural networks is proposed for detecting mismatched conditions using as input a reduced selection of points around the knee of the *I*–*V* curve, avoiding performing an entire curve sweep and losing the tracking of the MPP, because the required points are available during the classical P&O operation. No additional irradiance nor temperature sensors are required for feeding the neural network. The normalization and re–sampling procedure proposed, made this method useful independently on the measurement rate of the *I*–*V* curve tracer. As the proposed models have been trained with synthetic data simulated taking into account a wide range for each parameter of the SDM model, a general applicability of the trained model is ensured.

Each multilayer perceptron considered has been trained using a set of synthetic *I*–*V* curves covering a very wide range of possible PV generators and external conditions to maximize the applicability of the proposed method. In order to improve the performance, different number of neurons in the hidden layer have been studied and the training procedure has been repeated enough number of times and then select those networks that have reached the lowest classification errors over a test set of experimental curves not seen during the training phase. It has been proved that the best results, in terms of mean number of misclassified curves, are achieved with 15 neurons in the hidden layer. In addition, for that best architecture, the standard deviation of that number of misclassified curves reaches its minimum value, meaning that the training process is less sensitive to the random initialization of the weights. Using

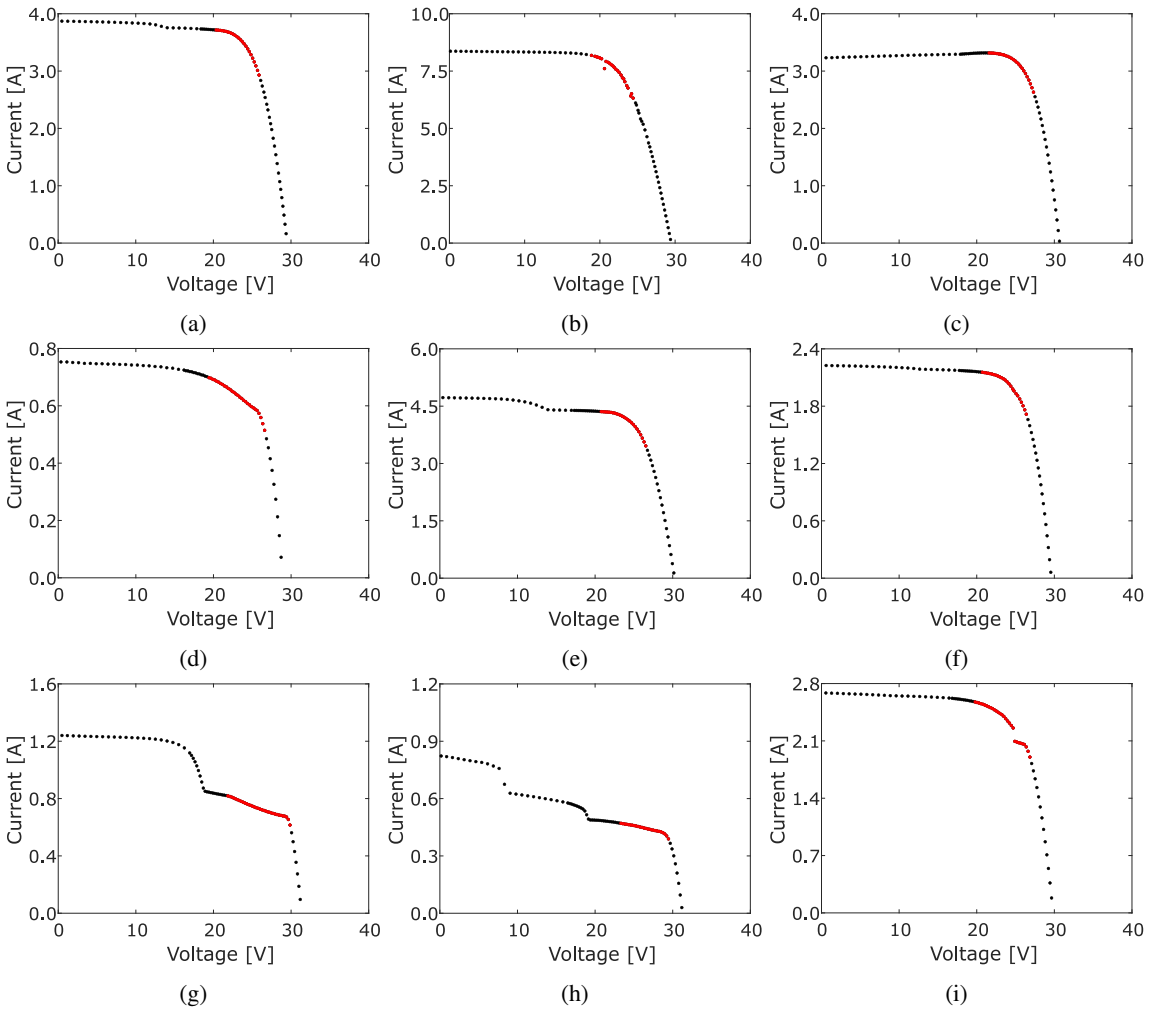


Figure 15: Selected *I–V* curves to analyze the behavior of the ensemble of MLPs

that architecture, there are available even 16 MLPs (from a total of 50 repetitions) that are able to classify perfectly all the curves of the experimental test set, proposing 15 of them to be included in the ensemble.

The work ends with some examples of application of the ensemble of multilayer perceptrons. A selection of challenging *I–V* curves with some difficulties to be classified correctly have been tested with the proposed ensemble of networks. It must be highlighted that each network behaves in a complementary way to the other, in such a way the weaknesses of some of them are covered by the strengths of the others and vice–versa.

CRediT authorship contribution statement

Michel Piliougine: Conceptualization, Methodology, Software, Validation, Data Curation, Writing – Original Draft, Writing – Reviewing and Editing, Visualization. **Giovanni Spagnuolo:** Conceptualization, Methodology Formal analysis, Writing – Original Draft, Writing – Reviewing and Editing, Visualization, Supervision.

Acknowledgments

The authors would like to acknowledge the availability of the measurements provided by the research group of the *Laboratory of Photovoltaic Systems* of the *University of Málaga* lead by Prof. Mariano Sidrach-de-Cardona.

Funding

This work has been supported by *Ministero dell'Istruzione, dell'Università e della Ricerca (Italy)* [grant PRIN2020–HOTSPHOT 2020LB9TBC and grant PRIN2017–HEROGRIDS 2017WA5ZT3_003]; *Università degli Studi di Salerno* [FARB funds]; *Ministerio de Ciencia, Innovación y Universidades (Spain)* [grant RTI2018-095097–B–I0].

Data Availability

Both synthetic and experimental datasets used in this article can be found at https://doi.org/10.5281/zenodo.4737921. The file has been stored using the native workspace Matlab format (*.mat).

References

[1] European Commission, The European Green Deal, Brussels (Belgium), 2019. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52019DC0640, Last accessed: May 13th, 2021.

[2] European Commission, Recovery Plan for Europe, 2021. URL: https://ec.europa.eu/info/strategy/recovery-plan-europe_en, Last accessed: May 13th, 2021.

[3] Renewable Capacity Statistics 2021, International Renewable Energy Agency IRENA, Abu Dhabi (UAE), 2021. URL: https://irena.org/publications/2021/March/Renewable-Capacity-Statistics-2021, ISBN: 978-92-9260-342-7.

[4] Global Energy Review 2021. Assessing the effects of economic recoveries on global energy demand and CO_2 emissions in 2021, International Energy Agency IEA, Paris (France), 2021. URL: https://www.iea.org/reports/global-energy-review-2021, Last accessed: May 13th, 2021.

[5] E. D. Chepp, A. Krenzinger, A methodology for prediction and assessment of shading on PV systems, Sol Energy 216 (2021) 537–550. URL: https://doi.org/10.1016/j.solener.2021.01.002.

[6] Cloud motion estimation from small-scale irradiance sensor networks: General analysis and proposal of a new method, Sol Energy 202 (2020) 276–293. URL: https://doi.org/10.1016/j.solener.2020.03.081.

[7] P. Kuhn, M. Wirtz, S. Wilbert, J. Bosch, G. Wang, L. Ramirez, D. Heinemann, R. Pitz-Paal, Field validation and benchmarking of a cloud shadow speed sensor, Sol Energy 173 (2018) 229–245. URL: https://doi.org/10.1016/j.solener.2018.07.053.

[8] S. Zhou, M. Mao, L. Zhou, Y. Wan, X. Xi, A shadow fault diagnosis method based on the quantitative analysis of photovoltaic output prediction error, IEEE J Photovolt 10 (2020) 1158–1165. URL: https://doi.org/10.1109/JPHOTOV.2020.2995041.

[9] S.-Q. Chen, G.-J. Yang, W. Gao, M.-F. Guo, Photovoltaic fault diagnosis via semisupervised ladder network with string voltage and current measures, IEEE J Photovolt 11 (2021) 219–231. URL: https://doi.org/10.1109/JPHOTOV.2020.3038335.

[10] M. Karakose, M. Baygin, K. Murat, N. Baygin, E. Akin, Fuzzy based reconfiguration method using intelligent partial shadow detection in PV arrays, Int J Comput Int Sys 9 (2016) 202–212. URL: https://doi.org/10.1080/18756891.2016.1150004.

[11] M. Ma, Z. Zhang, P. Yun, Z. Xie, H. Wang, W. Ma, Photovoltaic module current mismatch fault diagnosis based on i-v data, IEEE Journal of Photovoltaics 11 (2021) 779–788. doi:10.1109/JPHOTOV.2021.3059425.

[12] Y. Li, K. Ding, J. Zhang, F. Chen, X. Chen, J. Wu, A fault diagnosis method for photovoltaic arrays based on fault parameters identification, Renew Energ 143 (2019) 52–63. URL: https://doi.org/10.1016/j.renene.2019.04.147.

[13] A. Dolara, G. C. Lazaroiu, S. Leva, G. Manzolini, Experimental investigation of partial shading scenarios on PV (photovoltaic) modules, Energy 55 (2013) 466–475. URL: https://doi.org/10.1016/j.energy.2013.04.009.

[14] M. Köntges, S. Kurtz, C. Packard, U. Jahn, K. A. Berger, K. Kato, T. Friesen, H. Liu, M. Van Iseghem, Review of Failures of Photovoltaic Modules, IEA-PVPS T13-01:2014, International Energy Agency IEA, 2014. URL: https://iea-pvps.org/key-topics/review-of-failures-of-photovoltaic-modules-final/, ISBN: 978-3-906042-16-9.

[15] M. Ma, Z. Zhang, Z. Xie, P. Yun, X. Zhang, F. Li, Fault diagnosis of cracks in crystalline silicon photovoltaic modules through I-V curve, Microelectron Reliab 114 (2020) 113848. URL: https://doi.org/10.1016/j.microrel.2020.113848.

[16] J. Ahmad, A. Ciocia, S. Fichera, A. F. Murtaza, F. Spertino, Detection of typical defects in silicon photovoltaic modules and application for plants with distributed MPPT configuration, Energies 12 (2019) 4547. URL: https://doi.org/10.3390/en12234547.

[17] C. Schill, S. Brachmann, M. Koehl, Impact of soiling on IV-curves and efficiency of PV-modules, Sol Energy 112 (2015) 259–262. URL: https://doi.org/10.1016/j.solener.2014.12.003.

[18] J. G. Bessa, L. Micheli, F. Almonacid, E. F. Fernández, Monitoring photovoltaic soiling: assessment, challenges, and perspectives of current and potential strategies, iScience 24 (2021) 102165. URL: https://doi.org/10.1016/j.isci.2021.102165.

[19] A. Dolara, G. C. Lazaroiu, S. Leva, G. Manzolini, L. Votta, Snail trails and cell microcrack impact on PV module maximum power and energy production, IEEE J Photovolt 6 (2016) 1269–1277. URL: https://doi.org/10.1109/JPHOTOV.2016.2576682.

[20] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536. URL: https://doi.org/10.1038/323533a0.

[21] S. Haykin, Multilayer perceptrons, in: Neural Networks and Learning Machines, 3 ed., Pearson, Upper Saddle River, NJ, USA, 2009, pp. 122–229. URL: https://www.pearson.com/us/higher-education/program/Haykin-Neural-Networks-and-Learning-Machines-3rd-Edition/PGM320370.html, ISBN:978-0-13-147139-9.

[22] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989) 359–366. URL: https://doi.org/10.1016/0893-6080(89)90020-8.

[23] P. J. Werbos, Backpropagation: Past and future, in: IEEE International Conference on Neural Networks, volume 1, San Diego (CA, USA), 1988, pp. 343–353. URL: https://doi.org/10.1109/ICNN.1988.23866.

[24] H. Yu, B. M. Wilamowski, Levenberg–Marquardt training, in: B. M. Wilamowski, J. D. Irwin (Eds.), Intelligent Systems, The Industrial Electronics Handbook, CRC Press, Boca Raton (FL, USA), 2011. URL: https://doi.org/10.1201/9781315218427, ISBN: 978-1-4398-0283-0.

[25] T. Windeatt, Ensemble MLP classifier design, in: L. Jain, M. Sato-Ilic, M. Virvou, G. Tsihrintzis, V. Balas, C. Abeynayake (Eds.), Computational Intelligence Paradigms, volume 137-147, Springer, Berlin, Heidelberg (Germany), 2008. URL: https://doi.org/10.1007/978-3-540-79474-5_6, ISBN: 978-3-540-79474-5.

[26] M. H. Beale, M. T. Hagan, H. B. Demuth, Matlab Deep Learning Toolbox. User's Guide. R2021a, MathWorks, Natick (MA, USA), 2021. URL: https://www.dropbox.com/s/ovzdr8ergaxxoym/nnet_ug.pdf, Last accessed: May 13th, 2021.

[27] M. Piliougine, J. Carretero, L. Mora-López, M. Sidrach-de-Cardona, Experimental system for current–voltage curve measurement of photovoltaic modules under outdoor conditions, Prog Photovoltaics 19 (2011) 591–602. URL: https://doi.org/10.1002/pip.1073.

[28] M. Piliougine, J. Carretero, L. Mora-López, M. Sidrach-de-Cardona, New software tool to characterize photovoltaic modules from commercial equipment, WEENTECH Proceedings in Energy (2018) 211–220. URL: https://doi.org/10.32438/WPE.6218.

[29] J. Sola, J. Sevilla, Importance of input data normalization for the application of neural networks to complex industrial problems, IEEE T Nucl Sci 44 (1997) 1464–1468. URL: https://doi.org/10.1109/23.589532.

[30] J. F. Mas, J. J. Flores, The application of artificial neural networks to the analysis of remotely sensed data, Int J Remote Sens 29 (2008) 617–663. URL: https://doi.org/10.1080/01431160701352154.

[31] M. Piliougine, D. Elizondo, L. Mora-López, M. Sidrach-de-Cardona, Photovoltaic module simulation by neural networks using solar spectral distribution, Prog Photovoltaics 21 (2013) 1222–1235. URL: https://doi.org/10.1002/pip.2209.

[32] M. Piliougine, D. Elizondo, L. Mora-López, M. Sidrach-de-Cardona, Multilayer perceptron applied to the estimation of the influence of the solar spectral distribution on thin-film photovoltaic modules, Appl Energ 112 (2013) 610–617. URL: https://doi.org/10.1016/j.apenergy.2013.05.053.

[33] M. Piliougine, D. Elizondo, L. Mora-López, M. Sidrach-de Cardona, Modelling photovoltaic modules with neural networks using angle of incidence and clearness index, Prog Photovoltaics 23 (2015) 513–523. URL: https://doi.org/10.1002/pip.2449.