An embedded system for remote monitoring and fault diagnosis of photovoltaic arrays using machine learning and the internet of things

A. Mellit [a,*], M. Benghanem [b], S. Kalogirou [c], A. Massi Pavan [d]

[a] Department of Electronics, Faculty of Science and Technology, University of Jijel, Jijel, Algeria
[b] Department of Physics, Faculty of Science, Islamic University of Madinah, Madinah, Saudi Arabia
[c] Department of Mechanical Engineering and Materials Sciences and Engineering of the Cyprus University of Technology, Limassol, Cyprus
[d] Department of Engineering and Architecture, and Center for Energy, Environment and Transport Giacomo Ciamician, University of Trieste, Italy

ARTICLE INFO

Keywords:
Photovoltaic array
Fault diagnosis
Monitoring system
Machine learning
Embedded system

ABSTRACT

In this paper a novel embedded system for remote monitoring and fault diagnosis of photovoltaic systems is introduced. The idea is to embed machine leaning algorithms into a low-cost edge device for real-time deployment. First, an artificial neural network is developed to detect faults. Then an effective stacking ensemble learning algorithm is developed to classify the nature of the fault. The method performance is evaluated through common error metrics such as RMSE, MAE, MAPE, r and confusion matrix. Additional algorithms are also embedded into the edge device in order to remotely control the photovoltaic array parameters. Users can be notified by email and SMS about the state of their photovoltaic array. The Blynk IoT platform is used to monitor remotely the photovoltaic array parameters. The experimental results demonstrate the ability of the proposed embedded system to diagnose and monitor the photovoltaic array with a good accuracy.

1. Introduction

According to the International Energy Agency more than 940 GW [1] of photovoltaic (PV) capacity were installed at the end of 2021, which means that a huge number of PV plants were installed worldwide. In order to keep these plants working efficiently, with small losses and short down times, and in order to avoid the risk of fire, it is important to detect faults early and take rapid decisions [2–5].

Currently, visual inspection is the mostly used method to check the correct functionality of PV plants. However, this method is relatively expensive and very time consuming as the operator needs to go to the site where the PV plant is installed, which is not always accessible or feasible. Therefore, the design of a remote-monitoring system with an innovative fault diagnosis method plays an important role for the early fault detection and diagnosis of PV plants [6,7].

In this direction, numerous monitoring, supervision such as SCADA [8] and protection systems [9] have been developed. Recently, researchers are more and more attracted by the Internet of Things (IoT) and Industrial IoT (IIoT) to design smart monitoring systems [10–12]. This can significantly reduce the efforts in helping users to control their installation remotely [13]. The feasibility of remote monitoring systems based on Embedded System (ES) and IoT is demonstrated in Ref. [14]. A new monitoring solution (narrowband-IoT) suitable for self-consumption was developed in Ref. [15]. The developed monitoring system can monitor parameters such as current, voltage, and other environmental variables. The main contribution is that the designed prototype was verified for a real PV self-consumption system.

In [16] the authors developed an IoT-enable smart energy meter by using LoRa (Long Range) network to monitor consumed energy by a PV system. The designed system is able to monitor and upload data to the cloud, the collected data are: current, power, voltage, temperature, light intensity and relative humidity). A centralized cloud-based solar conversion recovery system was proposed in Ref. [17]. This configuration combines IoT and centralized architecture to remotely control the amount of soiling deposits on PV modules. An artificial neural network (ANN) is integrated with this system to reduce the amount of required hardware in this system.

Recently, Artificial Intelligence (AI) techniques including Machine Leaning (ML), Ensemble Learning (EL), and Deep Learning (DL) based-algorithms have gained popularity in dealing with PV fault detection and classification problems [18–21]. They showed their capability to classify major faults or defects, such as shading, dust or sand

* Corresponding author.
E-mail addresses: adel_mellit@univ-jijel.dz (A. Mellit), benghanem_mohamed@yahoo.fr (M. Benghanem), soteris.kalogirou@cut.ac.cy (S. Kalogirou), apavan@units.it (A. Massi Pavan).

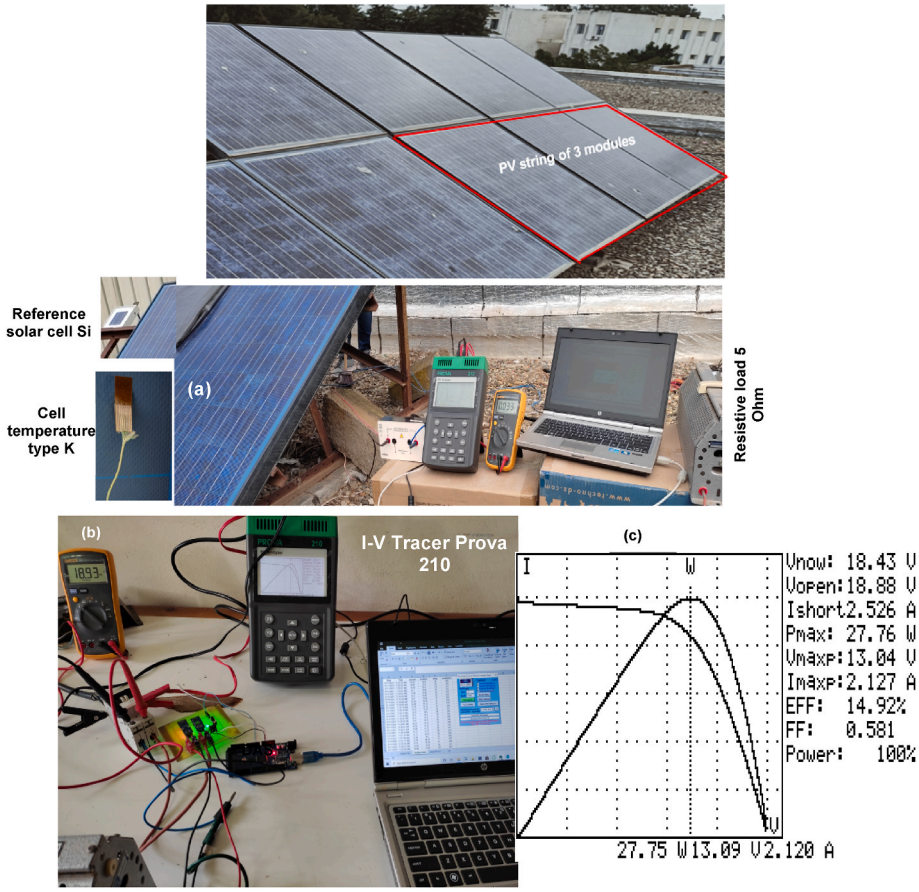

Fig. 1. Test facility: a) I–V curves logged with a "Prova-210" tracer, b) solar irradiance, air temperature and PV power measurement using a data-acquisition system (DAS) based on an Arduino™ Microcontroller, c) example of measured I–V and P–V curves.

accumulation on the PV surface, bypass diodes failures, disconnections of PV modules, shunting of PV modules, and other defects. An explainable ANNs for PV fault detection is designed in Ref. [22]. The obtained results showed the usefulness to create valuable insights on model comportment. In Ref. [23] the authors developed and applied ensemble learning (EL) algorithms to detect and classify possible line-to-line (LL) faults. This approach is based on the measurement of the I–V curves and the main features were extracted to fed the model. In this work only LL fault is addressed. A 3-D convolutional neural network (CNN) was developed for fault detection and classification of faults in a PV system. The method is based on the image processing of a the collected I–V curves and the results demonstrated that this approach outperforms ML-based algorithms, such as K-Nearest Neighbour (K-NN), Random Forest (RF) and Support Vector Machine (SVM) [24]. In Ref. [25], the authors have clearly demonstrated the importance and the capability of ML-based methods to classify complex fault based on the I–V curves. In terms of accuracy, the classification rate was in the range [81%–99%]. Nevertheless, the developed method was not verified experimentally. A robust logistic regression method for fault diagnosis of PV arrays is developed [26] and the results demonstrated that data processing can increase diagnosis accuracy up to 18.4%.

It should be pointed out that most available fault diagnosis methods-based on ML have been tested and simulated using Matlab/Simulink, Python or other programming languages, while very limited attempts have been done to check them experimentally. According to the literature, there are no papers related to the development of embedded systems using IoT and ML techniques for the purpose of fault detection and diagnosis of PV arrays.

Recently, in our previous work [27] we have designed a PV fault detection and classification method using decision tree and random forest algorithms. However, in this method only single faults were addressed. In addition, the system was not able to notify the users about the state of their PV system. To address the limitations of this system, a novel fault diagnosis and monitoring system is proposed and verified experimentally.

The main contributions can be listed as follows:

- A hybrid fault diagnosis method based on an Artificial Neural Networks (ANN) and a Stacking Ensemble Machine Learning-based algorithm (SEL) has been developed.
- The developed algorithms have been embedded into a microprocessor (i.e., design an embedded machine learning system) for a real-time deployment.
- Both single and multiple faults are addressed in this work. In such a situation the identification process became relatively complex and require more advanced ML algorithms.
- Two additional algorithms have been also integrated in order to monitor remotely and notify the users regarding the state of their PV system.
- The Blynk IoT dashboard application is developed to store and visualize remotely data (such as currents, voltages, incident solar irradiance, air temperature, cell temperature and the state of the system).

2. Methods and data

2.1. Datasets description

In order to develop the fault detection and diagnosis method, two datasets have been built, the first one consists of 4500 samples with a

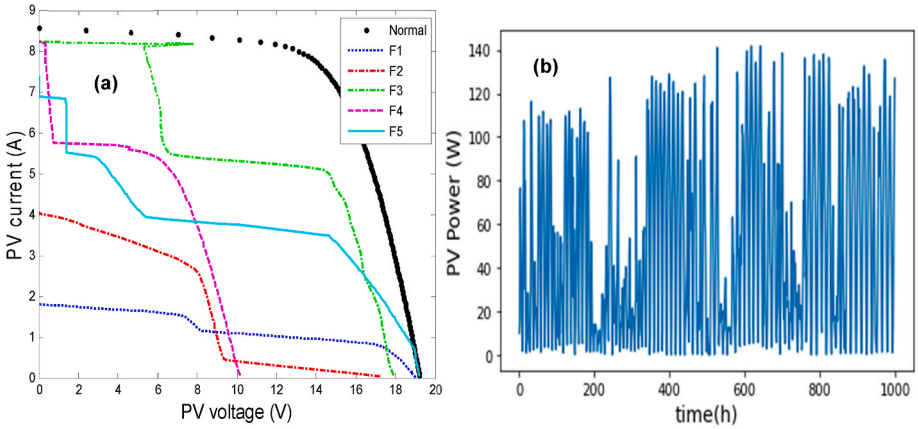

Fig. 2. a) Measured I–V curves under normal and abnormal (Faulty curves: F1, F2, F3, F4 and F5) working conditions, b) PV powers for a period of 1000 h.

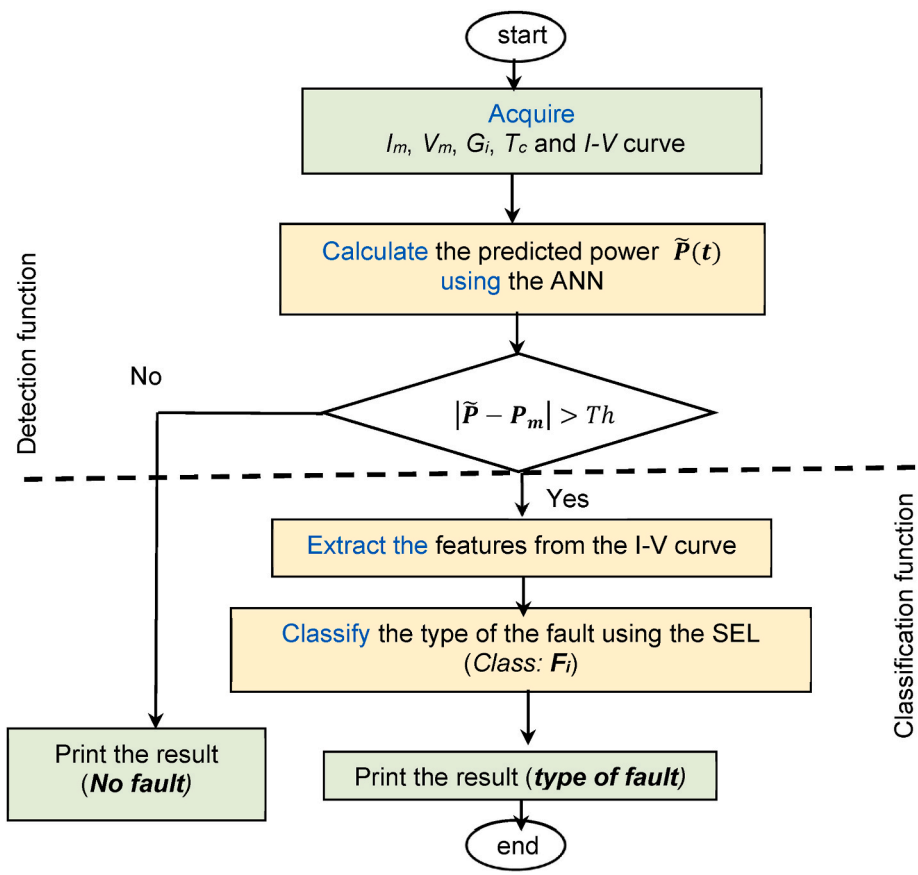

Fig. 3. Flowchart of the fault detection and diagnosis method based on ANN and SEL.

time step of 1 h which includes measured data of solar incident irradiance, air temperature, cell temperature, and PV output power. The second dataset contains 1585 measured I–V curves collected at various working conditions and include both normal and faulty cases. Fig. 1 shows the test facility used to collect the data in order to build both datasets. The considered PV array consists of three PV modules connected in parallel; the nominal power of the PV modules is 60 Wp.

In this work two categories of faults have been addressed:

1) **Single faults**: dust deposit on PV module surface (F1) and partial shading (F2);
2) **Multiple faults:** open circuited diode and dust accumulation on a PV module (F3), partial shading and dust accumulation (F4), and shunted diode in a shaded PV module (F5).

As an example, Fig. 2a shows the collected I–V curves under both normal and abnormal working conditions. Fig. 2b depicts the measured PV power (P) for 1000 samples.

2.2. Features extraction and selection

From an I–V characteristic curve, six main features can be easily extracted: voltage at open circuit (V_{oc}), current at short circuit (I_{sc}), voltage at the maximum power (V_{mp}), current at the maximum power

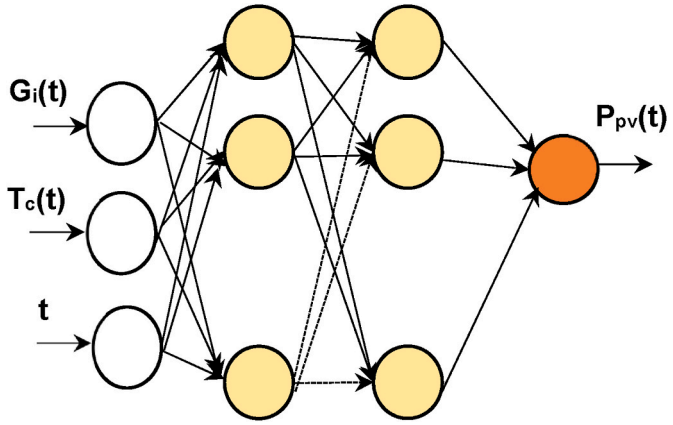

Fig. 4. MLP-based PV power prediction.

(I_{mp}), maximum power (P_{mp}), and Fill Factor (FF). Other features could be extracted from the I–V curve, however, to simplify and reduce the complexity of the algorithms only the above-mentioned features are considered. These features are extracted from the I–V curve as follows: Let's consider two vectors $\mathbf{I_n(k)}$ and $\mathbf{V_n(k)}$, corresponding to the measured current and voltage respectively, where n represents the size of the vectors. Therefore:

$$V_{oc} = \mathbf{I}_n(0),\ I_{sc} = \mathbf{V_n}(0)$$

$$\mathbf{P}_n = \mathbf{I}_n * \mathbf{V}_n$$

$$\{x, y\} = \mathrm{Max}(\mathbf{P_n}),\ V_{mp} = V_n(x),\ I_{mp} = I_n(y)$$

$$P_{mp} = \max(\mathbf{P_n})$$

$$FF = P_{mp} * (I_{sc} * V_{oc})^{-1}$$

2.3. The proposed fault detection and classification method

This subsection aims to describe the fault detection and classification method. First, an ANN is developed to detect the fault based on the predicted PV power, and then a Stacking Ensemble Machine Learning-based algorithm (SEL) is developed to classify and identify the type of fault. The flowchart of the developed PV fault detection and classification is illustrated in Fig. 3 which includes two main functions: the detection and the classification function.

The parameters I_m, V_m, G_i, T_c, and I–V are the measured current, voltage, incident solar irradiance, cell temperature and I–V curve respectively, Th is a defined threshold (estimated after a number of experiments), P_m is the measured power, $\widetilde{P}$ is the predicted power and F_i is the fault type $\{i = 1, \ldots, 5\}$.

2.3.1. Fault detection based on artificial neural networks

To develop the fault detection algorithm, the well-known multilayer perceptron neural networks (MLP) is used. Fig. 4 shows the basic structure of the MLP used to predict the PV power. The inputs of the MLP are G_i, T_c at time t, while the output is the produced PV power P_{pv} at time t.

The first dataset is divided into two parts: the first contains 80% (3600 samples) of data used to train the MLP, and the rest 20% (900 samples) is used to test the model. The mathematical formula of the MLP-based model can be given as follows:

$$\widetilde{P} = MLP(G_i, T_c, t) \tag{1}$$

The fault detection procedure is carried out by comparing the predicted power with the measured P_m one. If the difference ($\Delta P = |\widetilde{P} - P_m| > Th$) is greater than a defined threshold (Th), a fault that is identified, otherwise the system is in normal operation.

The main tuned parameters are: number of neurons in each hidden layer, the activation function type (Tansig, ReLU, etc.), the training algorithm (trainlm, traingdx, adam, etc.), and the learning rate parameter.

2.3.2. Fault classification based on a stacking ensemble learning

To classify the faults, a stacking ensemble learning classifier is developed. Stacking belongs to the category of EL algorithms that learns how to best combine predictions from multiple high-performance ML models. Fig. 5 shows the workflow of the SEL classifier. It consists of two levels: the first level contains the base selected models: CatBoost [28], XGBoost [29] and LightGBM [30]. The second level consists of a Random Forest (RF) algorithm [31], which can be considered as a meta-model.

The second dataset is used to develop the SEL classifier. Open source libraries of CatBoost (Gradient decision on boosting tree), XGBoost (eXtreme Gradient Boosting) and LightGBM (Light Gradient Boosting Machine) have been used to develop the classifier. The three ML algorithms have different hyper-parameters that should be tuned; these are essential for the good performance of the model. In this work a grid search technique is used to hyper-parameter tuning [32].

The main steps for training and testing the stacking EL classifier are summarized in the following procedure:

Step #1. Split the dataset (**k**-folds, e.g. $\mathbf{k} = 10$)

Step #2. Select 1 fold for validation and **k**-1 (9 folds) for training

Step #3. Train the selected base models (CatBoost, XGBoost and LightGMB) on the training set

Step #4. Generate the corresponding 3 predictions on the validation set

Step #5. Repeat steps 2 and 4 for the **k**-1 folds and create an augmented dataset with the predictions of each base model

Step #6. Train the meta-mode (random forest algorithm) on the

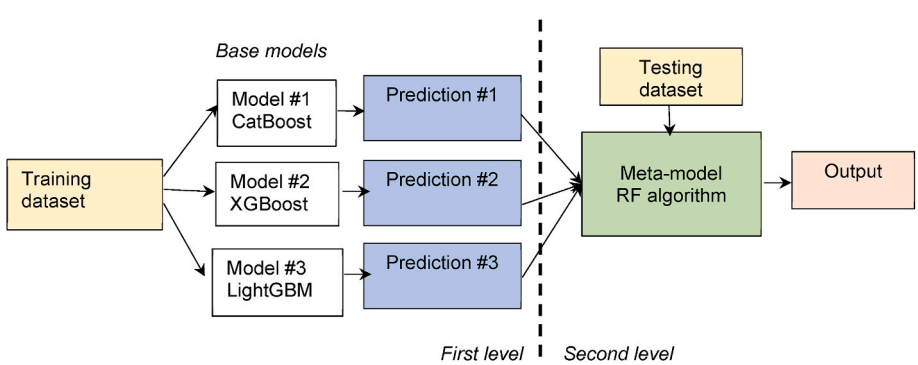

Fig. 5. Stacking ensemble learning-based fault classification.

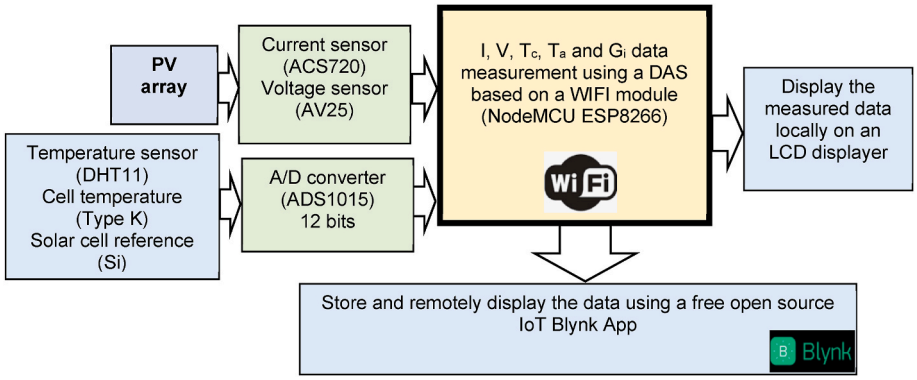

Fig. 6. Basic configuration of the developed smart remote monitoring system based on the IoT technique using an open source IoT Blynk application.

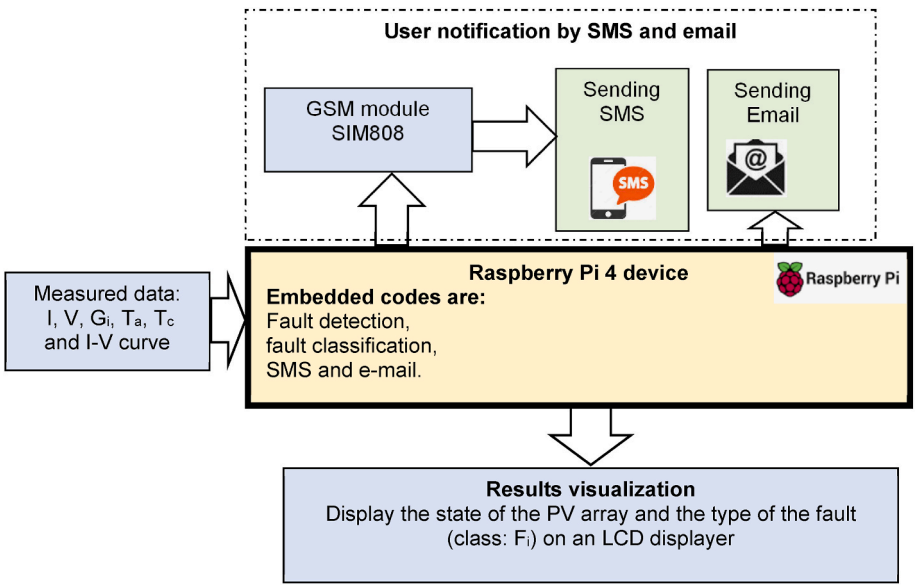

Fig. 7. The proposed embedded system for fault diagnosis and remote monitoring of PV arrays.

augmented dataset

The expression of the SEL classifier can be given as:

$$\textit{Class} = SEL\left(I_{sc}, V_{oc}, I_{mp}, V_{mp}, P_{mp}, FF\right) \quad (2)$$

This stacking EL classifier is then used to classify the nature of PV faults based on the extracted features.

2.4. The proposed remote-monitoring and fault diagnosis system

2.4.1. The smart remote monitoring system

The smart monitoring system aims to monitor remotely the considered PV array by displaying the following parameters: G_i, T_c, T_a, I, V and P. Fig. 6 depicts the basic configuration of the PV monitoring system using the IoT technique. An open source IoT platform named Blynk is used. This comprises an analogue to digital converter for the irradiance and the cell temperature, and a number of sensors for the measurements of currents, voltages, solar irradiances and temperatures. A low-cost WiFi module (NodeMCU ESP8266) is used to collect and display the measured parameters in a webpage. The parameters can be displayed locally into an LCD display too.

The algorithm is embedded into an ESP8266 microcontroller using the C programming language. To monitor and visualize the measured data using a smart-phone, a Blynk App is employed. The Blynk is an IoT company which provides a platform for building mobile (IOS and Android) applications that can connect electronic devices to the internet and remotely monitor and control these devices.

2.4.2. The proposed embedded system

In order to use the developed method in real-time, the codes have been implemented into the Raspberry Pi 4. The flowchart of the developed embedded system is shown in Fig. 7 where the essential part is the embedded fault detection and diagnosis method. This consists of two incorporated ML-based algorithms. The first is the MLP used to check if the PV array is healthy or unhealthy. The second is a SEL-based multi-class classifier used to identify the nature of the fault, which should be immediately started in order to make a prompt decision. Two additional algorithms have been also developed and embedded into the Raspberry Pi 4 in order to notify the users by SMS and email about the status of their PV array.

The implemented codes were written in Python language, which is suitable for implementing such ML-based algorithms into the microprocessor Raspberry Pi 4. In addition, various machine learning and deep learning libraries are open sources (e.g. Keras, Tensorflow, etc.) and provide a Python interface, which is the main reason why programming language was used.

The steps of the embedded system are summarized as:

Step #1. Read the I–V curve

Step #2. Extract features

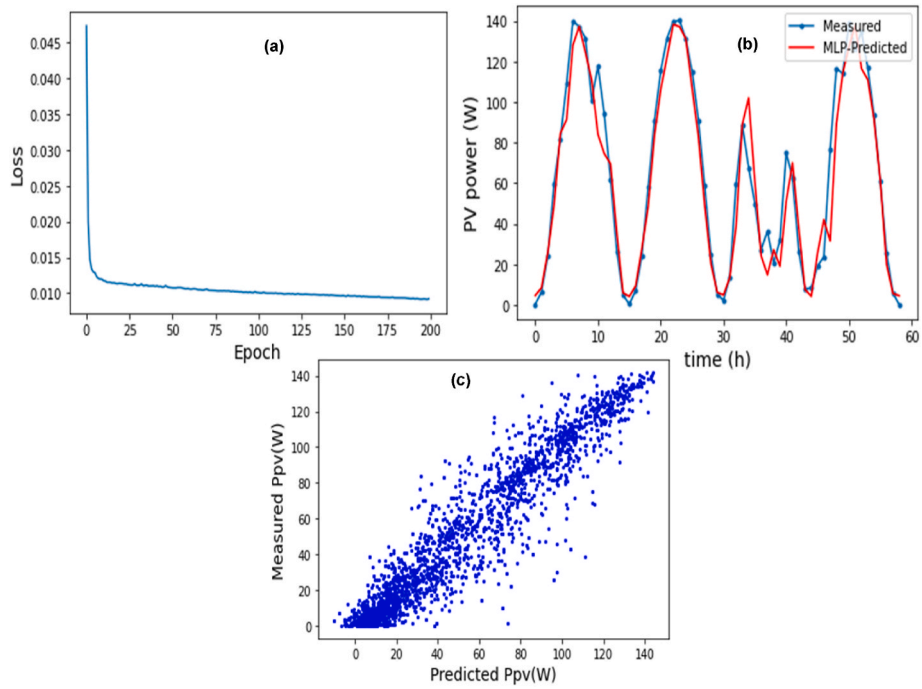

Fig. 8. a) Loss function, b) Comparison between measured and MLP-predicted PV powers, c) Scatter curve.

Table 1
ANN-model parameters and calculated error metrics (RMSE, MAE, r and MRPE).

ANN-Model parameters	RMSE (W)	MAE (W)	r (%)	MRPE (%)
Training algorithm = *Trainlm*, Epoch = 200, Activation function = *TanSig*, Number of hidden layer = 1, Number of neurones = 50, Learning rate = 10^{-2}	0.05	0.02	97.5	−2.85

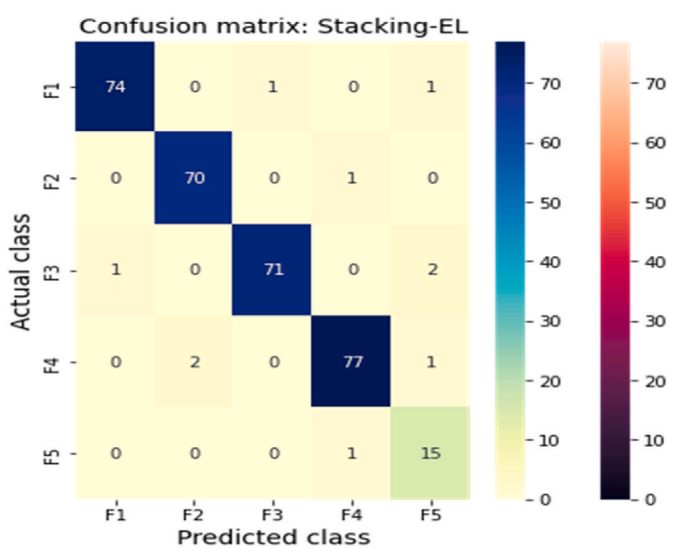

Fig. 9. The confusion matrix of the SEL-classifier.

Step #3. Import and run the optimized file

If the class ≠{1} **then** go to **Step #5**.

Step #4. Import and run the multiclass procedure

Table 2
Model hyper-parameters and the calculated error metrics (Precision, recall, F1-score and accuracy of the SEL-classifier).

Model hyper-parameters	Faults' classes	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
XGBoost:	**Fault classification using SEL algorithm**				
min_data_in_leaf = 200, max_depth = 4 colsample_bytree = 0.63, gamma = 3.2, max_depth = 5.0, min_child_weight = 1.0, reg_alpha = 55, reg_lambda = 0.91	F1 {class #1 }	99	97	98	96.80
	F2 {class #2 }	97	99	98	
	F3 {class #3 }	99	96	97	
CatBoostClassifier: (iterations = 2000, learning_rate = 0.05, random_strength = 0.78, depth = 5, border_count = 190)	F4 {class #4 }	97	96	97	
	F5 {class #5 }	79	94	86	
LGBMClassifier: learning_rate = 0.014, boosting_type = gbdt, num_leave = 25, max_depth = 5, iterations = 1000, bagging_fraction = 0.8.					

Step #5. Display the result on the LCD, send email and SMS

Step #6. Display the measured data on the Blynk platform

2.5. *Performance metrics*

To assess the performance of the developed ML-based models, errors metrics such as the Mean Absolute Error (MAE), Mean Relative Percentage Error (MRPE), Root Mean Squared Error (RMSE), and the correlation coefficient have been used.

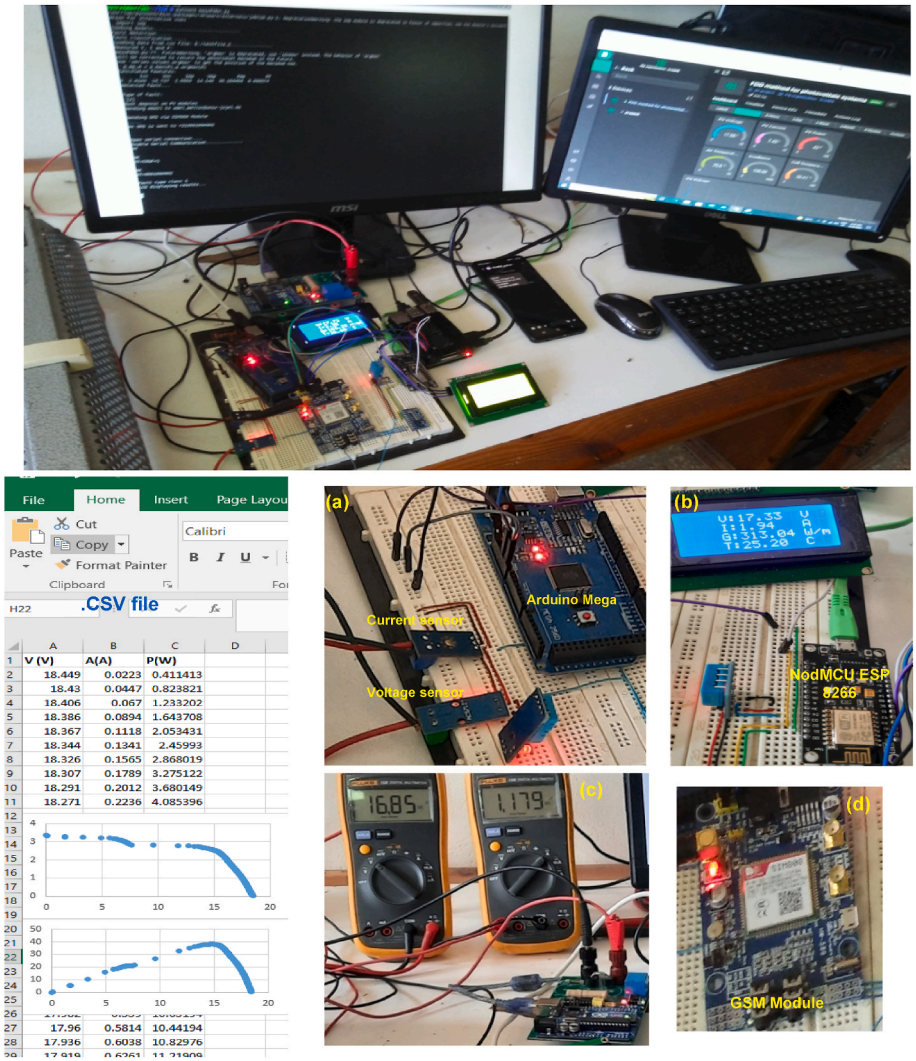

Fig. 10. Smart remote monitoring and PV fault diagnosis system: a) DAS for measuring I, V, G_i, T_c and T_a, b) Remote monitoring system using NodeMCU ESP8266 WiFi module, c) I–V tracer circuit for measuring and saving data into CSV format and d) GSM module (SIM808).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i| \tag{3}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{x_i - y_i}{x_i}\right| \tag{4}$$

$$r = \frac{\sum_{i=1}^{n}((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x}_i)^2 \sum_{i=1}^{n}(y_i - y_i)^2}} \tag{5}$$

$$RMSE = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{n}(x_i - y_i)^2\right)} \tag{6}$$

Where x_i and y_i are the measured and forecasted values, respectively, and $\bar{x}$ and $\bar{y}$ are the average values of the measured and the forecasted data, respectively. The parameter n represents the length of the observation.

The confusion matrix including precision, recall, F1-score and accuracy have beeneen calculated and used to evaluate the SEL classifier algorithm.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{7}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{8}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{9}$$

$$F1 - score = \frac{2(Precision * Recall)}{(Recall + Precision)} \tag{10}$$

Where *TP* is the number of true positive, *TN* is the number of true negative, *FP* is the number of fault positive, and *FN* is the number of fault negative.

3. Results

3.1. Simulation results

Concerning the fault detection stage, the predicted PV power for eight days is depicted in Fig. 8. Here, the cost function is less than 0.01, and an acceptable correlation between measured and predicted PV powers (Fig. 8b and c) can be observed, particularly for sunny days.

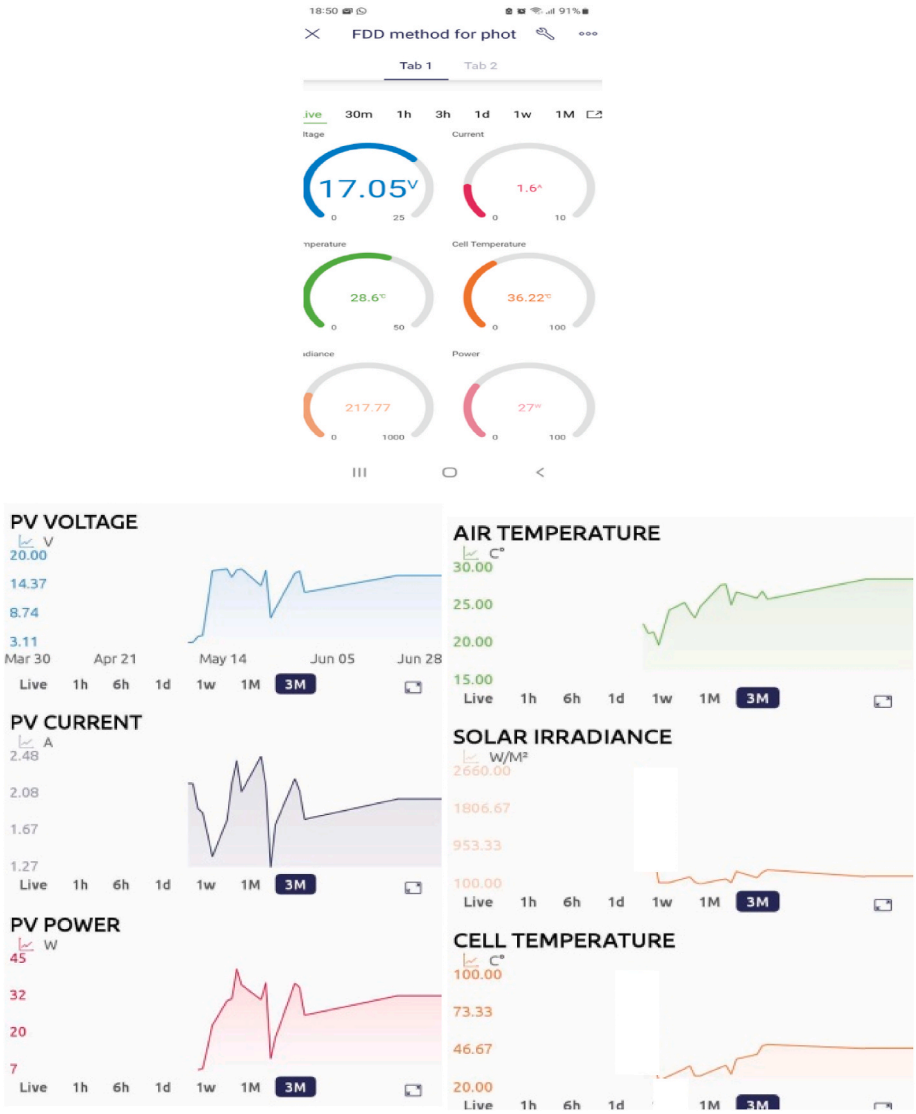

Fig. 11. Dashboard based on the Blynk App (smart-phone) displaying the measured data in real time (I, V, P, T_a, T_c and G_i).

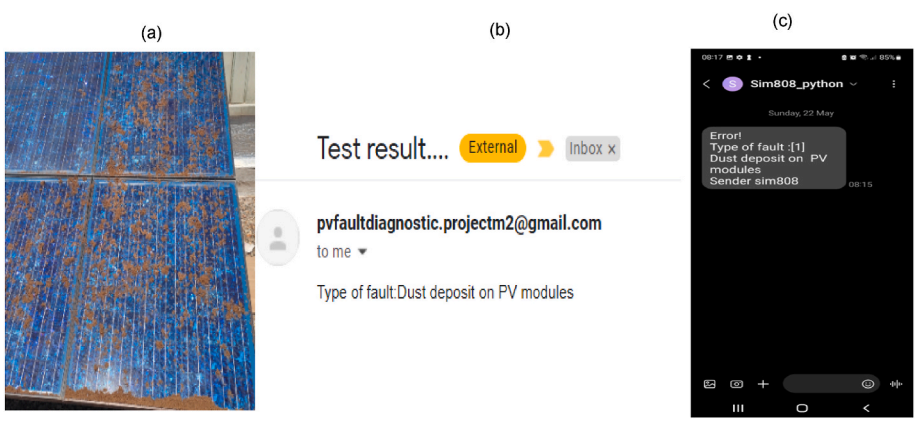

Fig. 12. Experimental results: a) Dust deposit on PV modules (fault class #2), b) Received email from the developed system, c) Received SMS from the GSM module.

Table 1 lists the parameters of the ANN model and the calculated errors. The correlation coefficient is 97.5%, the MRPE is 2.43%, while the RMSE and MAE are quite small. The MLP-model has an approximate tolerance of 2.5% that needs to be considered when defining the threshold to avoid false alarm (fault detection function). In addition, this model should be trained periodically over time to keep its accuracy.

Regarding the classification stage, the Confusion Matrix (CM) is shown in Fig. 9. From the first row of the CM, it can be seen that two samples of class F1 were misclassified in F3 and F5 classes. In the second row, one sample of class F3 is misclassified in F4. In the third row three

Table 3
Advantages, disadvantages and recommendations for the considered system.

Advantages	Disadvantages	Recommendations
Inexpensive as embedded fault diagnosis solution.	Database preparation requires an expert supervision (e.g. labelling defects).	Increase the size and improve the quality of the dataset.
Help operators to make adequate decisions.	Databases with faults are not always available.	Generalize the method for different PV module technologies.
Maintain PV systems working safely.	The classification accuracy needs to be improved, particularly for multi-classification (complex faults).	Study the scalability of the method.
Help to reduce the maintenance operation costs.	The database should be periodically updated.	Use ultra-low power edge device.
Smart remote PV monitoring system.		Apply more efficient ML algorithms.

Table 4
The estimated cost and uncertainty of the used sensors and devices.

Components	Specification	Sensitivity/ Resolution	Uncertainty (%)	Cost ($)
Sensors				
Current sensor	ACS720 (30A)	66 mV/A	0.2	3
Voltage sensor	AV25	5 V/25 mA	1	2
Cell temperature	Type K	0.1 °C	±0 .75	5
Air temperature	DHT11	±1 °C	±2 °C	2
Reference solar cell	Mono crystalline Si	< ms; matched to PV response	±5%	55
Electronic devices				
ADC converter	ADS1015	12 bits 4 channels	0.01%	6
NodeMCU ESP8266 Wifi	WiFi module	–	–	8
Raspberry pi 4	4G	–	–	58
LCD display	16 × 04	–	–	3
SIM808	GSM module	–	–	15
I–V tracer	I–V	–	–	50
Total cost	**207**			

samples of F3 are misclassified in F1 and two in F5 classes. In the fourth row, three samples of class F4 are also misclassified in F2 and F5 classes. For the last row, one sample of class F5 is misclassified in F4. The calculated precision, recall, F1-score and accuracy are listed in Table 2. The misclassified rate is 3.5%.

Table 2 reports the tuned hyper-parameters and the calculated error metrics of the developed classifier model. The classification accuracy is 96.50%. It can be concluded that the SEL classifier can identify the faults F1 and F3 (corresponding to dust and open circuited diode + dust) with good precision (99%). Fault F2 and F3 (corresponding to shading and shading dust) with acceptable precision (97%), as the I–V curves in both faults are relatively similar. The model classifies the F5 with (shunted diode + shading) with a low precision. This error (misclassified rate 3.5%) was mainly due to the complex I–V curve and similarity between certain types of fault, which make the classification task difficult. Also, this error should be considered when calculating the uncertainty of the designed embedded system.

3.2. Experimental results

Fig. 10 shows the test facility including the designed smart remote monitoring and the PV fault diagnosis system with the data-acquisition system for measuring I, V, G_i, T_c and T_a (Fig. 10a) using low-cost sensors. A remote monitoring system is based on the IoT technique (Fig. 10b) using a WiFi module (NodeMCU ESP8266). An I–V tracer circuit for measuring and saving the I–V characteristic into a CSV format is shown in Fig. 10c. A GSM module (SIM808) is used for sending messages (SMS) to the user (Fig. 10d).

The developed codes have been optimized and then embedded into the Raspberry Pi 4. Fig. 11 illustrates an example of the online collected and posted data in the Blynk App. The posted data have been collected for a cloudy day with a PV voltage, current and power of 17.05 V, 1.6 A, and 28 W respectively.

To check experimentally the designed remote-monitoring and fault diagnosis system various experiments have been carried out. As an example, a fault class #1 (dust accumulated on the PV module surface, Fig. 12a) was created, and the embedded codes inside the Raspberry Pi 4 have been run. As a result, Figs. 12b and c show the email and the SMS received indicating the type of the fault (F1: Dust deposit on a PV module PV module).

3.3. Cost and uncertainties

Table 4 lists the used components, specification, cost and uncertainties. The budget of sensors including current, voltage, air temperature, cell temperature and reference solar cell was approximately 67 $. While the budget of the electronics devices including WiFi-module, microprocessor, digital displayer, GSM module and I–V curve tracer was approximately 140 $. It should be noted that these costs can be reduced if the prototype is scaled up from the laboratory scale to mass production.

The accuracy of the designed embedded system depends on the accuracy of the classifiers (MLP = 2.5%, SEL = 3.5%), and on the uncertainty of the sensors used to measure currents, voltages, cell temperatures and solar irradiances.

4. Discussion

In some of the tests (such as F2, F4 and F5 faults), a wrong notification was received. This means that the multiclass classifier needs to be improved working on the dataset quality and size. In the case of the fault detection procedure based on the ANN, particularly for cloudy days, the produced power is estimated with low accuracy, and this leads to false alarms. Thus, a more efficient algorithm should be also developed to predict the PV power accurately. The dataset plays a vital role to increase the accuracy of the fault detection procedure. It is worth noting that the capacity (RAM, CPU or GPU) of the used hardware (Microprocessor or Microcontroller) should be carefully considered when using complex ML-based algorithms. Some advantages and disadvantages, of the designed system, are given in the following Table 3. Recommendations are also listed in Table 3.

The main results and findings can be summarized as follows:

- An MLP-based model was developed for the fault detection and an SML algorithm was developed for the classification purpose.
- Simulation results showed a good accuracy for both classifiers (the detection rate is 97.5%, while the classification rate is 96.8%).
- Single and multiple faults have been addressed at the same time while measuring the I–V curve.
- Experimental tests indicate the capability of such embedded system to remotely monitor and diagnose the PV array.
- The Blynk mobile App was employed to monitor and display data in real-time via a digital dashboard.

5. Conclusion and perspectives

In this work a smart embedded system for remote monitoring and fault diagnosis of a photovoltaic array is proposed and verified experimentally. The feasibility of designing an embedded machine learning for fault diagnosis of PV arrays was demonstrated. The designed system is

able to notify the users by email and SMS of the status of their photovoltaic system. Future work will focus on:

- The possibility to generalize the method and use it for different photovoltaic technologies.
- The accuracy of both models which could be improved by increasing the size of the dataset and by applying more efficient algorithms (e.g. multi-stacking ensemble learning or deep reinforcement machine learning).
- The average cost which could be reduced by using low-cost and ultra-low-power microcontrollers.
- Test the system in different countries under different weather conditions.

CRediT authorship contribution statement

A. Mellit: Conceptualization, Methodology, Resources, Software, Validation, Writing - original draft. **M. Benghanem:** Methodology, Data curation, Formal analysis, Funding acquisition, Visualization. **S. Kalogirou:** Supervision, Visualization, Writing - review & editing. **A. Massi Pavan:** Supervision, Conceptualization, Project administration, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the "Ministero dell'Istruzione, dell'Università e della Ricerca" (Italy) under the Grant PRIN2020–HOTSPHOT 2020LB9TBC.The first author would like to acknowledge support from the ICTP through the Associates Programme (2023–2028). The second author extend their sincere gratitude to the Deanship of Scientific Research at the Islamic University of Madinah for the support provided to the Post-Publishing Program 1.

References

[1] Snapshot of Global PV Markets, Report IEA-PVPS T1-42:2022. https://iea-pvps.org/snapshot-reports/snapshot-2022/, April 2022. (Accessed 25 April 2022).
[2] L. Hernández-Callejo, S. Gallardo-Saavedra, V. Alonso-Gómez, A review of photovoltaic systems: design, operation and maintenance, Sol. Energy 188 (2019) 426–440, https://doi.org/10.1016/j.solener.2019.06.017.
[3] D.S. Pillai, F. Blaabjerg, N. Rajasekar, A comparative evaluation of advanced fault detection approaches for PV systems, IEEE J. Photovoltaics 9 (2019) 513–527, https://doi.org/10.1109/JPHOTOV.2019.2892189.
[4] 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. Energy 143 (2019) 52–63, https://doi.org/10.1016/j.renene.2019.04.147.
[5] S. Sairam, S. Seshadhri, G. Marafioti, S. Srinivasan, G. Mathisen, K. Bekiroglu, Edge-based explainable fault detection systems for photovoltaic panels on edge nodes, Renew. Energy 185 (2022) 1425–1440, https://doi.org/10.1016/j.renene.2021.10.063.
[6] S. Silvestre, L. Mora-López, S. Kichou, F. Sánchez-Pacheco, M. Dominguez-Pumar, Remote supervision and fault detection on OPC monitored PV systems, Sol. Energy 137 (2016) 424–433, https://doi.org/10.1016/j.solener.2016.08.030.
[7] A. Mellit, S. Kalogirou, Artificial intelligence and internet of things to improve efficacy of diagnosis and remote sensing of solar photovoltaic systems: challenges, recommendations and future directions, Renew. Sustain. Energy Rev. 143 (2021), 110889, https://doi.org/10.1016/j.rser.2021.110889.
[8] L.O. Aghenta, M.T. Iqbal, Development of an IoT based open source SCADA system for PV system monitoring, in: IEEE Canadian Conference of Electrical and Computer Engineering, 2019, pp. 1–4, https://doi.org/10.1109/CCECE.2019.8861827.
[9] F. Spertino, F. Corona, Monitoring and checking of performance in photovoltaic plants: a tool for design, installation and maintenance of grid-connected systems, Renew. Energy 60 (2013) 722–732, https://doi.org/10.1016/j.renene.2013.06.011.
[10] N.S. Deshmukh, D.L. Bhuyar, A smart solar photovoltaic remote monitoring and controlling, in: Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 67–71, https://doi.org/10.1109/ICCONS.2018.8663127.
[11] M. Belouda, A. Mami, Embedded solution for data acquisition and management strategy dedicated to a hybrid renewable energy source for remote electricity supply, Microprocess. Microsyst. 90 (2022), 104496, https://doi.org/10.1016/j.micpro.2022.104496.
[12] M. Dong, J. Zhao, D.A. Li, B. Zhu, S. An, Z.I.S.E.E. Liu, Industrial Internet of Things perception in solar cell detection based on edge computing, Int. J. Distributed Sens. Netw. 17 (2021), 15501477211050552, https://doi.org/10.1177/15501477211050552.
[13] A. Mellit, A. Hamied, V. Lughi, A. Massi Pavan, A low-cost monitoring and fault detection system for stand-alone photovoltaic systems using IoT technique, in: ELECTRIMACS, Springer, Cham, 2019, pp. 349–358. https://link.springer.com/chapter/10.1007/978-3-030-37161-6_26.
[14] R.I. Pereira, I.M. Dupont, P.C. Carvalho, S.C. Jucá, IoT embedded linux system based on Raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant, Measurement 114 (2018) 286–297, https://doi.org/10.1016/j.measurement.2017.09.033.
[15] J.M. Paredes-Parra, R. Jiménez-Segura, D. Campos-Peñalver, A. Mateo-Aroca, A. P. Ramallo-González, A. Molina-García, Democratization of PV micro-generation system monitoring based on narrowband-IoT, Sensors 22 (13) (2022) 4966, https://doi.org/10.3390/s22134966.
[16] W.A. Jabbar, S. Annathurai, T.A.A. Rahim, M.F.M. Fauzi, Smart energy meter based on a long-range wide-area network for a stand-alone photovoltaic system, Expert Syst. Appl. 197 (2022), 116703, https://doi.org/10.1016/j.eswa.2022.116703.
[17] M. Ul Mehmood, A. Ulasyar, W. Ali, K. Zeb, H.S. Zad, W. Uddin, H.J. Kim, A new cloud-based IoT solution for soiling ratio measurement of PV systems using artificial neural network, Energies 16 (2) (2023) 996, https://doi.org/10.3390/en16020996.
[18] A. Mellit, S. Kalogirou, Machine learning and deep learning methods for fault diagnosis of photovoltaic systems, in: A. Mellit, S. Kalogirou (Eds.), Handbook of Artificial Intelligence Techniques in Photovoltaic Systems: Modeling, Control, Optimization, Forecasting and Fault Diagnosis, Academic Press, 2022, ISBN 9780128206416.
[19] C. Kapucu, M. Cubukcu, A supervised ensemble learning method for fault diagnosis in photovoltaic strings, Energy 227 (2021), 120463, https://doi.org/10.1016/j.energy.2021.120463.
[20] S.Q. Chen, G.J. Yang, W. Gao, M.F. Guo, Photovoltaic fault diagnosis via semi-supervised ladder network with string voltage and current measures, IEEE J. Photovoltaics 11 (2020) 219–231, https://doi.org/10.1109/JPHOTOV.2020.3038335.
[21] A. Mellit, S.A. Kalogirou, Machine learning and deep learning for photovoltaic applications, in: Artificial Intelligence for Smart Photovoltaic Technologies, AIP Publishing LLC, Melville, New York, 2022, https://doi.org/10.1063/9780735424999_001, 1-1.
[22] C. Utama, C. Meske, J. Schneider, R. Schlatmann, C. Ulbrich, Explainable artificial intelligence for photovoltaic fault detection: a comparison of instruments, Sol. Energy 249 (2023) 139–151, https://doi.org/10.1016/j.solener.2022.11.018.
[23] A. Eskandari, J. Milimonfared, M. Aghaei, Line-line fault detection and classification for photovoltaic systems using ensemble learning model based on IV characteristics, Sol. Energy 211 (2020) 354–365, https://doi.org/10.1016/j.solener.2020.09.071.
[24] Y.Y. Hong, R.A. Pula, Detection and classification of faults in photovoltaic arrays using a 3D convolutional neural network, Energy 246 (2022), 123391, https://doi.org/10.1016/j.energy.2022.123391.
[25] A. Mellit, S. Kalogirou, Assessment of machine learning and ensemble methods for fault diagnosis of photovoltaic systems, Renew. Energy 184 (2022) 1074–1090, https://doi.org/10.1016/j.renene.2021.11.125.
[26] M. Wang, X. Xu, Z. Yan, Online fault diagnosis of PV array considering label errors based on distributionally robust logistic regression, Renew. Energy 203 (2023) 68–80, https://doi.org/10.1016/j.renene.2022.11.126.
[27] A. Mellit, O. Herrak, C. Rus Casas, A. Massi Pavan, A machine learning and internet of things-based online fault diagnosis method for photovoltaic arrays, Sustainability 13 (23) (2021), 13203, https://doi.org/10.3390/su132313203.
[28] A.V. Dorogush, V. Ershov, A. Gulin, CatBoost: gradient boosting with categorical features support, arXiv:1810.11363, https://doi.org/10.48550/arXiv.1810.11363, 2018.
[29] T. Chen, C. Guestrin, Xgboost: a scalable tree boosting system, in: Proceedings of the 22nd ACM - International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.
[30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.Y. Liu, Lightgbm: a highly efficient gradient boosting decision tree, Adv. Neural Inf. Process. Syst. 30 (2017) 1–9. https://dl.acm.org/doi/10.5555/3294996.3295074.
[31] C. Zhang, Y. Ma (Eds.), Ensemble Machine Learning: Methods and Applications, Springer Science & Business Media, 2012, https://doi.org/10.1007/978-1-4419-9326-7.
[32] P. Liashchynskyi, Grid search, random search, genetic algorithm: a big comparison for NAS, arXiv:1912, https://doi.org/10.48550/arXiv.1912.06059, 2019.