

Neural network based approaches for fault diagnosis of photovoltaic systems^{★,★★}

Jonas Van Gompel, Domenico Spina, and Chris Develder

IDLab, Dept. of Information Technology, Ghent University – imec,

Technologiepark Zwijnaarde 126, 9052 Gent, Belgium

{jonas.vangompel, domenico.spina, chris.develder}@ugent.be

<https://UGentAI4E.github.io>

Abstract. Faults in photovoltaic (PV) systems due to manufacturing defects and normal wear and tear are practically unavoidable. The effects thereof range from minor energy losses to risk of fire and electrical shock. Thus, several PV fault diagnosis techniques have been developed, usually based on dedicated on-site sensors or high-frequency current and voltage measurements. Yet, implementing them is not economically viable for common small-scale residential systems. Hence, we focus on cost-effective techniques that enable introducing fault diagnosis without incurring costs for on-site sensor systems. In this chapter, we will present in particular two machine-learning-based approaches, built on recent neural network models. The first technique relies on recurrent neural networks (RNNs) using satellite weather data and low-frequency inverter measurements for accurate fault detection, including severity estimation (i.e., the power loss caused by the fault, usually not quantified in state-of-the-art methods in literature). The second technique is based on graph neural networks (GNNs), which we use to monitor a group of PV systems by comparing their current and voltage production over the last 24 h. By comparing outputs from multiple (geographically nearby) PV installations, we avoid any need for additional sensor data. Moreover, our results suggest that the GNN-based model can generalize to PV systems it was not trained on (as long as nearby sites are available) and retains high accuracy when multiple PV systems are simultaneously affected by faults.

Keywords: Photovoltaics · Predictive maintenance · Fault detection · Recurrent neural networks · Graph neural networks · Time series classification.

1 Introduction

In facing today’s climate change challenges, the ambition has been expressed to limit the rise of global temperature to +1.5 °C, which amounts to achieve net

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

** This chapter is based on published papers [1,2] by the same authors, from which we reproduce selected results.

zero emissions by 2050 [3]. Since solar photovoltaic (PV) power generation forms a crucial element of realizing these ambitions, we observe a rapid increase of the number of installed PV capacity in recent years. While PV cell efficiency has been improving steadily, they still can fail, and PV system faults may cause non-negligible energy losses, especially when such faults remain undetected [4]. To limit the impact of such losses and thus maximize the efficiency and lifetime of PV installations, predictive maintenance solutions to detect and identify PV system faults from their earliest manifestation are essential. Indeed, since climate change is expected to incur more extreme weather events, PV systems will likely be subject to higher levels of thermal and mechanical stress, which in turn impacts some of the possible PV faults. Thus, we expect the importance of widespread PV fault diagnosis only to increase.

The types of faults that may occur in PV systems include short circuits, wiring degradation, hot spots, etc. The majority of such PV faults can be identified through visual inspection either by human technicians, or even infrared thermography with drones. Yet, such inspection is costly and therefore typically not adopted for smaller scale systems: the majority of PV systems thus remains unmonitored [5]. To provide cost-effective fault diagnosis, we explore methods based on artificial intelligence: these methods can achieve reasonably high fault detection accuracy, while avoiding the deployment of costly sensor infrastructure and without requiring to define system-specific detection thresholds [6].

In Section 2, we will first give a brief overview state-of-the-art models for PV fault diagnosis based on machine learning [7], and classify them in 3 categories. Subsequently, we will present and analyze two recent neural network models we proposed for this task: (i) a recurrent neural network (RNN) model (Section 4) taking a single site's local PV installation's current and voltage measurements and satellite based weather information as input, and (ii) a graph neural network (GNN) model (Section 5) to detect faults from multiple sites, without using any weather information at all. Both models are developed to detect and identify 6 different fault types. The RNN model also estimates their impact severity in terms of relative reduction of the PV installation's output power as a result of the defect. Note that in this chapter we will outline the models' architectures and key results from our experiments. For in-depth discussions, we refer to [1] for the RNN model and [2] for the GNN. We summarize our key take-away messages in the concluding Section 6.

2 Related work

We will discuss PV fault diagnosis methods according to the type of information they use as *inputs* to determine whether the PV system is not performing as expected (and possibly determine the type of underlying fault):

- (1) Comparison of the *expected and actual* power output (or the actual current I and voltage V profiles that determine it), which essentially relies on *environmental information* (i.e., irradiance) that is typically gathered from sensors;

- (2) Pure *current (I)* and *voltage (V)* based classification, i.e., without relying on environmental sensor information, which thus needs more detailed measurements in terms of either (i) *I-V curves*, or (ii) high frequency *transient* behavior in I_{MPP} and V_{MPP} , i.e., the current and voltage at maximum power point (MPP);
- (3) Comparison with *reference PV output*, which may be either (i) on the module level, i.e., comparing to a reference module at the same PV site, or (ii) from a geographically nearby PV site.

For each of these 3 categories, the following subsections discuss associated related works, which are summarized in Tables 1 to 3 (wherein we also position our proposed methods, as discussed and analyzed in detail in Sections 4 and 5).

2.1 Comparison to expected PV output

To assess whether a PV system suffers from any defect, a common method is to compare its measured output to a prediction of the expected output, given the current environmental conditions [8–14]. Such prediction is realized either using physics-based PV simulations [8, 10, 11] or rather through data-driven regression [12, 13]. The environmental conditions in both cases include solar irradiance and ambient and/or PV module temperature.

Both [12] and [13] adopt a multilayer perceptron (MLP) for prediction. De Benedetti et al. [12] define a threshold on the difference between measured and predicted *power* output, above which a failure is assumed. Also Jiang et al. [13] define a threshold, this time on 1 kHz *voltage and current* measurements vs. predictions: such high-frequency measurements allow further identification of the failure type, yet are costly to collect and process. Chine et al. [8] rather assumes *I-V curve characteristics*, both from the actual PV system and a simulated counterpart, which thus requires an I-V curve tracer (which again is not cost-effective in typical residential set-ups).

Assessing deviations between predicted and measured outputs, and thus determining faults based on that deviation exceeding a predefined threshold, seems conceptually simple. Yet, the definition of appropriate thresholds requires expert knowledge and becomes increasingly complex when multiple different fault types need to be identified [6]. Therefore, more advanced machine learning models have been adopted: Garoudja et al. [10] use probabilistic neural networks, while Adhya et al. [11] adopt gradient boosted trees. The downside of these largely black-box approaches is that they are not as easily interpretable as threshold-based rules.

While the methods discussed above rely on environmental information, essentially from local weather sensors, Zhao et al. [14] avoid needing detailed sensory information but rather rely on designated reference PV modules at the site itself. Clearly, the method will suffer if those reference modules themselves are experiencing faults.

Our RNN-based approach that we will detail in Section 4 avoids any local measurements, but rather relies on environmental data (irradiance and ambient

Ref.	Machine learning model	Inputs	Imple-mentation cost	# identified fault types
[8]	Multilayer perceptron	<ul style="list-style-type: none"> • I-V curve • Irradiance • Module $^{\circ}\text{T}$ 	High	6
[9]	Ensemble model	<ul style="list-style-type: none"> • I_{MPP} & V_{MPP} • Irradiance • Module $^{\circ}\text{T}$ 	Medium	2
[10]	Probabilistic neural network	<ul style="list-style-type: none"> • I_{MPP} & V_{MPP} • Irradiance • Module $^{\circ}\text{T}$ 	Medium	2
[11]	Gradient boosted trees	<ul style="list-style-type: none"> • I_{MPP} & V_{MPP} • Irradiance • Ambient $^{\circ}\text{T}$ 	Medium	4
[12]	Multilayer perceptron	<ul style="list-style-type: none"> • P_{MPP} • Irradiance • Ambient $^{\circ}\text{T}$ 	Medium	Detection only
[13]	Multilayer perceptron	<ul style="list-style-type: none"> • High-freq. I_{MPP} & V_{MPP} • Irradiance • Ambient $^{\circ}\text{T}$ 	High	5
Ours (RNN) [1]	Recurrent neural network	<ul style="list-style-type: none"> • I_{MPP} & V_{MPP} • Satellite irradiance • Satellite ambient $^{\circ}\text{T}$ 	Low	6

Table 1. Methods based on comparison to expected PV output. **Bold:** our proposed RNN model. (MPP: maximum power point; I_{MPP} : MPP current; V_{MPP} : MPP voltage; P_{MPP} : power at the MPP; $^{\circ}\text{T}$: temperature.)

temperature) readily available from satellites. Thus being pragmatic and cost-effective, our method will prove to still exhibit high classification performance despite the less accurate information used.

2.2 I,V-based classification

The methods that avoid using any environmental sensor information at all need more detailed local measurements to detect possible failures. In particular, in our overview listed in Table 2, we find methods that rely either on (i) *I-V curves*, which implies that I-V tracers are installed at the PV site, or (ii) high-frequency measurements of I_{MPP} and V_{MPP} , to identify *transient effects* stemming from defects.

Chen et al. [15] manually define parameters from the *I-V curves*, based on analysis of such curves for various faults and weather conditions, and feed them to a kernel learning machine to perform 4-way classification. Similarly, Spataru et al. [16] rely on I-V curve parameters fed into fuzzy classifiers. To avoid manually engineering the input features to determine failures from, others have proposed purely data-driven deep learning approaches, using, e.g., convolutional neural networks [17] or residual neural networks (ResNet) [18].

If a fault occurs, this typically will also lead to *transient behavior* that can be observed in high frequency measurements of current and voltage, even though such transients typically last no longer than a few seconds [19]. Since the footprint of faults in such transients differ, various methods have been exploited for failure detection using the high resolution I_{MPP} and V_{MPP} measurements: random forests [19, 20], wavelet packet transforms [21], and convolutional neural networks [22]. Practical drawbacks are that (i) only the short period of transients is useful to detect the fault (i.e., if measurements are missing from such period, the fault will remain undetected), and (ii) high frequency measurements require costly sensors (e.g., in the order of € 5,000 for the 1 kHz measurements in [21]).

2.3 Comparison to reference PV system

To avoid additional sensors (for either weather or current/voltage tracing), several researchers have proposed to compare PV output to a given reference, as summarized in Table 3. This can be done either (i) on the *module level*, thus comparing the individual modules to one another, or (ii) on the full *system level*, using nearby PV systems to compare against. Clearly, methods in category (i) require the PV system to be equipped with micro-inverters (per module), whereas in practice string inverters are more common. Examples of techniques adopted in such *module-based* solutions include convolutional neural networks [23] and random forests [24]. The latter technique has also been used in a *system-level* comparison approach [25]. Our second proposed method, detailed in Section 5, will use graph neural networks to exploit the same idea of comparing system outputs (but realizing detection of also the fault type, as well as generalization to unseen PV systems — whereas [25] only considers binary classification, i.e., fault vs. no fault, on a fixed set of PV systems).

3 Problem definition

The desired **output** of our predictive model is twofold: (i) the *type* of fault the PV system suffers from (if any), and (ii) the relative *power reduction*, compared to normal operation, that results from the fault. The first essentially is a categorical variable, where we want to distinguish different fault types, as well as the *no fault* class. In our case studies, we will discriminate 6 different fault types, described briefly in Table 4.

As **input** variables, we will consider time series data spanning hourly measurements of the PV system's current (I) and voltage (V), as measured by the

Ref.	Machine learning model	Inputs	Imple-mentation cost	# identified fault types
[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 °T	High	4
[18]	ResNet	• I-V curve • Irradiance • Ambient °T	High	5
[21]	Wavelet packet transforms	• High-freq. I_{MPP} & V_{MPP}	High	Detection only
[19]	Multi-grained cascade forest	• High-freq. I_{MPP} & V_{MPP}	High	3
[22]	Convolutional neural network	• High-freq. I_{MPP} & V_{MPP} • I & V of reference module	High	2
[20]	Random forest	• High-freq. I per substring • High-freq. V_{MPP}	High	4

Table 2. I,V-based classification methods. *Top:* based on I-V curve, *Bottom:* based on transients in I_{MPP} and V_{MPP} . (MPP : maximum power point; I_{MPP} : MPP current; V_{MPP} : MPP voltage; P_{MPP} : power at the MPP; $^{\circ}T$: temperature.)

inverter. Furthermore, our single-site model (based on RNNs, see Section 4) will take environmental data (zenith angle, ambient temperature, irradiance), whereas our multi-site model (based on GNNs, see Section 5) will assume to know the differences between pairs of sites in terms of distance, altitude, azimuth and tilt.

Since both models are essentially supervised machine learning approaches, we need **training data**. Given that it is difficult and possibly dangerous to induce various faults in real-world systems, we rely on simulated data to train our models. We thus use physics-based PV simulations, based on the well established single-diode model. The model we adopt has been experimentally validated on real-world residential PV systems [27,28], and has been shown to deliver output power values that are significantly more accurate than what the commercial tool

Ref.	Machine learning model	Inputs	Imple-mentation cost	# identified fault types
[14]	Local and global consistency algorithm	<ul style="list-style-type: none"> I_{MPP} & V_{MPP} I & V of ref. modules 	Medium	2
[23]	Convolutional neural network	P_{MPP} per module	Medium	Detection only
[24]	Random forest	P_{MPP} per module	Medium	3
[25]	Random forest	P_{MPP}	Low	Detection only
Ours (GNN) [2]	Graph neural network	I_{MPP} & V_{MPP}	Low	6

Table 3. Methods based on comparison to a reference PV system. *Top:* comparison to reference PV modules at the same site, *Bottom:* comparison to nearby PV systems. **Bold:** our proposed GNN model. (MPP: maximum power point; I_{MPP} : MPP current; V_{MPP} : MPP voltage; P_{MPP} : power at the MPP.)

PVsyst predicts.¹ The rightmost column of Table 4 indicates how the various fault types were modeled in the simulation tool.

The PV system layout we used is sketched in Fig. 1, which also indicates how/where we introduced the various faults. This layout was aligned with the PV systems assumed in literature [15, 29] to facilitate a fair comparison of results. Weather data was taken from the real-world dataset that is publicly available from the National Renewable Energy Laboratory (NREL) [30], including irradiance, ambient temperature, relative humidity, wind speed and wind direction. Since we assume that current and voltage measurements are taken from the inverter, and in practice these measurements are imperfect, we introduce 5% random noise on the simulated current and voltage. For further details, we refer to [1, 2].

4 Recurrent Neural Network model for a single PV system using satellite weather information

4.1 Model: A stacked GRU network

As outlined in Section 3, the aim is to infer the type and severity of possible PV system faults from a multivariate time series covering the past 24 h of

¹ In [27], PVsyst version 6.23 was used to establish this result: the authors' tool achieved a mean absolute error (MAE) of predicted energy yield of $3.6\% \pm 2.8\%$, while PVsyst achieved an MAE of $5.5\% \pm 4.1\%$.

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.	Add a resistor on the connection to the inverter of 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)	Potential-induced performance degradation: 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 [26].	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 our work.

environmental parameters as well as current and voltage, measured at hourly resolution. Recurrent neural networks (RNN) are well suited for such time series, and recent well-performing models include long short-term memory (LSTM) and gated recurrent unit (GRU) networks: such RNN models have been successfully applied to a broad range of tasks, ranging from speech recognition, over machine translation, to time series forecasting [31]. We adopt a GRU-based architecture rather than using an LSTM, since it achieves similar performance but is characterized by fewer parameters. To minimize computational requirements, and aiming for maximal generalization, we share the same GRU layers to generate common representations for both the classification (fault identification) and regression (severity estimation) tasks. Note that we apply layer normalization in between the GRU layers (to speed up training and act as regularization), and add a residual connection (to avoid vanishing gradients). We feed the last GRU cells' outputs to fully connected layers with a ReLU activation, and subsequently add a softmax and sigmoid head to finally obtain respectively the fault type classification and severity regression outputs, as sketched in Fig. 2. To feed in the 5-dimensional time series into the GRU layers, we apply standardization (i.e.,

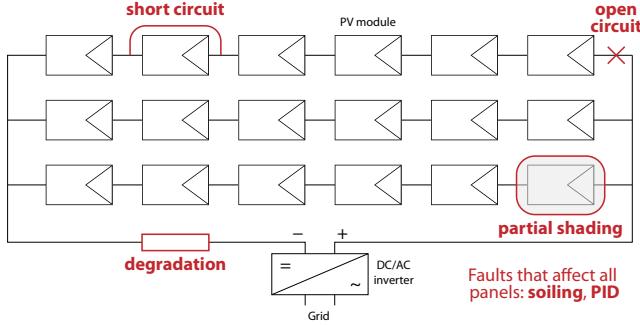


Fig. 1. Schematic layout of the 18-module PV system that was simulated with an indication of the simulated **fault types**. (Adapted from [1].)

rescaling a feature x_i as $x'_i = \frac{x_i - \mu_i}{\sigma_i}$, where μ_i and σ_i are the feature's mean and standard deviation respectively).

To train the model, we define a loss function that combines a cross-entropy loss term for the classification part, and a mean square error loss term for the severity regression part. Additionally, to enforce consistency between both outputs, we add a penalization loss to avoid predicting “*no fault*” with a severity greater than zero. Since our training set is imbalanced, i.e., not all fault types are equally represented (e.g., we consider various levels of shading, which is not applicable for the “*no fault*” class), samples of a more prevalent class are weighed less in the loss function. This results in a loss function as defined in Eq.(1), where for each data sample j we define w_j as its weight, \hat{y}_j^c as its softmax probability for class c (with 0 representing the “*no fault*” class), s_j as its true and \hat{s}_j as its predicted severity level (i.e., the average relative power reduction compared to fault-free operation, thus 0 for “*no fault*”), and c_j as its true fault type.

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\text{CE}} + \alpha \mathcal{L}_{\text{MSE}} + \beta \mathcal{L}_{\text{NFS}} \\ &= \frac{1}{N} \sum_{j=1}^N w_j \left(-\log \hat{y}_j^{c_j} + \alpha (s_j - \hat{s}_j)^2 + \beta \hat{s}_j \hat{y}_j^0 \right) \end{aligned} \quad (1)$$

For evaluation, we will use 5-fold cross-validation, where in each fold we keep a separate year as test set to assess performance of a model. In the training loop, we keep 100 randomly selected days as a validation set and shuffle the remaining data to construct the training batches. That validation set is used for hyperparameter tuning and early stopping. The latter implies that we stop training if the model performance on the validation set does not improve compared to the previous training epoch, since this can help prevent overfitting [31]. For the full training procedure details, we refer to [1].

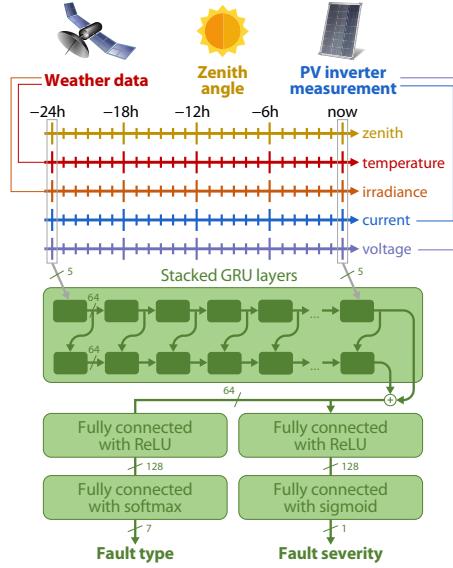


Fig. 2. We propose a stacked GRU architecture to process the multivariate time series, and add (i) a softmax layer to classify the fault into 7 types (including *no fault*), and (ii) a sigmoid to predict the power reduction impact (a factor in $[0, 1]$) on the PV system’s output. We use ReLU activation in fully connected layers. Connections are annotated with the dimensions of the vector passed between model components.

4.2 Experiment setup

We implement the aforementioned GRU-based model, which we train for the fault types described in Section 3. As a baseline, we use CatBoost as a reference, which is based on gradient boosted decision trees, since it has been shown to outperform other boosting implementations [32]. Note that CatBoost cannot perform both the (fault type) classification and (severity) regression tasks simultaneously, and we thus construct two independent models for each of them — implying that we cannot include the 3rd loss function to enforce consistency between them, as we did in Eq. (1) for our RNN-model.

We further setup experiments to answer three research questions: (Q1) What is the maximal performance our model can achieve, assuming it has *perfect weather data*? (Q2) How much does that performance suffer when we can only rely on approximate environmental information by relying on *satellite weather data*? (Q3) Can our model detect *unknown faults* by looking solely at its fault severity prediction (exceeding a certain threshold)?

For (Q1), we simply feed the exact environmental parameter values as we also use to generate the simulated PV system output data (still adding noise to the inverter’s V and I values, as explained previously).

With (Q2), we aim to assess our model performance in real-world, practical settings, where residential PV systems are not equipped with local weather

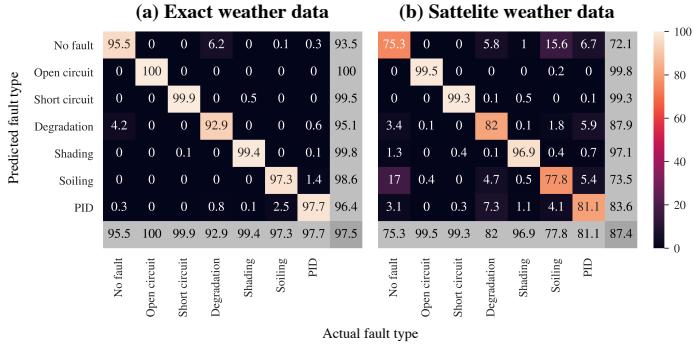


Fig. 3. Confusion matrix for our RNN model using **(a)** *exact* local weather, or **(b)** *satellite* weather data inputs.

sensors. Thus, we rely on weather satellite data, which we obtain from MERRA-2 [33]. This only offers a single time series for an entire US state (North Carolina in our experiments), and thus deviations between the actual local conditions and satellite data values, in terms of mean absolute error (MAE), amount to 49.9 W/m for irradiance and 2.9 °C for ambient temperature. We use this satellite data as environmental input to our RNN prediction model, along with the I and V values from PV system simulations (which obviously are still fed with the actual local weather data, rather than satellite weather data).

Finally, to answer (Q3), we leave out one fault type from the training set, and then use that fault type's test data to assess whether our model would still detect a failure by comparing the predicted severity against a (learned) threshold. Specifically, we learn the threshold as the maximal severity level predicted by the model for correctly classified “*no fault*” data points in a validation set.

4.3 Results

Exact weather data — With the first experiment, we answer (Q1). In terms of *fault identification*, Fig. 3(a) shows the confusion matrix between our RNN model's prediction and the ground truth. We obtain a high balanced classification accuracy (i.e., weighing each fault type equally) of $96.9\% \pm 1.3\%$ (averaged over the 5-fold cross validation test sets ± 3 times the standard deviation). We note that the most mistakes are made by confusing “*no fault*” and wiring degradation.

Looking at the *severity* predictions in Fig. 4, we note that the misclassified samples (orange dots) tend to also suffer from poor severity estimation. First of all, this is unsurprising because both models rely on the shared GRU layers to derive the representations fed into the classification and regression heads. Second, we also note the effect of the loss term to make both outputs consistent (i.e., \mathcal{L}_{NFS} in Eq. (1)). For example, in the actual “*no fault*” samples, we only observe (erroneously) high severity levels for the misclassified ones. Over all

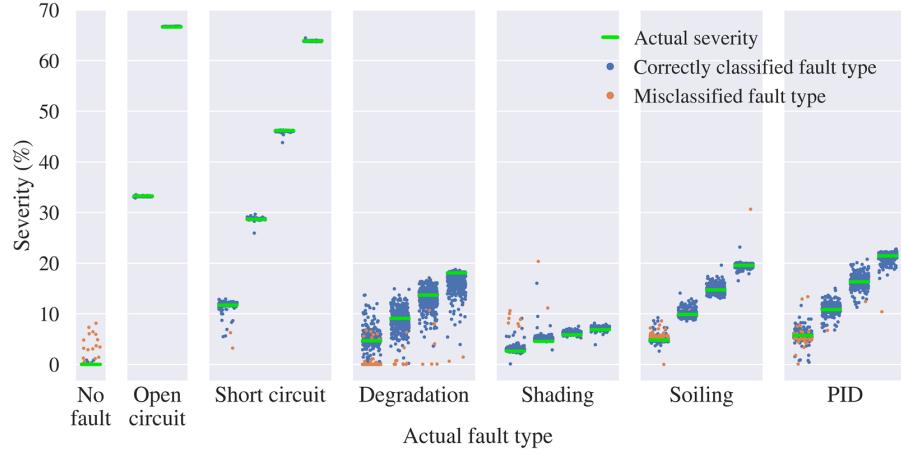


Fig. 4. Actual and predicted severity level from our RNN model, using actual *exact* local sensor weather data as inputs. Each dot represents a test sample, where we only pick one per 24 h window to avoid cluttering, and we group them by the gold truth label on the X-axis.

Weather Data	Type prediction accuracy		Severity prediction MAE	
	CatBoost	RNN (Ours)	CatBoost	RNN (Ours)
Exact	93.2% \pm 1.3%	96.9% \pm 1.3%	1.66% \pm 0.12%	0.67% \pm 0.14%
Satellite	83.5% \pm 2.1%	86.4% \pm 2.1%	3.6% \pm 0.24%	2.09% \pm 0.18%

Table 5. Performance metrics averaged over the 5-fold cross-validation results, where error margins are 3x the standard deviation. Prediction accuracy is balanced over the 7 possible classes (6 fault types and “no fault”). (*Values reproduced from [1]*).

samples together, we find a balanced MAE of our model’s severity estimation of $0.67\% \pm 0.14\%$, which is quite limited.

Finally, comparing our RNN model to the CatBoost baseline in Table 5, we note a significant benefit of our model (+3% for fault type classification accuracy and -1% MAE for severity estimation).

Satellite weather data — The second set of experimental results looks into (Q2). Given the discrepancy between actual local conditions and the low-resolution satellite information (cf. the stated irradiance MAE of 49.9 W/m^2 , with individual data points deviating up to 850 W/m^2), the fault classification and severity estimation gets more challenging: a difference between expected power output and the actually observed values could be due to a PV system fault or just the environmental parameter error. Inspecting the satellite vs. local data deviations more closely (for the studied location), we observed a general over-estimation of

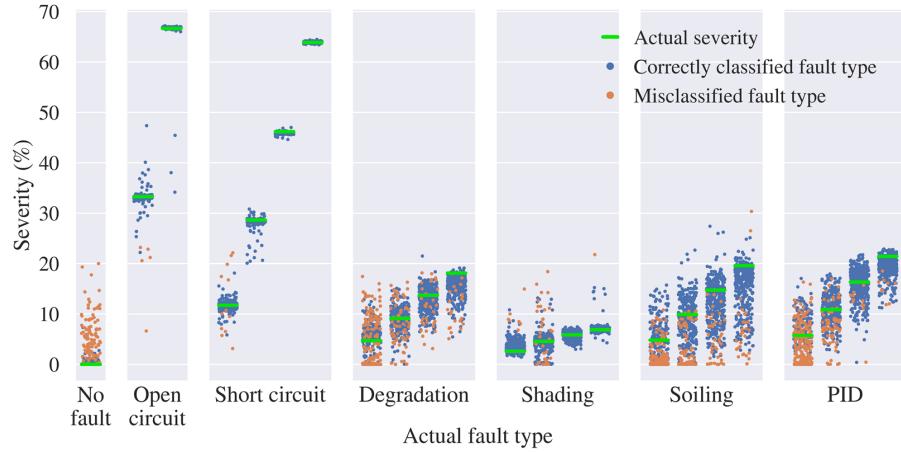


Fig. 5. Actual and predicted severity level from our RNN model using *satellite* weather data as inputs. Each dot represents a test sample, where we only pick one per 24 h window to avoid cluttering, and we group them by the gold truth label on the X-axis.

irradiance and under-estimation of ambient temperature by relying on satellite data.

Despite these inaccurate input feature values, our RNN-based model still achieves $86.4\% \pm 2.1\%$ balanced *classification* accuracy. Intuitively, this can be explained by the fact that our model can learn to take into account the noise on the inputs, since it is also present in the training data. Looking at the confusion matrix in Fig. 3(b), we note that the model mostly has difficulty distinguishing between soiling and “no fault”, given that an overestimated irradiance also results in seemingly under-performing PV output power (that also arises from dust accumulation on the panels in the “soiling” case).

In terms of *severity* estimation as illustrated in Fig. 5, we note an overall MAE in severity estimation of $2.09\% \pm 0.18\%$, largely due to an underestimation of soiling severity (whereof the effect is hard to distinguish from the satellite data overestimating irradiation). To a lesser degree, we note similarly natured severity estimation errors for wiring degradation and PID.

Comparing our RNN-model’s performance to the CatBoost baseline solution in Table 5 still reveals notable advantages of our model (+3% balanced classification accuracy, -1.5% severity prediction MAE).

Testing on unknown faults — Our final set of results presented here answer (Q3). Figure 6 shows the severity predictions by our model when trained on all data except that with PID faults. We determine a binary fault/no fault classification threshold on the severity predictions as the maximal severity that is predicted for correctly classified “no fault” samples from the validation set. We note that 97.5% of the PID samples — which are obviously misclassified, since the fault type was not considered in the training set — are thus detectable as a “fault”.

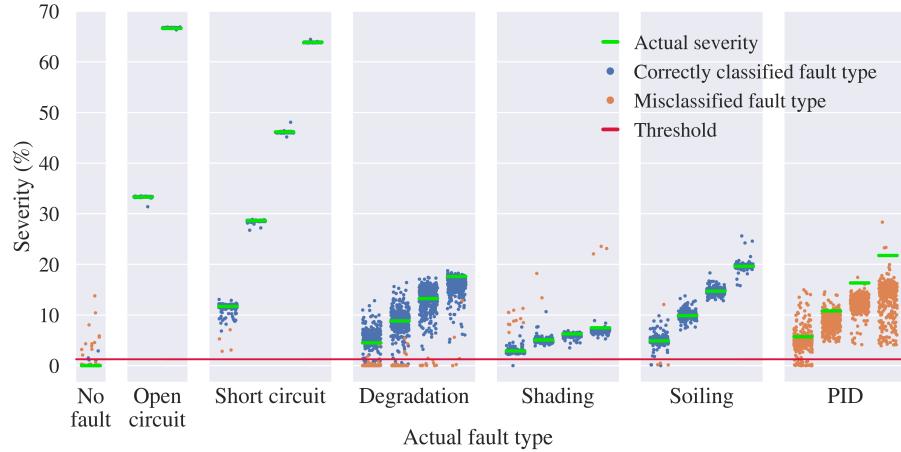


Fig. 6. Actual and predicted severity level from our RNN model trained on all but the PID fault data. Each dot represents a test sample, where we only pick one per 24 h window to avoid cluttering, and we group them by the gold truth label on the X-axis.

Weather data	Detection accuracy of excluded fault type as “unknown fault”						Average
	Open circuit	Short circuit	Degradation	Shading	Soiling	PID	
Exact	100%	99.6%	70.7%	98.4%	99.7%	97.5%	94.3%
Satellite	100%	98.4%	83.4%	93.0%	22.3%	85.2%	80.4%

Table 6. Binary classification performance of models trained on all but 1 fault type. Final column is the overall macro-average (i.e., average of the preceding 6 columns). (*Values reproduced from [1]*).

Repeating this setup for all of the faults (i.e., training an instance of our RNN-based model without supplying such fault data), we obtain the binary fault detection accuracies listed in Table 6.

Averaging the binary fault detection accuracies among all 6 fault types, we find an “unknown fault” detection accuracy of 94.3% when using *exact weather data*. We note that the performance is substantially lower for the model that has not seen degradation. We hypothesize that this is due to the difficulty to differentiate between degradation and “*no fault*” as previously observed in the confusion matrix of the model trained on all faults (see Fig. 3(a)).

When we only have access to (inaccurate) weather information based on *satellite data*, we see from Table 6 that the overall unknown fault detection capability is reduced to 80.4% on average. In Table 6, we particularly note the poor performance when trying to detect soiling with a model that did not see any such faults during training, while compared to model based on exact weather

data the detection of degradation improves. Again, looking at the all-fault model's confusion table (see Fig. 3(b)) gives some idea why: we note a high occurrence of mistakenly classifying soiling faults as "*no fault*" and vice versa — whereas among degradation and "*no fault*" there is less confusion than when using exact weather data.

5 Graph Neural Network model for multiple PV system sites

5.1 Our GNN model

The primary goal of our graph neural network (GNN) based model, detailed below, is to enable failure detection that does not require any weather information (i.e., no such local sensors, nor relying on coarser estimations based on satellite data). The basic intuition is to use and compare the inverter measurements from nearby PV sites, assuming that some of them are operating without faults, and thus can serve to (implicitly) construct a reference to compare against. We now consider only fault type *classification*, hence we do not include a severity estimation output — although that can be straightforwardly added with a separate sigmoid layer, as in the previously discussed RNN model.

The principle of a GNN is that nodes and edges, which we respectively note as v_i and $e_{i,j}$,² are represented by feature vectors as input to a GNN layer that transforms them by (i) first calculating output edge representations $e'_{i,j}$ derived from the original edge representations and its pair of adjacent nodes, and then (ii) calculating output node representations v'_i based on the original node representations, and an aggregated representation of its incident edges. This implies the use of parameterized functions f_e and f_v to consecutively calculate the output edge and node representations as follows:³

$$e'_{i,j} = f_e(e_{i,j}, v_i, v_j) \quad (2)$$

$$v'_i = f_v(v_i, E'_i), \quad \text{with } E'_i = \sum_j e'_{i,j} \quad (3)$$

The model parameters to learn thus are those that define f_e and f_v , which are multilayer perceptrons.

Model architecture — The overall architecture of our GNN model is sketched in Fig. 7. We construct a graph, where each node corresponds to a PV site. For the *node* representations, we thus only use local measurements from each site individually, being 24 h of voltage and current measurements from the inverter. As in the the RNN model, we process this time series with a stacked GRU

² For simplicity, in our notation we assume undirected edges and use $e_{i,j}$ to represent an edge between nodes v_i and v_j .

³ Note that instead of simply summing the incident edge representations, in general GNNs other aggregation functions can be used to obtain E_i , e.g., mean-, min- or max-pooling.

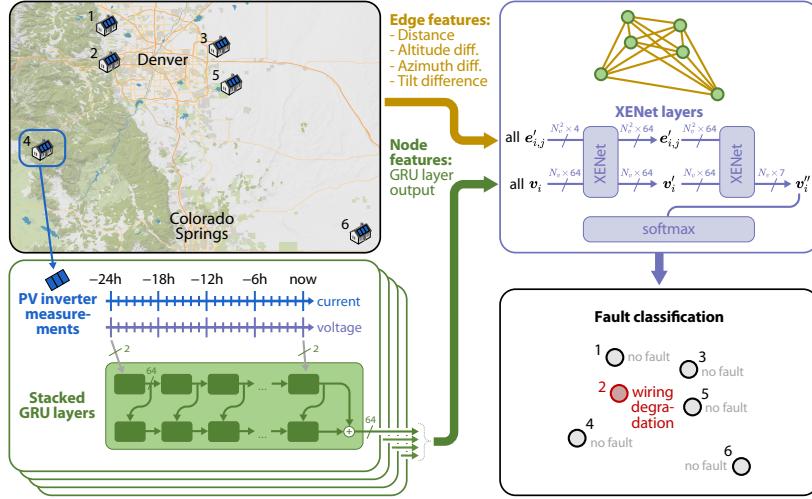


Fig. 7. The proposed graph neural network (GNN) based model. Each PV site is represented as a node in the graph, where (i) *node features* are the outputs of a stacked GRU (as in our RNN-based model, see Fig. 2) that processes a site’s hourly current and voltage measurements over a 24 h time window, and (ii) *edge features* are the distance between PV sites and their difference in altitude, azimuth and tilt. Note that we do not use/calculate edge representations coming out of the 2nd XENet layer.

architecture (as in Fig. 2). We create a fully connected graph between these nodes, where each edge between a pair of PV sites has as features the distance between those sites, as well as their difference in altitude, azimuth and tilt (which all remain constant over time). The graph features then go through GNN layers, more specifically 2 XENet layers [34]. We opted for XENet rather than graph convolutional networks (GCN) [35], since it explicitly supports edge features, which GCN does not. Moreover, we found that XENet outperformed other GNN types that support edge features, particularly edge-conditioned convolutional networks [36] and crystal graph convolutional networks [37]. Finally, we note that we normalized the input features as detailed in [2, §4].

Training — As before, we rely on physics-based simulations to produce the training data [27, 28]. We consider the same faults listed previously in Table 4, but will use different PV system layouts and module types for each of the 6 considered sites, as listed in Table 7. For the weather data, we use measurements publicly available from the National Renewable Energy Laboratory (NREL) [30], taking data from weather stations at the locations shown on the map in Fig. 7, spanning a time range of 2012–18 for sites 1–4 and 2012 for sites 5 and 6.

Since we are dealing with a pure classification problem (cf. we do not include severity estimation in this section), we use a cross-entropy minimization objective (i.e., the \mathcal{L}_{CE} part of Eq. (1)). As before, we weigh the instances to compensate for

Site	PV module type	Number of modules	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°
Module parameter	SW 325 XL duo	Scheuten P6-60 i30		
Number of cells (series \times 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}	$-0.43\%/\text{K}$	$-0.42\%/\text{K}$		
Temperature coefficient of V_{OC}	$-0.31\%/\text{K}$	$-0.30\%/\text{K}$		
Temperature coefficient of I_{SC}	$0.044\%/\text{K}$	$0.040\%/\text{K}$		

Table 7. PV system and module configurations for each of the 6 sites.

the class imbalance in the data, such that each fault type contributes equally to the overall loss. Note that, in contrast to the RNN-based model of Section 4, in the GNN case the “*no fault*” instances are far more numerous, since we consider at most one of the PV sites to suffer from a fault. Note that the results do not change significantly when multiple PV systems are simultaneously faulty [1, §7.4].

5.2 Experiment setup

We set up experiments to answer three research questions: (Q4) How does the GNN model compare to our previous RNN-based model and the CatBoost baseline? (Q5) How good do our models perform without any weather information? (Q6) Can a trained model generalize to PV system sites that were not included in the training data? (The latter thus amounts to zero-shot classification for new PV sites.)

Baseline models — As (Q4) states, we will compare the presented GNN-based model against our previously introduced RNN-based model, as well as the CatBoost baseline described in Section 4.2. Since we consider only fault classification (i.e., we do not perform severity estimation as with the original RNN model), we slightly alter the RNN model to only have the classification output and thus remove the ‘fault severity’ branch from the architecture sketched in Fig. 2. Similarly, we train a single CatBoost model, only for fault classification. All of our models are trained for a balanced objective where each class is weighed equally, by multiplying the various samples with an appropriate factor depending on their actual fault type, as explained earlier (cf. the weight factors w_j in Eq. (1)).

Note that in answering (Q4), we initially will consider providing also satellite weather information (irradiance, temperature, solar zenith). This means we also add weather time series as inputs of the GRU layers (now taking 5-dimensional vectors as input) for both our GNN model and the baseline RNN model. Next, for (Q5) we will omit weather information and thus only consider voltage and current time series as input.

Since all of our models in this section now only consider fault classification, we use cross-entropy loss minimization as objective for all of them, where we weigh each sample to achieve balancing of the loss across all 7 classes. For training details, we refer to [2].

5.3 Results

Performance with satellite weather data — We first test our GNN model and compare it with our earlier RNN as well as CatBoost, for fault classification based on current, voltage, irradiance, ambient temperature and solar zenith — essentially the same inputs as in Section 4. We used the freely available satellite data from MERRA-2 [33] for the weather inputs (which are for the entire state of Denver, and thus only coarse approximations of the actual weather conditions at each of the 6 selected locations of the PV sites; e.g., amounting to a MEA of 51.11 W/m^2 for irradiation at site 1). We train and evaluate all models on data for sites 1–4, using 5-fold cross-validation as explained before.

Looking at Fig. 8(a) to answer (Q4), we find that our proposed GNN model significantly outperforms both CatBoost and RNN models,⁴ especially in terms of discriminating between “*no fault*” and soiling. As we noted in Section 4, this is difficult for both the RNN model and CatBoost, because the satellite weather information does not accurately match the local PV site conditions: the irradiance overestimations of the satellite data lead to seemingly under-performing PV output, which is hard to distinguish from impaired performance due to soiling. The GNN model does not suffer as much from imprecise satellite measurements, since it can differentiate inaccurate weather inputs (which affect all sites) from fault conditions (at only 1 or a limited number of PV sites). The full cross-validation results over all 5 folds presented in Table 8(a) confirm our GNN model’s superiority.

Performance without any weather data — Our motivation for using a GNN model and combining input data from multiple PV sites was that this should enable fault detection without any weather information. Yet, since the various PV sites considered may have different orientations (as in our experiments, cf. Table 7), the PV output at a particular time of the day cannot be readily compared: a south-facing PV system will have its maximal power (in clear sky conditions)

⁴ Note that the results presented here for CatBoost and RNN differ from those in Section 4 because there we only had 1 PV site, where here we have a heterogeneous set of PV systems across the considered sites 1–4, and also the current models are not additionally trained on severity regression.

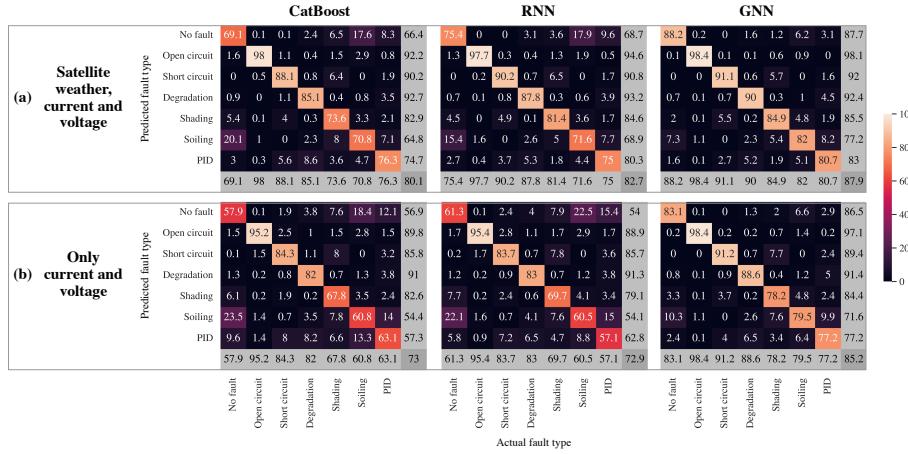


Fig. 8. Confusion matrices for the CatBoost, RNN and GNN models on the 1st fold (of the 5-fold cross-validation test sets), using (a) weather info as well as voltage and current, or (b) only current and voltage measurements from the PV inverter. Gray shaded columns and rows state the precision and recall values, respectively.

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

Table 8. 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.

earlier than a west-facing system. However, since we provide the past 24 h time window of data, our GNN model should learn to adjust for such differences.

To answer (Q5) we thus omit the weather data time series from the model inputs, only keeping the inverter voltage (V) and current (I) measurements (as originally sketched in Fig. 7). The results in Fig. 8(b) show the confusion matrices for the CatBoost, RNN and GNN models using only V and I time series as inputs. As expected, compared to the model variant *including* weather data, the GNN model does not suffer much performance loss. Conversely, the CatBoost and RNN models do deteriorate considerably, even though Table 8(b) shows they still attain an overall accuracy well over 70%. The latter may come as a surprise, but the still ‘decent’ performance may be due to the models being able to infer the ‘expected’ current and voltage levels from the training data without faults.

Performance on unseen PV sites — As explained in Section 5.1, a GNN model essentially contains parameterized functions to map input node and edge representations to output node and edge representations. Given that the output

Inputs	Site	Accuracy		
		CatBoost	RNN	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 9. Accuracy on *unseen PV sites* 5 and 6 (as indicated on the map in Fig. 7), of which no data was included in the training or validation set of the models. Note that the GNN model predictions for all 6 sites, but only the accuracy on the unseen sites is reported here.

node features are based on *aggregated* edge features (cf. the summation in Eq.(3)), implying that the mapping functions do not depend on the actual number of nodes (nor edges) in the graph, we can apply the learned mappings also for new nodes that were not included in training the GNN model. Thus, we can train a GNN model on a limited number of sites (in casu sites 1–4 of our case study), and afterwards apply it for fault detection of all sites, including the unseen sites 5 and 6. We set up this experiment to answer (Q6), and compare the GNN model against the CatBoost and RNN baselines, which only use single-site time series data as inputs and thus can also be tested on unseen sites — we thus also train these baselines on sites 1–4 and will test on sites 5–6. Since we only have 1 year of weather data for sites 5–6, we use this full year of 2012 as test data, while training the GNN model on only sites 1–4 for a five-year period, excluding 2012.

Table 9 shows that our GNN model successfully generalizes to site 5, which is situated relatively close to some of the sites the model was trained on (and thus inter-site distances, which serve as edge features in the GNN, are comparable to that in the training data). Yet, this is still a non-trivial achievement since the tilt, azimuth, and module configuration of site 5 is still quite different from the training sites 1–4 (recall Table 7). We note that the RNN model and especially CatBoost do show a considerably steeper performance drop on this unseen site 5, especially when looking at the model variants that can only use current and voltage measurements — which is what one would intuitively expect. Still, looking at the results for site 6, we observe that also our GNN model is far less successful (even though still beating the RNN and CatBoost models). This performance drop for site 6 is likely due to the site being quite distant from any of the training sites 1–4 (of which site 6’s closest neighbor is 196 km away). Hence, our answer to (Q6) seems to be that, yes, our GNN model can generalize to unseen sites, but only as long as the new PV system is not too remote from any of the sites used for training the model.

6 Conclusions

In this chapter, we discussed data-driven solutions for cost-effective fault detection for PV panels. By *cost-effective* we mean that they do not rely on additional sensor equipment beyond (low temporal resolution) voltage and current measurements from the inverter, which converts the PV modules' DC to AC power. More specifically, we investigate the adoption of recent state-of-the-art neural network models: (i) a *recurrent neural network (RNN)* based architecture, using gated recurrent units (GRUs) to process hourly measurements from a *single PV site*, and subsequently (ii) a *graph neural network (GNN)* based model, taking the per-site representations from such GRU layers, and jointly processing them across *multiple sites*.

We have shown that the *RNN-based model* is effective even when only using (inaccurate) satellite based weather data — although overall fault type accuracy does drop from around 96% when using exact local weather information to 86% when using satellite based data — and thus constitutes a workable solution for independent single site fault monitoring. Compared to state-of-the-art in PV fault detection literature, our RNN model (i) requires neither high temporal resolution data nor I-V curves nor local weather measurements, (ii) supports both fault detection and severity level estimation (i.e., relative power reduction compared to fault-free operation), and (iii) is shown to be also effective in detecting unseen/new fault types (i.e., not considered during training of the model).

Further, the *GNN-based model* performing fault detection jointly for multiple PV sites is shown to be successful for a set of PV systems even when their configurations and orientations vary substantially. For example, in our case study on 6 sites with 2 types of PV modules arranged in 6 different system sizes and configurations, we attain overall classification accuracies of 85%. Moreover the GNN approach does not require any weather information at all. Additionally, we show that our GNN model also is capable of generalizing to PV sites unseen during training (as long as they are subject to similar weather conditions, i.e., are geographically nearby).

Limitations — Since our solutions were targeted to be practical and cost-effective, they do not reach maximal performance: using dedicated sensors, approaches reaching over 99% accuracy have been reported [10, 15, 20].

From our experiments testing a trained GNN model on PV sites unseen during training, as discussed in Section 5.3, we found that the GNN model does not perform well on an isolated, remote PV system (site 6 in our case study) that is far from the sites included in the model's training set. Yet, in practice this would only be problematic in sparsely populated regions. In such cases, a dedicated model (e.g., using our RNN model) trained on local historical data of that site would be a more meaningful (i.e., better performing) solution. Further, if the remote PV system would be a large scale one, comprising multiple inverters, the GNN approach would still make sense considering each inverter individually as nodes in the model graph. Since in the latter scenario all strings essentially

are subject to almost exactly the same weather conditions, we expect the thus trained GNN model to be highly accurate.

Although our models are generic with respect to the PV module technology, climate/weather conditions, etc., of the sites to perform fault identification/classification for, we only have tested them in a limited set of case studies (and a relatively low number of sites for the GNN model). Yet, *a priori* we do expect the RNN model to perform as well for any PV system technology, configuration, and geographical location: e.g., further analysis in [1, §5.2] showed that the performance of an RNN-based model trained on North Carolina weather data was barely affected when tested in Nevada weather conditions. For the GNN model, we believe it is likely that the accuracy of the fault identification model will only improve when more (located relatively closely together) sites are considered. Considering GNNs have been applied to graphs with over 100 million nodes [38], scaling up our approach to monitor more PV systems is clearly feasible.

Note that our RNN model includes a severity level prediction component, while our considered GNN-based model does not. Yet, clearly such a component could also be added there. Still, it is unclear whether severity level prediction would work equally well, especially when considering heterogeneous PV site technologies and orientations.

Future work — From the above limitations, it is clear that a number of research questions need further analysis.

For both models, we note that we only considered fault cases where only a single fault type occurs. Thus, how to identify possibly multiple simultaneously occurring faults, and/or how well our models fare under such multi-label classification conditions, remains to be investigated. Similarly, more detailed analysis is required to assess the RNN/GNN model performance in function of time resolution (we only considered 1 h measurement intervals) and input time window size (we used a 24 h history), which can guide deployment strategies to balance performance vs. computational and sensor equipment requirements. We also note that in both model types we only considered stationary cases, i.e., we trained and evaluated on time series for which the fault was either present or absent the entire time. It would be interesting to study how to train our models (and/or tweak them) to most effectively and as quickly as possible identify faults, for time series that include the transition of fault-free conditions to the occurrence of a fault. A last common direction for both models is to validate them in the field, using actual real-world PV systems (including cases with known faults).

Specifically for the GNN model, we advocate a study to establish up to what scale, in terms of both absolute number and geographical spread of the PV sites, it is meaningful to jointly process multiple sites.

Acknowledgments

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

We further would like to thank Arnaud Schils (who was with imec at the time) for facilitating the PV simulations.

References

1. Van Gompel, J., Spina, D., Develder, C.: Satellite based fault diagnosis of photovoltaic systems using recurrent neural networks. *Appl. Energy* 305, 1–12 (Jan 2022)
2. Van Gompel, J., Spina, D., Develder, C.: Cost-effective fault diagnosis of nearby photovoltaic systems using graph neural networks. *Energy* 266(126444) (Mar 2023)
3. International Energy Agency (IEA): World Energy Outlook 2022 (Oct 2022), <https://www.iea.org/reports/world-energy-outlook-2022>
4. Madeti, S.R., Singh, S.: Online fault detection and the economic analysis of grid-connected photovoltaic systems. *Energy* 134, 121–135 (Sep 2017)
5. Livera, A., Theristis, M., Makrides, G., Georgiou, G.E.: Recent advances in failure diagnosis techniques based on performance data analysis for grid-connected photovoltaic systems. *Renew. Energy* 133, 126–143 (Apr 2019)
6. Pillai, D.S., Rajasekar, N.: A comprehensive review on protection challenges and fault diagnosis in PV systems. *Renew. Sust. Energ. Rev.* 91, 18–40 (Aug 2018)
7. Mellit, A., Tina, G.M., Kalogirou, S.A.: Fault detection and diagnosis methods for photovoltaic systems: A review. *Renew. Sust. Energ. Rev.* 91, 1–17 (Aug 2018)
8. Chine, W., Mellit, A. and Lugh, V., Malek, A., Sulligoi, G., Massi Pavan, A.: A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks. *Renew. Energy* 90, 501–512 (May 2016)
9. Kapucu, C., Cubukcu, M.: A supervised ensemble learning method for fault diagnosis in photovoltaic strings. *Energy* 227, 120463 (15 Jul 2021)
10. Garoudja, E., Chouder, A., Kara, K., Silvestre, S.: An enhanced machine learning based approach for failures detection and diagnosis of PV systems. *Energy Conv. Manag.* 151, 496–513 (Nov 2017)
11. Adhya, D., Chatterjee, S., Chakraborty, A.K.: Performance assessment of selective machine learning techniques for improved PV array fault diagnosis. *Sustain. Energy Grids Netw.* 29, 100582 (Mar 2022)
12. De Benedetti, M., Leonardi, F., Messina, F., Santoro, C., Vasilakos, A.: Anomaly detection and predictive maintenance for photovoltaic systems. *Neurocomputing* 310, 59–68 (Oct 2018)
13. Jiang, L.L., Maskell, D.L.: Automatic fault detection and diagnosis for photovoltaic systems using combined artificial neural network and analytical based methods. In: Proc. Int. Joint Conf. Neural Netw. (IJCNN 2015). pp. 1–8. Killarney, Ireland (11–16 Jul 2015)
14. Zhao, Y., Ball, R., Mosesian, J., de Palma, J.F., Lehman, B.: Graph-based semi-supervised learning for fault detection and classification in solar photovoltaic arrays. *IEEE Trans. Power Electron.* 30(5), 2848–2858 (May 2015)
15. Chen, Z., Wu, L., Cheng, S., Lin, P., Wu, Y., Lin, W.: Intelligent fault diagnosis of photovoltaic arrays based on optimized kernel extreme learning machine and I-V characteristics. *Appl. Energy* 204, 912–931 (Oct 2017)
16. Spataru, S., Sera, D., Kerekes, T., Teodorescu, R.: Diagnostic method for photovoltaic systems based on light I-V measurements. *Sol. Energy* 119, 29–44 (Sep 2015)
17. Liu, Q., Yang, B., Wang, Z., Zhu, D., Wang, X., Ma, K., Guan, X.: Asynchronous decentralized federated learning for collaborative fault diagnosis of PV stations. *IEEE Trans. Netw. Sci. Eng.* pp. 1680–1696 (May–Jun 2022)

18. Lin, P., Qian, Z., Lu, X., Lin, Y., Lai, Y., Cheng, S., Chen, Z., Wu, L.: Compound fault diagnosis model for photovoltaic array using multi-scale SE-ResNet. *Sustain. Energy Technol. Assess.* 50, 101785 (Mar 2022)
19. Gao, W., Wai, R.J., Chen, S.Q.: Novel PV fault diagnoses via SAE and improved multi-grained cascade forest with string voltage and currents measures. *IEEE Access* 8, 133144–133160 (Jul 2020)
20. Chen, Z., Han, F., Wu, L., Yu, J., Cheng, S., Lin, P., et al.: Random forest based intelligent fault diagnosis for PV arrays using array voltage and string currents. *Energy Conv. Manag.* 178, 250–264 (Dec 2018)
21. Kumar, B.P., Ilango, G.S., Reddy, M.J.B., Chilakapati, N.: Online fault detection and diagnosis in photovoltaic systems using wavelet packets. *IEEE J. Photovolt.* 8(1), 257–265 (2017)
22. Lu, X., Lin, P., Cheng, S., Lin, Y., Chen, Z., Wu, L., Zheng, Q.: Fault diagnosis for photovoltaic array based on convolutional neural network and electrical time series graph. *Energy Conv. Manag.* 196, 950–965 (Sep 2019)
23. Huuhtanen, T., Jung, A.: Predictive maintenance of photovoltaic panels via deep learning. In: Proc. IEEE Data Sci. Workshop (DSW 2018). pp. 66–70. Lausanne, Switzerland (4–6 Jun 2018)
24. Feng, M., Bashir, N., Shenoy, P., Irwin, D., Kosanovic, D.: SunDown: Model-driven per-panel solar anomaly detection for residential arrays. In: Proc. 3rd ACM SIGCAS Conf. Comput. Sustainable Soc. (COMPASS 2020). pp. 291–295. Guyacquil, Ecuador (15–17 Jun 2020)
25. Iyengar, S., Lee, S., Sheldon, D., Shenoy, P.: SolarClique: Detecting anomalies in residential solar arrays. In: Proc. 1st ACM SIGCAS Conf. Comput. Sustainable Soc. (COMPASS 2018). pp. 1–10. Menlo Park and San Jose, CA, USA (20–22 Jun 2018)
26. Schils, A., Breugelmans, R., Carolus, J., Ascencio-Vásquez, J., Wabbes, A., Bertrand, E., Aldalali, B., Daenen, M., Voroshazi, E., Scheerlinck, S.: A grey box model for shunting-type potential induced degradation in silicon photovoltaic cells under environmental stress. In: Proc. 38th Eur. Photovolt. Solar Energy Conf. Exhib. (EU PVSEC 2021). pp. 578–582. Online (6–10 Sep 2021)
27. Goverde, H., Herteleer, B., Anagnostos, D., Köse, G., Goossens, D., Aldaladi, B., J., G., Baert, K., Catthoor, F., Driesen, J., Poortmans, J.: Energy yield prediction model for PV modules including spatial and temporal effects. In: Proc. 29th Eur. Photovolt. Solar Energy Conf. Exhib. (EU PVSEC 2014). pp. 3292–3296. Online (22–26 Sep 2014)
28. Anagnostos, D., Goverde, H., Herteleer, B., Catthoor, F., Dimitrios, S., Driesen, J., Poortmans, J.: Demonstration and validation of an energy yield prediction model suitable for non-steady state non-uniform conditions. In: Proc. 6th World Conf. Photovolt. Energy Convers. Kyoto, Japan (23–27 Nov 2014)
29. Appiah, A.Y., Zhang, X., Ayawli, B.B.K., Kyeremeh, F.: Long short-term memory networks based automatic feature extraction for photovoltaic array fault diagnosis. *IEEE Access* 7, 30089–30101 (Mar 2019)
30. Jager, D., Andreas, A.: NREL National Wind Technology Center (NWTC): M2 Tower; Boulder, Colorado (Data). NREL Report No. DA-5500-56489 (1996)
31. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), <http://www.deeplearningbook.org>
32. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: CatBoost: unbiased boosting with categorical features. In: Proc. 32nd Int. Conf. Neural Inf. Process. Sys. (NIPS 2018). pp. 6639–6649. Montreal, Canada (3–8 Dec 2018), https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf

33. Gelaro, R., McCarty, W., Suárez, M.J., Todling, R., Molod, A., Takacs, L., Randles, C.A., Darmenov, A., Bosilovich, M.G., Reichle, R., Wargan, K., Coy, L., Cullather, R., Draper, C., Akella, S., Buchard, V., Conaty, A., da Silva, A.M., Gu, W., Kim, G.K., Koster, R., Lucchesi, R., Merkova, D., Nielsen, J.E., Partyka, G., Pawson, S., Putman, W., Riendecker, M., Schubert, S.D., Sienkiewicz, M., Zhao, B.: The modern-era retrospective analysis for research and applications, version 2 (merra-2). *J. Climate* 30(14), 5419–5454 (Jan 2017)
34. Maguire, J.B., Grattarola, D., Mulligan, V.K., Klyshko, E., Melo, H.: XENet: Using a new graph convolution to accelerate the timeline for protein design on quantum computers. *PLOS Comput. Biol.* 17(9), 1–21 (Sep 2021)
35. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proc. 5th Int. Conf. Learn. Represent. (ICLR 2017). pp. 1–14. Toulon, France (24–26 Apr 2017), <https://openreview.net/forum?id=SJU4ayYgl>
36. Simonovsky, M., Komodakis, N.: Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: Proc. IEEE Conf. Comput. Vis. Patt. Recogn. (CVPR 2017). pp. 29–38. Honolulu, HI, USA (22–25 Jul 2017)
37. Xie, T., Grossman, J.C.: Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.* 120, 145301 (Apr 2018)
38. Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., Leskovec, J.: Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430* (2021)