Cost-effective fault diagnosis of nearby photovoltaic systems using graph neural networks

Jonas Van Gompel*, Domenico Spina, Chris Develder

IDLab, Department of Information Technology, Ghent University - imec, Technologiepark-Zwijnaarde 126, 9052, Ghent, Belgium.

Abstract

The energy losses and costs associated with faults in photovoltaic (PV) systems significantly limit the efficiency and reliability of solar power. Since existing methods for automatic fault diagnosis require expensive sensors, they are only cost-effective for large-scale systems. To address these drawbacks, we propose a fault diagnosis model based on graph neural networks (GNNs), which monitors a group of PV systems by comparing their current and voltage production over the last 24 h. This methodology allows for monitoring PV systems without sensors, as hourly measurements of the produced current and voltage are obtained via the PV systems' inverters. Comprehensive experiments are conducted by simulating 6 different PV systems in Colorado using 6 years of real weather measurements. Despite large variations in number of modules, module type, orientation, location, etc., the GNN can accurately detect and identify early occurrences of 6 common faults. Specifically, the GNN reaches $84.6\% \pm 2.1\%$ accuracy without weather data and $87.5\% \pm 1.6\%$ when satellite weather estimates are provided, significantly outperforming two state-of-the-art PV fault diagnosis models. Moreover, the results suggest that GNN can generalize to PV systems it was not trained on and retains high accuracy when multiple PV systems are simultaneously affected by faults.

Keywords: Photovoltaics, Predictive maintenance, Fault detection, Graph neural network, Time series classification

1. Introduction

The globally rising average temperature is having a visible impact on our planet, including an increased frequency and intensity of heat waves, wildfires and storms [1]. The International Energy Agency reports that the average temperature is currently 1.1 °C higher than in the pre-industrial era and that exploiting solar energy will play a key role in halting this rising trend [2]. To this end, increasing the number of photovoltaic (PV) installations and maximizing their efficiency is crucial. Although the efficiency of PV cells has been steadily improving, faults in PV systems cause significant energy losses and damage, especially if these faults remain undetected for years [3]. Therefore, performing predictive maintenance by detecting and identifying early manifestations of faults in PV systems is essential to maximize their efficiency and lifetime. The increase in extreme weather events due to climate change will also give rise to more thermal and mechanical stresses in PV systems, further increasing the need for widespread PV fault diagnosis.

Faults that occur in PV systems include short circuits, wiring degradation, hot spots, etc. Although most faults can be diagnosed through visual inspection by technicians or infrared thermography via drones, these techniques are only cost-effective for large-scale PV systems, meaning most PV systems remain unmonitored [4].

Fault diagnosis methods based on artificial intelligence, and specifically machine learning, provide an interesting alternative for monitoring PV systems. These methods avoid the need to define system-specific thresholds and do not necessarily require expensive sensors, while retaining high accuracy [5]. Various machine learning-based fault diagnosis techniques have been proposed [6]. According to the principle adopted to detect and identify faults, these can be divided into the following five categories:

1) Comparing the expected and measured output of the PV system. This is a straightforward and commonly adopted methodology [7–14]. The expected output can be predicted by either physics-based PV simulations [7, 9, 10] or via regression [11, 12]. In both cases, ambient weather conditions, such as irradiance and temperature, are required as inputs to obtain the expected power output.

The methods proposed in [11] and [12] both predict the power output using a multilayer perceptron, which is a type of artificial neural network. In [11], faults are detected by defining a threshold on the difference between the expected and measured power output. Similarly, the technique proposed in [12] determines thresholds on 1kHz voltage and current measurements to classify faults. These high-frequency measurements allow for accurate identification of the fault type, but are costly to gather, trans-

*Corresponding author.
E-mail address: Jonas.VanGompel@UGent.be

mit and process. Instead of using a multilayer perceptron to predict the power output, the method proposed in [7] combines threshold rules and a multilayer perceptron to facilitate accurate fault identification. This method performs fault diagnosis using characteristics of the I-V curve as inputs, which provide more information than simply the current and voltage at maximum power point (MPP). Unfortunately, an I-V curve tracer is required at the PV system site to gather this data.

Despite the apparent simplicity of using thresholds, defining optimal threshold rules requires expert knowledge and becomes increasingly complex as more fault types are considered [5]. The methods proposed in [9] and [10] perform fault diagnosis using exclusively probabilistic neural networks and gradient boosted trees, respectively. The downside of avoiding threshold rules is that the technique is less interpretable, since most machine learning models are black boxes.

Instead of relying on weather sensors at the PV system site, the technique proposed in [13] obtains the expected current and voltage via reference PV modules. The ratio of the measured and expected output is then classified as normal, short or open circuit via the local and global consistency algorithm, which is based on semi-supervised clustering in a graph [13]. Although this method does not require installation of weather sensors, the fault diagnosis will fail if any fault occurs in the reference modules. Another alternative approach is to use satellite estimations of the irradiance and temperature as inputs [14]. This method requires neither weather sensors nor reference modules. However, the weather estimation errors caused by the satellite negatively impact the fault diagnosis accuracy.

2) I-V curve classification. The PV system's I-V characteristics curve provides substantially richer information than solely the produced current and voltage at maximum power point (I_{mpp} and V_{mpp}) [15]. Hence, several methodologies have been proposed to perform fault diagnosis based on I-V curves. For instance, the authors of [15] manually define relevant parameters from I-V curves, based on the analysis of I-V curves under various fault and weather conditions. These parameters are used as input for a kernel extreme learning machine, which accurately classifies the input as either normal or one of the four considered faults. A similar methodology was proposed in [16], where a set of fuzzy classifiers performs fault diagnosis using parameters which are extracted from I-V curves. As opposed to manually defining variables from I-V curves, deep learning models were proposed in [17] and [18] to automatically extract features from the raw I-V curves. These methods use a convolutional neural network and ResNet, respectively, to identify faults based on I-V curves, irradiance and ambient temperature measurements. Note that the on-site sensor requirements to gather both I-V curves and weather measurements make these methods relatively costly to implement.

3) Identifying fault transients in I_{mpp} and V_{mpp}. When monitoring the produced current and voltage with sufficiently high temporal resolution, transient behavior can be detected at the moment a fault occurs. Moreover, particular types of faults give rise to different footprints in the transient, allowing identification of the fault type [20]. Different models have been proposed which adopt this methodology; including models based on random forests [20, 22], wavelet packet transforms [19] and convolutional neural networks [21].

A significant drawback of this methodology is that not all faults can be detected. Faults which cause a gradual reduction of power output, such as wiring degradation or potential induced degradation (PID), do not give rise to transient behavior and will therefore remain undetected. Moreover, a fault transient generally lasts at most a few seconds [20]. If the transient is missed because the monitoring is interrupted, the fault cannot be detected afterwards via these methods. Finally, the sensors required to gather high-frequency current and voltage measurements are expensive, leading to a high implementation cost for this methodology. For instance, the 1 kHz measurements used in [19] are collected by sensors which cost over €5000.

4) Comparing output of adjacent PV modules. Generally, the power produced by each module in a PV system is very similar, given that the modules have the same orientation. Therefore, each module can be monitored by comparing its output to the output of adjacent modules. For example, a convolutional neural network has been proposed in [23] to detect faulty modules in this manner. Similarly, a random forest classifier proposed in [24] is able to identify open circuit, shading and snow cover. Additionally, the random forest retains high accuracy when multiple modules are simultaneously suffering from faults. Note that all methods based on this methodology assume that the power production per module is known, meaning it is only suitable to monitor PV systems which are equipped with micro-inverters instead of the more common string inverter.

5) Comparing output of nearby PV systems. Much like adjacent PV modules, the power generated by geographically nearby PV systems is highly correlated. These correlations can be exploited to perform PV fault diagnosis using only the power produced by each PV system, which is readily available at the inverter. Hence, this methodology is compelling because it does not require weather measurements, reference modules, I-V curves, high-frequency measurements or micro-inverters, leading to significantly lower implementation costs compared to the approaches presented so far. An example of this methodology is presented in [25], where random forest regression is used to predict a PV system's output based on the output of nearby PV systems. Subsequently, faults are detected via a threshold on the difference between predicted and measured power output.

Table 1 summarizes the discussed methodologies and papers. Although the fault diagnosis accuracy is an impor-

Methodology	Ref.	Machine learning model	Inputs	Implementation cost	# identified fault types
Comparing expected and measured PV output	[7]	Multilayer perceptron	• I-V curve • Irradiance • Module temperature	High	6
	[8]	Ensemble model	• I_{mpp} & V_{mpp} • Irradiance • Module temperature	Medium	2
	[9]	Probabilistic neural network	• I_{mpp} & V_{mpp} • Irradiance • Module temperature	Medium	2
	[10]	Gradient boosted trees	• I_{mpp} & V_{mpp} • Irradiance • Ambient temperature	Medium	4
	[11]	Multilayer perceptron	• P_{mpp} • Irradiance • Ambient temperature	Medium	Detection only
	[12]	Multilayer perceptron	• High-frequency I_{mpp} & V_{mpp} • Irradiance • Ambient temperature	High	5
	[13]	Local and global consistency algorithm	• I_{mpp} & V_{mpp} • I & V of reference modules	Medium	2
	[14]	Recurrent neural network	• I_{mpp} & V_{mpp} • Satellite irradiance • Satellite ambient temperature	Low	6
I-V curve classification	[15]	Kernel extreme learning machine	• I-V curve	Medium	4
	[16]	Fuzzy classifier	• I-V curve	Medium	3
	[17]	Convolutional neural network	• I-V curve • Irradiance • Ambient temperature	High	4
	[18]	ResNet	• I-V curve • Irradiance • Ambient temperature	High	5
Identifying fault transients in I & V	[19]	Wavelet packet transforms	• High-frequency I_{mpp} & V_{mpp}	High	Detection only
	[20]	Multi-grained cascade forest	• High-frequency I_{mpp} & V_{mpp}	High	3
	[21]	Convolutional neural network	• High-frequency I_{mpp} & V_{mpp} • I & V of reference module	High	2
	[22]	Random forest	• High-frequency I per substring • High-frequency V_{mpp}	High	4
Comparing output of adjacent PV modules	[23]	Convolutional neural network	• P_{mpp} per module	Medium	Detection only
	[24]	Random forest	• P_{mpp} per module	Medium	3
Comparing output of nearby PV systems	[25]	Random forest	• P_{mpp}	Low	Detection only
	Proposed method	Graph neural network	• I_{mpp} & V_{mpp}	Low	6

Table 1: Overview of recently proposed PV fault diagnosis techniques based on machine learning.

tant aspect of the proposed methods, this is not included in the table, because the methods were all evaluated on different datasets. This prohibits an objective comparison of accuracy.

From the discussion above, it becomes apparent that comparing the output of a group of nearby PV systems provides an interesting approach to monitor PV systems without relying on expensive sensors, unlike most existing methods. Nevertheless, this approach is only followed by the method proposed in [25], which is limited to fault detection. As described in Section 2 and visualized in Fig. 1, a group of PV systems can be naturally represented as a graph. Since graph neural networks (GNNs) are the only type of deep learning models that can readily take graph structures as input [26], we believe that GNNs are well-suited for fault detection and identification of PV systems. Additionally, as described in Section 5, a trained GNN can take any number of PV systems as input, in contrast to other types of neural networks (fully-connected, convolutional, recurrent, etc.). This property enables monitoring of new PV systems without retraining the model. To the best of our knowledge, ours is the first work adopting GNNs for fault diagnosis of PV systems.

Section 2 outlines our technique, while Section 3 and Section 4 describe the collection and preprocessing of the data, respectively. Section 5 provides an overview of the GNN modeling principles. In Section 7, the performance of the developed model is compared to two state-of-the-art PV fault diagnosis methods, which are based on gradient boosted decision trees [10] and recurrent neural networks [14]. The architecture and training algorithm of the three models are presented in Section 6. The main contributions of this work are:

- A graph neural network (GNN) solution is developed to monitor a group of nearby PV systems. The proposed method is designed to have a minimal implementation cost: in contrast to existing PV fault diagnosis techniques, our model does not require any additional sensors beyond hourly current and voltage measurements, which are readily available from the inverter.

- The GNN accurately detects and identifies six relevant fault types by comparing the current and voltage produced by the PV systems during the past 24 h. We experimentally demonstrate that our GNN is an accurate and general fault diagnosis model, outperforming two recently proposed methods. Moreover, we show that the GNN remains accurate when multiple PV systems suffer from faults at the same time.

- By incorporating information about the distance and relative orientation between PV systems in the edge features of the GNN, our model learns to take into account the differences between the PV systems and can generalize to PV systems it was not trained on.

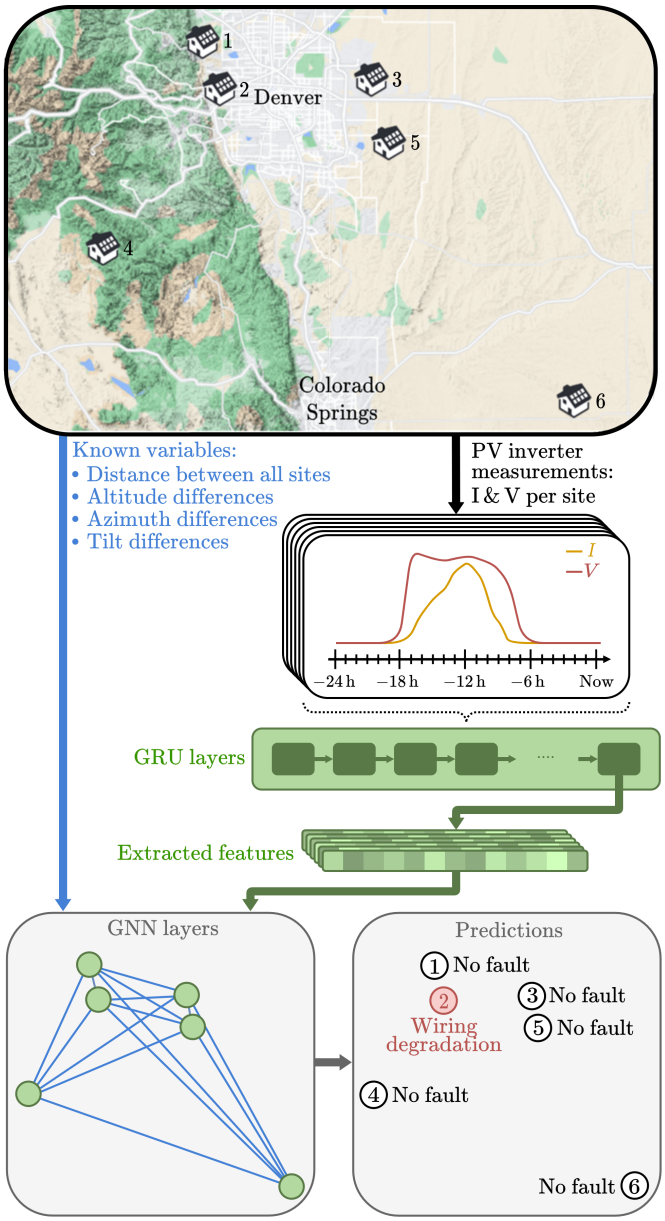


Figure 1: The proposed PV fault diagnosis methodology. In this figure, wiring degradation is identified in PV system 2, based on current and voltage measurements of all PV systems.

Hence, a trained GNN can be applied to new PV systems without requiring retraining. Combined with the low implementation cost, these properties make the proposed method especially cost-effective for residential PV systems.

2. Methodology

Our objective is to monitor a group of nearby PV installations by comparing their produced output. Besides the produced current and voltage of each PV system, other relevant variables include the distance, altitude difference and relative orientation between all PV systems. This highly structured data is most naturally represented as a graph (see Fig. 1), suggesting that GNNs are a suitable tool to monitor a group of PV systems.

The machine learning model is trained using supervised learning, where its parameters are optimized to produce the desired output corresponding to each input. Suppose that PV system 6 in Fig. 1 is experiencing a short circuit, while the other PV systems are working correctly. The desired outputs of the GNN are then 'no fault' for PV systems 1–5 and 'short circuit' for PV system 6. Machine learning models need to be trained on a large amount of data to offer high prediction accuracy. Since collecting sufficient data from faulty PV systems in the field is both expensive and potentially dangerous, we use physics-based PV simulations to generate large amounts of synthetic training data, as described in Section 3. After training, the GNN is evaluated on unseen testing data, i.e., data containing days or even PV systems which are not in the training data.

The proposed methodology, which is visualized in Fig. 1, compares the current and voltage produced during the last 24 h. Hourly current and voltage measurements can be readily collected from the PV inverters, meaning no sensors need to be installed. For each PV system, the 24 h window is first processed by recurrent neural network layers, allowing to take the temporal nature of the data into account. The features extracted from the current and voltage of site i are then given to the graph node corresponding to site i. The edge features of the graph are based on the orientations and geographical locations of the PV systems. Therefore, these are indicated as 'known variables' in Fig. 1. Specifically, each edge contains the distance and altitude difference between the two sites it connects. Furthermore, each edge also contains the difference in tilt and azimuth between the two PV systems. The constructed graph is used as input for the GNN layers. The output of the model consists of the predicted status of each PV system: either no fault or one of the six considered faults.

3. Simulated PV systems and faults

Since collecting large amounts of real data from faulty PV arrays is infeasible, we generate synthetic training

Site	PV module type	Number of modules (series × parallel)	Tilt	Azimuth
1	SW 325 XL duo	6 × 3	15°	180°
2	Scheuten P6-60 i30	15 × 1	25°	90°
3	Scheuten P6-60 i30	10 × 2	35°	135°
4	SW 325 XL duo	12 × 1	45°	270°
5	SW 325 XL duo	8 × 2	30°	225°
6	Scheuten P6-60 i30	4 × 3	20°	160°

Table 2: Configuration and orientation of each PV system. The numbering of sites corresponds to the map in Fig. 1.

Module parameter	SW 325 XL duo	Scheuten P6-60 i30
Number of cells (series × parallel)	24 × 3	20 × 3
Maximum power (P_{mpp})	325 W	230 W
Maximum power point voltage	37.7 V	29.3 V
Maximum power point current	8.68 A	7.84 A
Open circuit voltage (V_{OC})	47.0 V	37.2 V
Short circuit current (I_{SC})	9.28 A	8.31 A
Temperature coefficient of P_{mpp}	−0.43 %/K	−0.42 %/K
Temperature coefficient of V_{OC}	−0.31 %/K	−0.30 %/K
Temperature coefficient of I_{SC}	0.044 %/K	0.040 %/K

Table 3: PV module parameters at standard test conditions of the two module types in Table 2.

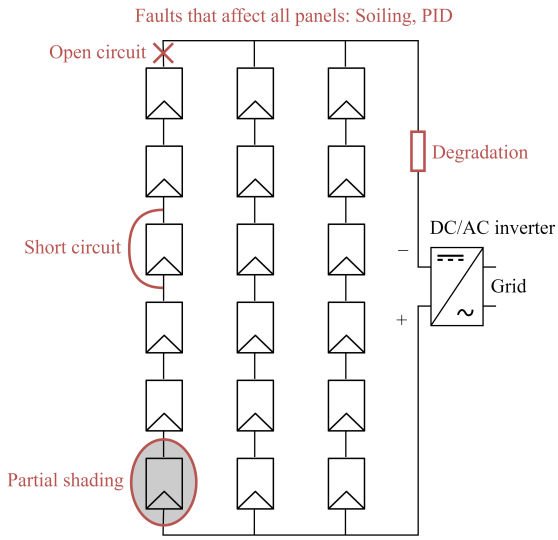


Figure 2: The PV system of site 1 and the investigated faults. Figure adapted from [14].

Fault type	Description	Simulated severities
Open circuit	Disconnection in the wiring.	Disconnection of a (sub)string.
Short circuit	Accidental low-resistance path between two points in the PV system.	Short circuit of 1, 2, 3 or 4 modules in a (sub)string.
Wiring degradation	Increased series resistance of PV modules.	Set resistor in Fig. 2 to 5Ω, 10Ω, 15Ω or 20Ω.
Partial shading	Local shading cast by clouds, chimneys, trees, etc.	Reduced irradiance of 1, 2, 3 or 4 modules in a (sub)string by 50% during low sun (zenith larger than 60°).
Soiling	Accumulation of dust on the surface of PV modules.	Reduced irradiance of all modules by 5%, 10%, 15% or 20%.
PID (shunting type)	Electrochemical degradation due to large voltage differences, leading to leakage current between PV cells and the array's frame. The PID simulation is described in [27].	PID severities corresponding to a 5%, 10%, 15% or 20% loss of average power output.

Table 4: Description of the PV system faults considered in this work.

data via physics-based PV simulations [28]. As described in [28], the PV simulation is based on the well-established single-diode model. The accuracy of the PV simulation has been experimentally validated in [28, 29], where the power output predictions are compared to measurements of a residential PV system in Oldenburg, Germany. The predicted power output is significantly more accurate than predictions of the commercial simulation tool PVsyst [30], illustrating that the employed PV simulation effectively models real PV systems [28, 29].

To provide realistic weather input for the simulations, we let each site in Fig. 1 correspond to a measurement station of the National Renewable Energy Laboratory (NREL). These weather measurements are publicly available [31] and were collected throughout 2012–2018 for sites 1–4 and 2012 for sites 5 and 6. Days with missing weather measurements for one or more sites were removed from the dataset. Over the 6 years, 168 days were removed in total.

Generally, a group of residential PV systems will have varying orientations, number of PV modules, electrical configurations and module types. Hence, different values for these parameters are randomly chosen for each site, within reasonable bounds. The PV system configuration and orientation for each site are listed in Table 2. The characteristics of the 2 module types used in this work are detailed in Table 3.

The PV system at site 1 is visualized in Fig. 2, along with the six fault types which are considered in this work. These faults are described in Table 4. For each fault type, four severities are included in the dataset to ensure that faults cannot be distinguished by simply comparing their severities. Where possible, the fault severities are chosen to be in the 5%–20% range. Hence, the models must be able to identify faults that reduce the power output by as little as 5%, which is challenging. However, such early identification of faults is crucial to maximize the efficiency and lifetime of PV systems. Note that only one severity of open circuit is considered because all modules of PV systems 2 and 4 are connected in series, so there is only one string which can be disconnected.

4. Data preprocessing

Most PV systems are not equipped with dedicated current and voltage sensors. Hence, we assume that the produced current and voltage will be measured by the inverter to minimize the implementation costs. As inverter measurements can contain significant measurement errors, we include 5% uniformly distributed noise in the simulated current and voltage.

Many machine learning models assume that input features have similar scales [32], e.g., that the values of all input features lie within the interval $[0, 1]$. The input features of the proposed GNN model are listed in Fig. 1. In this work, we divide each input feature by its nominal maximum value so their scales become comparable. The maximum current and voltage produced by a PV system highly depend on the number of modules, the module characteristics and how the modules are connected. This is taken into account when normalizing the current and voltage, allowing the model to generalize to different PV system configurations. Following [20], the current and voltage produced by PV system i are normalized via

$$\mathrm{I}' = \frac{\mathrm{I}}{p_i \cdot \mathrm{I}_{\mathrm{SC},\,i}}$$
$$\mathrm{V}' = \frac{\mathrm{V}}{s_i \cdot \mathrm{V}_{\mathrm{OC},\,i}}\,,$$

where p_i and s_i represent the number of substrings and the number of panels in series of PV system i, respectively, while $\mathrm{I}_{\mathrm{SC},\,i}$ and $\mathrm{V}_{\mathrm{OC},\,i}$ are the short circuit current and open circuit voltage of the modules in PV system i. The normalization is performed for each timestep, so the entire current and voltage time series are normalized. All other input features are normalized by simply dividing by the constants defined in Table 5.

Input feature	Constant
Distance	100 km
Altitude difference	2 km
Azimuth difference	360°
Tilt difference	90°
Irradiance	1000 W/m^2
Temperature	50 °C
Zenith	360°

Table 5: Normalization constants of the input features. Note that irradiance, temperature and zenith are only used when satellite weather data is included as input (see Section 7.1).

Each hour, the GNN model predicts the status of all PV systems based on measurements during the past 24 h (see Fig. 1). This is achieved by sliding a window of 24 h over the input time series. Note that we do not consider transitions from, e.g., fault-free conditions to suddenly appearing faults within a 24 h sample. Including such samples with transitions is left for future work.

The GNN model receives the current and voltage produced by all PV systems as input. We are free to choose in which PV systems we introduce faults to train and test the model. We first include at most one faulty PV system as follows. Each time the sliding window slides to a new hour, one PV system is randomly selected to have its output reduced by a fault during that 24 h window. This procedure is repeated for each fault type and severity, where the same weather data is reused every time. Note that we also include the case where all PV systems operate normally throughout the 6 years. Second, we will explore the GNN's performance when faults occur in multiple PV systems simultaneously. This will be discussed in Section 7.4.

5. Graph neural network background

As outlined in Section 2, GNNs provide a promising model architecture for fault diagnosis of a group of PV systems. In this work, we consider undirected graphs defined by the tuple $(\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V}$ is the set of nodes $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{N_v}\}$, where N_v is the number of nodes and $\boldsymbol{v}_n$ represents the features of node n. Likewise, $\mathcal{E}$ represents the set of edges $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{N_e}\}$, where N_e is the number of edges and $\boldsymbol{e}_j$ represents the features of edge j, such as the distance between two PV systems.

During training, a GNN layer updates both the edge and node features of the graph, as visualized in Fig. 3. First, the features of each edge $\boldsymbol{e}_j$ are updated to new values $\boldsymbol{e}'_j$ based on the features of the two nodes $\boldsymbol{v}_{l_j}$ and $\boldsymbol{v}_{r_j}$ which the edge j connects:

$$\boldsymbol{e}'_j = f_e\left(\boldsymbol{e}_j, \boldsymbol{v}_{l_j}, \boldsymbol{v}_{r_j}\right) \quad \text{with} \quad j = 1, \ldots, N_e \tag{1}$$

Note that each edge is updated using the same function f_e, which is often a multilayer perceptron. Similarly, each node $\boldsymbol{v}_n$ is then updated using a second multilayer perceptron f_v. Let S_n be the set of edges which are connected to node n. The features of each edge $\boldsymbol{v}_n$ are updated to new values $\boldsymbol{v}'_n$ based on E'_n, which is the aggregated features of the edges that are connected to node n.

$$\boldsymbol{v}'_n = f_v\left(\boldsymbol{v}_n, E'_n\right) \quad \text{with} \quad E'_n = \sum_{k \in S_n} \boldsymbol{e}'_k \quad \text{and} \quad n = 1, \ldots, N_v \tag{2}$$

Note that the edge features can be aggregated by any function which is invariant to permutations of its inputs and can take any number of inputs [26]. The latter requirement stems from the fact that a node can be connected to an arbitrary number of edges. Hence, the summation in Eq. (2) can be replaced by mean, maximum, etc.

Since f_e and f_v are applied to, respectively, all edges and nodes in a graph, these functions can be applied to any graph. Consequently, a single GNN can be trained on graphs with varying number of nodes and / or edges. Moreover, GNNs can make predictions for graphs which were not in its training data. This property will be exploited in Section 7.3 to monitor PV systems which are not in the training data.

6. Machine learning models

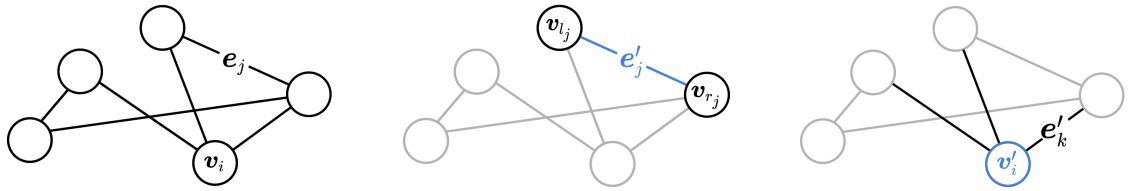
The accuracy of the proposed GNN model is compared to two state-of-the-art models, namely CatBoost [33] and stacked GRU (gated recurrent unit) [14]. The three models are visualized in Fig. 4. CatBoost is a machine learning model based on gradient boosted decision trees and was recently proposed for PV fault diagnosis [10]. The stacked GRU model is a recurrent neural network developed for PV fault diagnosis [14]. In the following, the structure and training algorithm of these three models are detailed.

6.1. Stacked GRU model architecture

The GRU model introduced in [14] consists of a GRU block with two stacked GRU layers and a residual connection over the second layer, followed by two fully-connected layers with ReLu and softmax activation functions, respectively. Both GRU layers have a hidden dimension of 64 with 50% recurrent dropout and layer normalization. The first fully-connected layer has hidden dimension 128 , while the fina layer has output dimension 7 (i.e., the number of output classes). Note that the GRU model from [14] also includes dedicated layers to estimate the fault severity. However, we omit these severity layers because severity prediction is not considered in this work.

6.2. GNN model architecture

The proposed GNN model is shown in Fig. 4. Since its inputs include time series, the GRU block is again used to automatically extract features from the 24 h windows of produced current and voltage. Note that the same GRU block is applied consecutively to the 24 h window of each PV system, i.e., the parameters of the GRU block are



(a) Original graph (b) Update each edge via Eq. (1) (c) Update each node via Eq. (2)

Figure 3: Operations performed by a GNN layer. In (b), the blue edge is updated with f_e, which uses the black nodes as inputs. This is performed for each edge in the graph. Similarly, (c) shows the update of the blue node using the sum of the features of the connected edges as input. Figure adapted from [26].

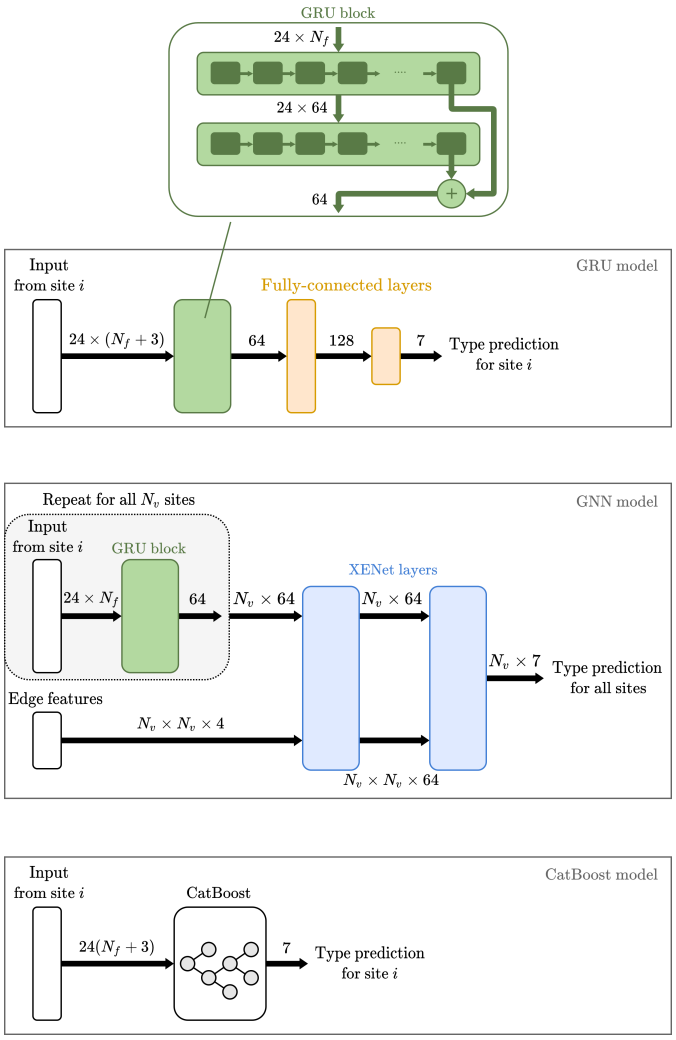


Figure 4: Schematic representation of the models' architectures. Note that in the GNN model, the same GRU block is used to process each site. The dimensions of the passed feature tensors are denoted next to each arrow, where N_v is the number of sites and N_f is the number of input features of the GNN. More details regarding the input dimensions of the models are given in Section 6.4.

shared across PV systems. This ensures that the model can still take any number of PV systems as input, even when the model is already trained (see also Section 5). Without the GRU layers (i.e., when the flattened input vector of length $24N_f$ is fed directly into the first GNN layer), the model cannot properly take the time dimension of the data into account, resulting in an accuracy loss of roughly 8%.

The two fully-connected layers from [14] are now replaced by two XENet layers, which is a particular type of GNN layer [34]. The first XENet layer takes the features extracted from the 24 h windows of all PV systems as input, where the features from PV system i are given to the graph node corresponding to PV system i. The input edge features are straightforwardly determined from the orientations and geographical locations of the PV systems and remain constant over time. Since we represent a group of PV systems as a graph, we can freely choose which nodes in the graph to connect with edges. For simplicity, we only construct complete graphs, i.e., we connect each node to all other nodes. This means that the number of edges scales as $\mathcal{O}(N_v^2)$ for a graph with N_v nodes, which can become prohibitively large if many PV systems are monitored. In that case, one could opt to only connect nodes if the distance is lower than a suitable threshold.

XENet was chosen over more well-known types of GNN, such as graph convolutional networks (GCN) [35], because XENet supports edge features, unlike GCN. Moreover, for the task at hand, XENet outperforms other GNN types which support edge features, such as edge-conditioned convolutional networks [36] and crystal graph convolutional networks [37]. The XENet implementation of the Python package Spektral [38] was used. The first XENet layer has hidden dimension 64, ReLu activation and 20% dropout, while the final XENet layer has output dimension 7 with softmax activation.

6.3. CatBoost

CatBoost is a state-of-the-art gradient boosting model proposed in [33] and applied to the PV fault diagnosis task in [10]. CatBoost performs ordered boosting of decision trees to prevent overfitting and reduce computational complexity. The default values of CatBoost's hyperparameters, as defined in [33], achieved the highest accuracy.

6.4. Input of the models

A key difference between the three models is the structure and dimensions of their input. The GNN first processes the 24 h windows of N_v PV systems one by one using its GRU layers. Each 24 h window has dimensions $24 \times N_f$, where N_f is the number of input features. Hence, N_f is 2 when only the produced current and voltage are given as input and 5 when satellite weather data is also included (see Section 7.1). The extracted features of all PV systems are then given to the nodes of the GNN layer, while each edge receives the distance and difference in altitude, azimuth and tilt between the two PV systems it connects. Since each node has an edge with all other nodes, the dimensions of the input edge features are $N_v \times N_v \times 4$.

The GRU model operates on a single PV system at a time and is unable to take the distance and difference in altitude, azimuth and tilt between the monitored PV systems into account. To make the comparison with GNN fair, we include each PV system's altitude, azimuth and tilt in the input of the GRU model. Therefore, an input sample of the GRU model is a $24 \times (N_f + 3)$ matrix. As CatBoost does not take the time dimension of the data into account, its input is a vector of size $24(N_f + 3)$.

Since GRU operates on a single PV system at a time, a different model could be trained for each PV system. However, training one GRU model on data of all PV systems results in a very similar performance. Moreover, the latter approach performs much better when the GRU model is tested on PV systems it was not trained on (see Section 7.3), since the model has already seen multiple different PV systems and thus generalizes better. The same holds true for CatBoost. Therefore, both models are trained on data from all PV systems.

6.5. Training algorithm

Since identifying faults in PV systems is essentially a classification problem, the models are trained by minimizing the cross-entropy loss [32]. The loss function is weighed to compensate for the class imbalance in the data. Specifically, four severities of each fault type are included in the data, except for no fault and open circuit, which only have one configuration (see Section 3). Accordingly, samples with no fault and open circuit have weight 4 in the loss function of CatBoost and GRU, while all other samples have weight 1. For GNN, the no fault class is significantly more prominent in the data because at most one of the PV systems is faulty, as described in Section 4. Therefore, the class imbalance in the loss of GNN is compensated by assigning weight 0.06 to no fault samples, which is the inverse of the prevalence of the no fault class. For open circuit, this is 4, as before. Note that all accuracies reported in this work are also balanced using these weights to obtain a representative view of the models' performance.

Both neural networks are trained using batch size 64 and the Adam optimizer [39]. During training, the learning rate $\gamma(t)$ at optimization step t is gradually decreased as follows

$$\gamma(t) = \frac{\gamma_0}{1 + \lambda t} = \frac{2 \cdot 10^{-3}}{1 + 5 \cdot 10^{-5}\, t}\,.$$

All models are thoroughly evaluated using 5-fold cross-validation. In each fold, a different year of the data is kept separate as test set, which is solely used to evaluate the trained model. For the remaining data, the order of the days is shuffled and 100 days are randomly selected as validation set, while the rest is used to train the model. The validation set is used for hyperparameter tuning and early stopping, where training is stopped prematurely if the accuracy on the validation set failed to improve compared to the previous training epoch. This can help prevent overfitting [32].

7. Results and discussion

7.1. Performance with satellite weather data

First, we compare the accuracy of the models when both satellite weather estimations and the produced current and voltage are given as input. The input time series now consists of: I, V, irradiance, temperature and solar zenith. This is essentially the same setting the GRU model was originally tested in [14]. The satellite weather data of Denver is obtained from MERRA-2 and is freely available [40]. The satellite estimations of irradiance and temperature are for the entire state of Denver, not for the specific locations of the PV systems. Consequently, the models receive the same satellite estimations for each site, which can significantly deviate from the true weather conditions. For instance, the mean absolute error between the satellite-derived irradiance and the true irradiance at site 1 is 51.1 $\mathrm{W/m^2}$.

After training the models until convergence, which took roughly 8 h for CatBoost and 1.5 h for both GRU and GNN, the results reported in Table 6 were obtained. The models were trained and tested on data from PV systems 1–4, for which we have 6 years of data. The results on PV systems 5 and 6 will be discussed in Section 7.3. Table 6 shows the accuracy averaged over sites 1–4 because the per-site accuracies are all very similar. For instance, the highest per-site accuracy GNN reached over all cross-validation iterations is 88.8% for site 4, while the lowest accuracy is 86.3% for site 1.

The confusion matrices of the models for the first cross-validation fold are shown in Fig. 5. GNN significantly outperforms both CatBoost and stacked GRU, especially when distinguishing between no fault and soiling. As noted in [14], this is difficult for GRU and CatBoost due to the inaccuracy of the satellite weather estimation. Specifically, both irradiance overestimations by the satellite and soiling give rise to an irradiance input for the models which is larger than what the modules are exposed to in reality. The GNN is superior in distinguishing these similar effects because it is able to compare the output of multiple PV

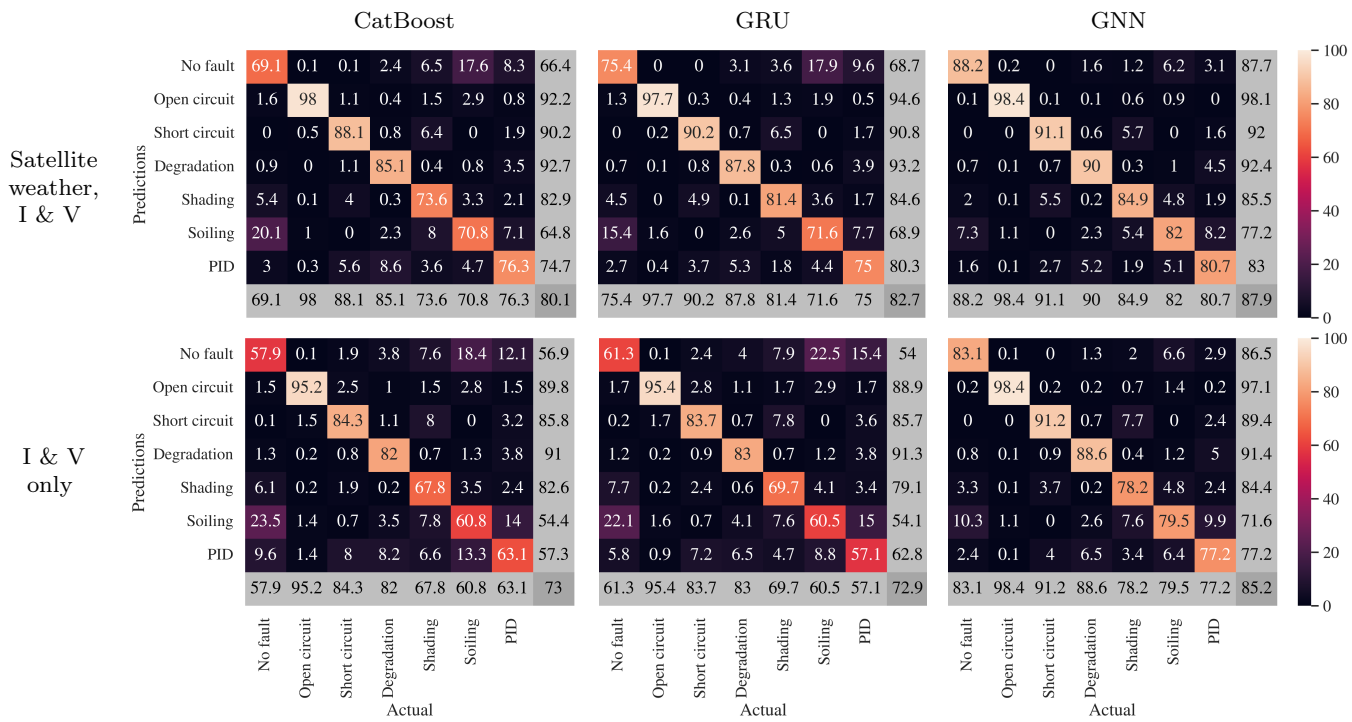


Figure 5: Confusion matrix of the models on the test set. In the top row, satellite weather data is included in the input. Conversely, in the bottom row, only the produced current and voltage are given as input. For each fault, the reported accuracy percentages are averaged over all severities. The grey column and row represent the precision and recall, respectively.

	Accuracy (%)		
Inputs	CatBoost	GRU	GNN
Satellite weather, I & V	79.8 ± 2.4	82.3 ± 2.9	87.5 ± 1.6
I & V only	73.0 ± 2.4	72.7 ± 2.7	84.6 ± 2.1

Table 6: Average accuracy on sites 1–4 over the 5-fold cross-validation, with 3 times the standard deviation over the cross validation folds as error margins.

systems. Irradiance overestimations cause all PV systems to produce less than expected, whereas soiling is more likely to affect a limited number of PV systems.

7.2. Performance without satellite weather data

Thanks to the GNN's ability to compare the output of PV systems, it should be able to operate without any weather input. This task is complicated by the fact that each PV system has a different orientation, meaning that the output of the PV systems at a given hour cannot be readily compared. For instance, the output of a southern facing PV system will peak earlier in the day than the output of a western facing system. By giving both 24 h windows and the relative orientations of the PV systems as input, GNN can learn to adjust for this complication.

To explore the potential to operate without weather information, we only provide the produced current and voltage as input time series to the models. The results are also shown in Table 6. Note that GNN without weather input outperforms CatBoost and GRU even when the latter do have access to satellite weather data. The confusion matrices of the models without weather inputs are presented in Fig. 5.

As expected, the accuracy loss for GNN is limited. Although CatBoost and GRU perform significantly worse without weather data, they are still able to reach over 70% accuracy. This is surprising, considering that both models cannot compare a PV system's output to weather data nor to the output of other PV systems. A possible explanation is that CatBoost and GRU infer 'expected' current and voltage time series from the training data, such as the average production of a PV system without faults. The models can then try to predict the fault type by comparing its input with its expected current and voltage. Note that these expectations could be tuned to the season: during winter, 24 h windows will contain more hours without power production because the days are shorter. Hence, depending on the number of hours without power production, the models can expect higher current and voltage production during summer than during winter. This could explain why CatBoost and GRU do not fail completely without satellite weather input.

7.3. Unseen PV systems

As described in Section 5 and Section 6.2, a trained GNN can make predictions for any graph, and can thus also be applied to topologies that differ from those in the training data. Here, we exploit this property to monitor

Inputs	Site	Accuracy (%)		
		CatBoost	GRU	GNN
Satellite weather, I & V	5	65.4	79.5	86.8
	6	57.6	70.0	62.2
I & V only	5	59.6	64.0	83.7
	6	55.6	65.5	62.2

Table 7: Accuracy on sites 5 and 6, which are not in the training or validation set of the models.

unseen PV systems without retraining. For this purpose, the GNN is trained on a graph of sites 1–4 using 5 years of data, as before, while it is now tested on a graph of all 6 sites. Because only data of 2012 is available for sites 5 and 6, this is the only year suitable as test set, meaning cross-validation is not possible. Table 7 shows the resulting accuracy on the test set, both for the case with and without satellite weather data. Note that even though the GNN makes predictions for all 6 sites, only the accuracies on the unseen sites 5 and 6 are reported in Table 7. Similarly, the reported CatBoost and GRU accuracies are for models trained on sites 1–4 and tested on sites 5 and 6.

These results show that GNN trained on sites 1–4 successfully generalizes to site 5, where it reaches an accuracy similar to its accuracy on sites 1–4. This is especially impressive considering that the tilt, azimuth, number of modules and module configuration of site 5 are all different compared to sites 1–4. The GNN also outperforms CatBoost and GRU for site 5. However, GNN performs significantly worse for site 6, where GRU achieves the highest accuracy. This is likely due to the relatively large distance between site 6 and the other sites: GNN assumes that the weather experienced by all PV systems is comparable, which is not necessarily the case for site 6 since its nearest site is 196 km away (see Fig. 1). Hence, GNN can be applied to new PV systems without retraining, as long as the new PV system is not too isolated from the others.

7.4. Multiple faulty PV systems

So far, we have assumed that at most one of the PV systems is suffering from a fault. However, faults can occur simultaneously in different PV systems, especially when a large number of PV systems is being monitored by the GNN. Therefore, we loosen the above assumption and adjust the data so that at least one PV system operates normally, while all other PV systems suffer from randomly chosen fault types and severities. Note that this is only relevant for the GNN, as the other models operate on one PV system at a time. In this setting, the GNN reaches $86.1\% \pm 2.2\%$ accuracy with satellite weather data and $82.6\% \pm 2.5\%$ accuracy without satellite data. Although these accuracies are lower than for the case with at most one faulty PV system, as reported in Table 6, the deviation does not exceed the error margins of the cross-validation. Furthermore, we considered an extreme case where nearly all PV systems in the group are simultaneously faulty, which is unlikely to occur in practice. These results demonstrate that the GNN retains high accuracy even when multiple PV systems suffer from faults simultaneously.

7.5. Limitations of the proposed method

When developing a fault diagnosis method, it is key to consider the trade-off between accuracy and implementation cost. Here, we have chosen to prioritise low implementation cost over accuracy to obtain a method that is applicable and cost-effective for both small and large-scale PV systems. Therefore, one of our objectives was to only use inverter measurements as input, instead of relying on dedicated sensors. Clearly, our GNN solution can be outperformed by methods that do rely on dedicated sensors, which are reported to reach accuracies of over 99% [9, 15, 22].

In contrast to CatBoost, GRU and most fault diagnosis models in literature, GNN does not perform well for an isolated PV system: in our test case, GNN fails for site 6 because it is over 196 km away from the other sites, as described in Section 7.3. However, such isolated PV systems are only common in sparsely populated regions. Moreover, if an isolated PV system includes multiple inverters, GNN can be applied by representing each string as a node in the graph, so that the model performs fault diagnosis by comparing the output of the strings. Therefore, our model is also suitable to monitor a single, large-scale PV system. In this scenario, the GNN is likely to be highly accurate, considering all strings experience essentially the same weather when they are next to each other.

Although a single GNN can monitor any number of PV systems, we have tested the model on a limited number of PV systems for now. Therefore, it is not yet clear how well the proposed method scales when implemented to hundreds or thousands of PV systems, located around, e.g., a given city. Nevertheless, it is likely that the accuracy of GNN further rises as the number of monitored PV systems increases, because the output of a PV system can be compared to additional nearby PV systems.

Finally, the GNN model predicts a single fault type per PV system. However, a PV system could suffer from, e.g., both wiring degradation and PID at the same time. Although GNN would likely classify this PV system as either wiring degradation or PID, it cannot identify both faults simultaneously. To address this limitation, the GNN should be extended to perform multi-label classification, allowing to identify simultaneously occurring faults. This is left for future work.

8. Conclusions

A graph neural network (GNN) is proposed for detecting and identifying faults in photovoltaic (PV) systems by comparing their produced current and voltage. As outlined in the literature overview, state-of-the-art PV fault

diagnosis methods still rely on expensive on-site sensors. The proposed methodology is able to monitor PV systems based solely on current and voltage measurements from the inverter. Since no sensors need to be installed, this methodology is cost-effective and straightforward to implement.

The GNN model is extensively validated and compared to existing state-of-the-art methods. The results show that the GNN is able to accurately monitor a group PV systems with widely varying configurations and orientations. The model even generalizes to PV systems which were not in the training data, provided that all PV systems are located sufficiently close to each other (e.g., near the same city) so they are exposed to similar weather conditions. Consequently, a single GNN could monitor all PV systems in a city and does not necessarily need to be retrained when new PV systems are included. Finally, we demonstrated that the model is also robust to the case where multiple PV systems are simultaneously faulty.

Acknowledgements

This work was supported by the DAPPER project, which is financed by Flux50 and Flanders Innovation & Entrepreneurship (project number HBC.2020.2144).

References

[1] A. Arneth, H. Barbosa, T. Benton, K. Calvin, E. Calvo, S. Connors, A. Cowie, E. Davin, F. Denton, R. van Diemen, *et al.*, "IPCC special report on climate change, desertification, land degradation, sustainable land management, food security, and greenhouse gas fluxes in terrestrial ecosystems," *Summary for Policy Makers. Geneva: Intergovernmental Panel on Climate Change (IPCC)*, 2019.

[2] IEA (2021), "World energy outlook 2021." https://www.iea.org/reports/world-energy-outlook-2021.

[3] S. R. Madeti and S. Singh, "Online fault detection and the economic analysis of grid-connected photovoltaic systems," *Energy*, vol. 134, pp. 121–135, 2017.

[4] A. Livera, M. Theristis, G. Makrides, and G. E. Georghiou, "Recent advances in failure diagnosis techniques based on performance data analysis for grid-connected photovoltaic systems," *Renewable energy*, vol. 133, pp. 126–143, 2019.

[5] D. S. Pillai and N. Rajasekar, "A comprehensive review on protection challenges and fault diagnosis in PV systems," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 18–40, 2018.

[6] A. Mellit, G. M. Tina, and S. A. Kalogirou, "Fault detection and diagnosis methods for photovoltaic systems: A review," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 1–17, 2018.

[7] W. Chine, A. Mellit, V. Lughi, A. Malek, G. Sulligoi, and A. M. Pavan, "A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks," *Renewable Energy*, vol. 90, pp. 501–512, 2016.

[8] C. Kapucu and M. Cubukcu, "A supervised ensemble learning method for fault diagnosis in photovoltaic strings," *Energy*, vol. 227, p. 120463, 2021.

[9] E. Garoudja, A. Chouder, K. Kara, and S. Silvestre, "An enhanced machine learning based approach for failures detection and diagnosis of PV systems," *Energy conversion and management*, vol. 151, pp. 496–513, 2017.

[10] D. Adhya, S. Chatterjee, and A. K. Chakraborty, "Performance assessment of selective machine learning techniques for improved PV array fault diagnosis," *Sustainable Energy, Grids and Networks*, vol. 29, p. 100582, 2022.

[11] M. De Benedetti, F. Leonardi, F. Messina, C. Santoro, and A. Vasilakos, "Anomaly detection and predictive maintenance for photovoltaic systems," *Neurocomputing*, vol. 310, pp. 59–68, 2018.

[12] L. L. Jiang and D. L. Maskell, "Automatic fault detection and diagnosis for photovoltaic systems using combined artificial neural network and analytical based methods," in *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2015.

[13] Y. Zhao, R. Ball, J. Mosesian, J.-F. de Palma, and B. Lehman, "Graph-based semi-supervised learning for fault detection and classification in solar photovoltaic arrays," *IEEE Transactions on Power Electronics*, vol. 30, no. 5, pp. 2848–2858, 2014.

[14] J. Van Gompel, D. Spina, and C. Develder, "Satellite based fault diagnosis of photovoltaic systems using recurrent neural networks," *Applied Energy*, vol. 305, p. 117874, 2022.

[15] Z. Chen, L. Wu, S. Cheng, P. Lin, Y. Wu, and W. Lin, "Intelligent fault diagnosis of photovoltaic arrays based on optimized kernel extreme learning machine and I–V characteristics," *Applied Energy*, vol. 204, pp. 912–931, 2017.

[16] S. Spataru, D. Sera, T. Kerekes, and R. Teodorescu, "Diagnostic method for photovoltaic systems based on light I–V measurements," *Solar Energy*, vol. 119, pp. 29–44, 2015.

[17] Q. Liu, B. Yang, Z. Wang, D. Zhu, X. Wang, K. Ma, and X. Guan, "Asynchronous decentralized federated learning for collaborative fault diagnosis of PV stations," *IEEE Transactions on Network Science and Engineering*, 2022.

[18] P. Lin, Z. Qian, X. Lu, Y. Lin, Y. Lai, S. Cheng, Z. Chen, and L. Wu, "Compound fault diagnosis model for photovoltaic array using multi-scale SE-ResNet," *Sustainable Energy Technologies and Assessments*, vol. 50, p. 101785, 2022.

[19] B. P. Kumar, G. S. Ilango, M. J. B. Reddy, and N. Chilakapati, "Online fault detection and diagnosis in photovoltaic systems using wavelet packets," *IEEE Journal of Photovoltaics*, vol. 8, no. 1, pp. 257–265, 2017.

[20] W. Gao, R.-J. Wai, and S.-Q. Chen, "Novel PV fault diagnoses via SAE and improved multi-grained cascade forest with string voltage and currents measures," *IEEE Access*, vol. 8, pp. 133144–133160, 2020.

[21] X. Lu, P. Lin, S. Cheng, Y. Lin, Z. Chen, L. Wu, and Q. Zheng, "Fault diagnosis for photovoltaic array based on convolutional neural network and electrical time series graph," *Energy Conversion and Management*, vol. 196, pp. 950–965, 2019.

[22] Z. Chen, F. Han, L. Wu, J. Yu, S. Cheng, P. Lin, *et al.*, "Random forest based intelligent fault diagnosis for PV arrays using array voltage and string currents," *Energy conversion and management*, vol. 178, pp. 250–264, 2018.

[23] T. Huuhtanen and A. Jung, "Predictive maintenance of photovoltaic panels via deep learning," in *2018 IEEE Data Science Workshop (DSW)*, pp. 66–70, IEEE, 2018.

[24] M. Feng, N. Bashir, P. Shenoy, D. Irwin, and D. Kosanovic, "Sundown: Model-driven per-panel solar anomaly detection for residential arrays," in *Proceedings of the 3rd ACM SIGCAS Conference on Computing and Sustainable Societies*, pp. 291–295, 2020.

[25] S. Iyengar, S. Lee, D. Sheldon, and P. Shenoy, "Solarclique: Detecting anomalies in residential solar arrays," in *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, pp. 1–10, 2018.

[26] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[27] A. Schils, R. Breugelmans, J. Carolus, J. Ascencio-Vásquez, A. Wabbes, E. Bertrand, B. Aldalali, M. Daenen, E. Voroshazi, and S. Scheerlinck, "A grey box model for shunting-type

potential induced degradation in silicon photovoltaic cells under environmental stress," European Photovoltaic Solar Energy Conference and Exhibition, 2021.

[28] H. Goverde, B. Herteleer, D. Anagnostos, G. Köse, D. Goossens, B. Aldaladi, *et al.*, "Energy yield prediction model for PV modules including spatial and temporal effects," in *European Photovoltaic Solar Energy Conference and Exhibition*, 2014.

[29] D. Anagnostos, H. Goverde, B. Herteleer, F. Catthoor, S. Dimitrios, J. Driesen, *et al.*, "Demonstration and validation of an energy yield prediction model suitable for non-steady state non-uniform conditions," in *The 6th World Conference on Photovoltaic Energy Conversion*, 2014.

[30] A. Mermoud, C. Rocker, and J. Bonvin, "Pvsyst (version 6.23)," *Computer Software. Geneva, Switzerland: ISE, University of Geneva. Retrieved May*, vol. 1, p. 2014, 1994.

[31] D. Jager and A. Andreas, "NREL National Wind Technology Center (NWTC): M2 Tower; Boulder, Colorado (Data)." NREL Report No. DA-5500-56489, 1996. http://dx.doi.org/10.5439/1052222.

[32] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[33] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018.

[34] J. B. Maguire, D. Grattarola, V. K. Mulligan, E. Klyshko, and H. Melo, "Xenet: Using a new graph convolution to accelerate the timeline for protein design on quantum computers," *PLoS computational biology*, vol. 17, no. 9, p. e1009037, 2021.

[35] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[36] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3693–3702, 2017.

[37] T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," *Physical review letters*, vol. 120, no. 14, p. 145301, 2018.

[38] D. Grattarola and C. Alippi, "Graph neural networks in tensorflow and keras with spektral [application notes]," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 99–106, 2021.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[40] S. Pfenninger and I. Staffell, "Long-term patterns of european PV output using 30 years of validated hourly reanalysis and satellite data," *Energy*, vol. 114, pp. 1251–1265, 2016. https://www.renewables.ninja/.