

Physics Informed Deep Learning for Tracker Fault Detection in Photovoltaic Power Plants

Jannik Zraggen¹, Yuyan Guo², Antonio Notaristefano³ and Lilach Goren Huber⁴

^{1,4} *Zurich University of Applied Sciences, Technikumstrasse 9, Winterthur, 8400 Switzerland*

jannik.zraggen@zhaw.ch

lilach.gorenhuber@zhaw.ch

^{2,3} *Fluence Energy, Hornbachstrasse 50, Zurich, 8008 Switzerland*

yuyan.guo@fluenceenergy.com

antonio.notaristefano@fluenceenergy.com

ABSTRACT

One of the main challenges for fault detection in commercial fleets of machines is the lack of annotated data from the faulty condition. The use of supervised algorithms for anomaly detection or fault diagnosis is often unrealistic in this case. One approach to overcome this challenge is to augment the available normal data by generating synthetic anomalous data that represents faulty conditions. In this paper we apply this approach to the detection of faults in the tracking system of solar panels in utility-scale photovoltaic (PV) power plants. We develop a physical model in order to augment the training data for a deep convolutional neural network. We show that the physics informed learning algorithm is capable of detecting faults in an accurate and robust manner under diverse weather conditions, outperforming a purely data-driven approach. Developing and testing the algorithm with real operational data ensures its efficient deployment for PV power plants that are monitored at string level. This in turn enables the early detection of root causes for power losses, thereby contributing to the accelerated adoption of solar energy at utility scale.

1. INTRODUCTION

A central and very common challenge for commercial applications of machine learning (ML) algorithms for machine fault detection and diagnosis is the lack of annotated data under faulty machine conditions. This makes the task of training fully supervised fault detection and diagnosis (FDD) algorithms using real operational data close to impossible (Fink et al., 2020).

As a result, a very common approach to fault detection is

based on training data from normal (healthy) conditions exclusively. In this case, anomalies are detected as deviations from the machine behavior expected by the normal state model (Fink et al., 2020). The disadvantage of such models, however, is that they typically allow for anomaly detection but do not enable diagnosis of the root cause of the anomaly. In order to proceed towards data-driven diagnostics, a combination of engineering and physical knowledge is essential (Rausch, Goebel, Eklund, & Brunell, 2005). The benefit of hybrid approaches, combining physics with machine learning methods has been demonstrated in diverse application fields and can be achieved in various manners (Karniadakis et al., 2021). One way is by augmenting the training data based on physical models. In PHM applications, such data augmentation can be done by simulating the system behavior under healthy conditions as well as under degraded or faulty conditions, and using data from the simulation model to train ML models.

In this paper we suggest a somewhat different hybrid approach: instead of using physical models to simulate the entire system, we use it to artificially synthesize faulty data patterns from existing healthy data. The synthetic fault data can then be used to augment the real operational healthy data in order to train a binary classifier between healthy and faulty input data. The performance of the classifier is then tested on operational data which includes real faults.

We demonstrate the above hybrid approach to fault detection in utility-scale operational photovoltaic (PV) power plants. One of the common fault mechanisms in PV plants are tracker faults. Solar trackers are devices that orient the solar panels towards the sun, thereby maximizing the amount of energy produced from a fixed amount of installed power generating capacity (Racharla & Rajan, 2017). A tracker fault usually occurs when the tracker gets stuck at a certain orientation in-

Jannik Zraggen et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

stead of tracking the sun.

Tracker faults can lead to a significant reduction in the power produced by the PV strings that are mounted on the faulty trackers. Early and automatic detection, diagnosis and localization of such faults can therefore prevent large production losses, thereby increasing the cost-effectiveness of solar energy and thus helping to accelerate the transition towards renewable energy sources. Despite the potential efficiency gains, solar tracker FDD has not yet been addressed in the scientific literature.

ML and in particular Deep Learning (DL) algorithms have been developed for FDD of other fault types in solar plants (Mellit, Tina, & Kalogirou, 2018; Haque, Bharath, Khan, Khan, & Jaffery, 2019; Mansouri, Trabelsi, Nounou, & Nounou, 2021; Pillai & Rajasekar, 2018; Triki-Lahiani, Abdelghani, & Slama-Belkhdja, 2018). However, only very few of those algorithms have reached commercial deployment. Some of the reasons for this are related to the development process of the algorithms. In most published researches, the algorithms are trained and tested on synthetic data, either from simulations (Chine et al., 2016) or from well-controlled experiments on small or medium scale PV devices (Chen, Chen, Wu, Cheng, & Lin, 2019; Li, Delpha, Diallo, & Migan-Dubois, 2021; Gao & Wai, 2020). In such conditions, it is difficult to induce realistic noise sources such as mixture of several fault types and diversity of operating conditions. Moreover, synthetic data can be generated at an arbitrary granularity level: except for very few research works (Guerriero, Piegari, Rizzo, & Daliento, 2017; Skomedal, Øgaard, Selj, Haug, & Marstein, 2019), all models assume data availability from single PV panels, despite the fact that real operational data is usually available at the level of PV-strings or even inverters, potentially gathering information from hundreds or thousands of panels. Thus, algorithms that are developed under synthetic conditions are not directly usable for commercial deployment. Another class of research focuses on deep learning algorithms for FDD based on image data collected by drones (Daliento et al., 2017). Since this is an expensive solution, requiring designated hardware, such algorithms are rarely deployed operationally.

In the following we suggest an algorithm for automatic tracker fault detection based on operationally available power data and use physics informed deep learning (PIDL) to augment the data. We train a Convolutional Neural Network (CNN) using data from an operational PV plant under various healthy (normal) conditions. The plant is monitored at string level, thus the input to the CNN is the measured produced power of the single PV strings. The training data is augmented by synthesizing faulty data from healthy data using a physical model that accounts for the mechanism of tracker faults. As we show below, the hybrid approach of combining available data with physical models enables accurate fault detection on

unseen data from the operational PV plant.

The contributions of this paper are the following:

- We suggest an algorithm for tracker FD in grid-scale PV plants that is developed and tested on data from a real operational plant. As such, it is ready for deployment on string-level monitored PV plants with no need for additional data acquisition.
- The algorithm combines deep learning with physical models. As a result it allows for high fidelity fault detection even in the absence of real data under faulty conditions. Moreover, the approach is extendable to fault classification given physical models for more fault mechanisms in this system.
- The physical knowledge is not used to model the entire system under healthy conditions. Instead, it is used to incorporate a known fault mechanism in the real operational healthy data. This PI approach is capable of capturing various complex properties of the real data without having a precise physical model for them.
- We introduce a data-driven pre-processing of the inputs and show that it helps to prevent over-fitting to the synthetic fault data. This enables tracker FD on real data even under complex weather conditions such as cloud coverage, without the need for additional satellite data.
- We introduce physics informed stochastic contributions into the fault simulation model. We show that this improves the performance and the robustness of the fault detection task.
- High performance FDD with operational data is often regarded as a challenge, due to the strong heterogeneity and complexity of such data. Our approach to data augmentation takes advantage of this heterogeneity, using it to diversify the training data in a physical manner.

2. METHOD

In order to bridge the gap between the extensive body of research on one hand and the dilute commercial deployment on the other hand, FDD algorithms for PV plants must rely on real operational data. One obvious challenge here is the scarcity of annotated data from faulty regimes. Since faults are rare and very diverse in nature, this is a general challenge in PHM applications aimed at detecting and diagnosing faults in technical assets.

In this paper we tackle this problem by synthesizing data under faulty conditions. The faulty data is generated by combining a physical model of the fault mechanism with available healthy data from an operational power plant. For simplicity we focus in the present research on a single fault type, tracker faults. The fault occurs when a tracker stops at a fixed angle instead of tracking the sun at any given moment. The effect of a tracker fault is particularly easy to observe if we look

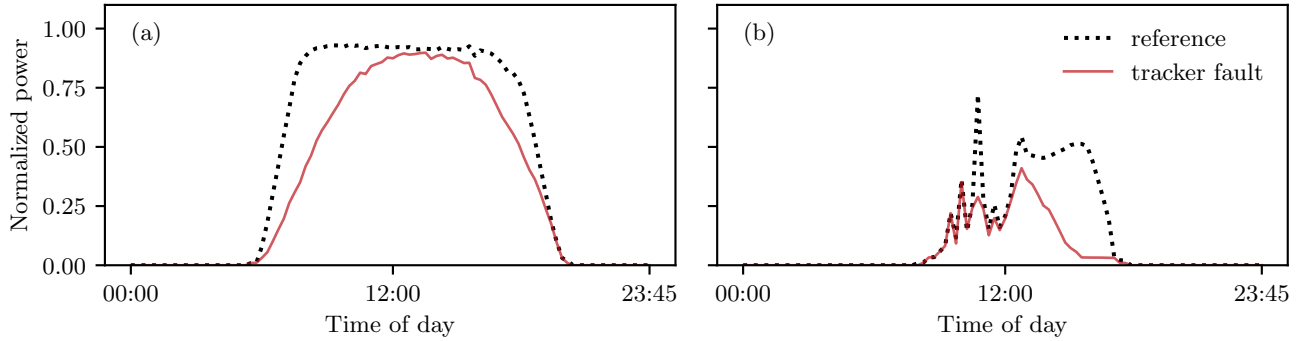


Figure 1. Daily profiles with and without tracker faults. A plant-wide reference power production (black dotted) is compared with the power produced by a string mounted on a faulty tracker (red solid). The normalized power (normalized by the maximal possible power for a single string) is plotted against time during two days: (a) a day with clear sky (b) a day with a strong cloud coverage.

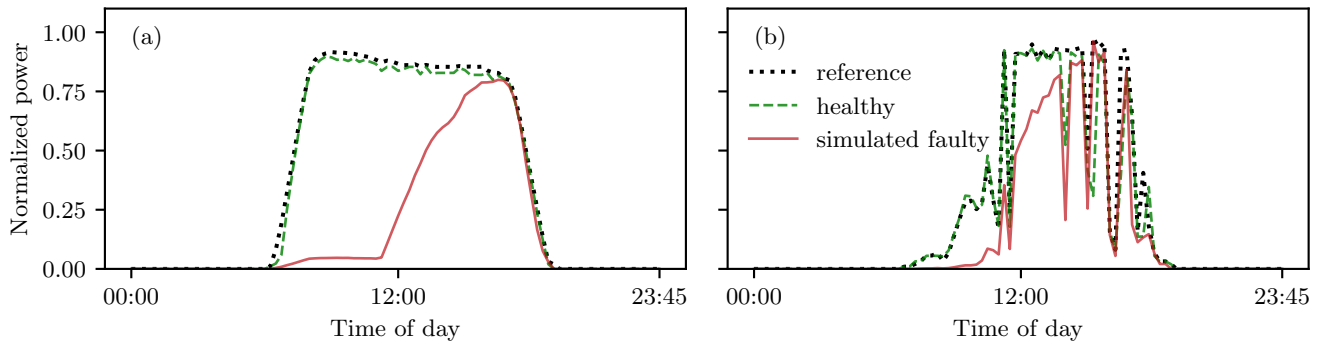


Figure 2. Daily profiles with and without simulated tracker faults. A plant-wide reference power production (black dotted) is compared with the simulation results for the power produced by string mounted on a faulty tracker (red solid), and with a real healthy profile used for the simulation (green dashed). The normalized power (normalized by the maximal possible power for a single string) is plotted against time during two days: (a) a day with clear sky (b) a day with a strong cloud coverage.

at a daily power production profile of a single PV string on a day with a clear sky (no clouds). This is shown in Figure 1(a). The solid red curve is a daily profile of a string mounted on a faulty tracker that was stuck when oriented towards west. The dashed line represents a reference for a healthy string. The calculation of the daily reference power is explained below in subsection 2.2. Figure 1(b) shows a similar scenario on a cloudy day, demonstrating the large diversity of daily profiles depending on the environmental conditions.

The data we use for this research is the produced power of an operational PV plant with 624 strings. The power is available for each string with a time resolution of 15 minutes. The entire data set includes data from several years. Here we use one year data for training the algorithms and another year for testing. It is important to note that operational solar power data displays a wide variety of phenomena such as heavy cloud coverage, sporadic clouds over part of the plant, partial shad-

ing, as well as natural degradation, and under-performance of random strings. A key challenge is to detect daily profiles affected by tracker faults in such heterogeneous and noisy data.

In the following subsection we describe the model used in order to synthesize daily profiles of PV strings that suffer from power losses due to tracker faults.

2.1. Synthetic fault generation

Daily power profiles of strings with tracker faults are generated by corrupting daily profiles of healthy strings, measured in a large scale operational solar plant. The corruption is done based on a physical model of a tracker, which can be stuck at a certain tilt angle. As a result, the tilt angle of the panels mounted on this tracker is no longer optimal at any given moment, and the power production from the panels is reduced compared to the optimum. The total produced power of a string s depends on the irradiance absorbed by its

PV-panels. This irradiance is composed of three components: (i) the beam contribution $G_s^{(B)}$ which is the direct sun light reaching the panels (ii) the ground component $G_s^{(G)}$, denoting light that reaches the panels after being reflected off the ground and (iii) the diffuse component $G_s^{(D)}$ denoting light resulting from all other reflections. For the sake of simplicity we assume that the diffuse component and ground reflected component are independent of the tilt angle and approximate the total irradiance onto a string s as

$$G_s = G_s^{(B)} + G_s^{(D)}, \quad (1)$$

where $G_s^{(D)}$ denotes the total irradiance of all diffuse sources. The beam component is given by the direct normal irradiance G_N reduced by $\cos \theta_i$ where θ_i is the angle of incidence. The irradiance absorbed by the panels is further reduced by small IAM losses (array-incidence losses) which we estimate using the Ashrae parametrization (A.F. Souka, 1966); $f_{IAM}(\theta_i) = 1 - b_0(1/\cos \theta_i - 1)$. The beam component at incidence angle θ_i is thus

$$G_s^{(B)} = G_N \cos \theta_i \cdot f_{IAM}(\theta_i) \quad (2)$$

The IAM reduction of the diffuse component is neglected. In addition we assume that if the tracker is intact, the diffuse component amounts to a fixed fraction of the total irradiance, $G_s^{(D)} = \gamma G_s$. This allows us to express the total irradiance G_s for string s which is tilted at an arbitrary incidence angle θ_i as function of the expected irradiance G_s^* at the optimal incidence angle θ_i^* :

$$G_s = \left[(1 - \gamma) \frac{\cos \theta_i \cdot f_{IAM}(\theta_i)}{\cos \theta_i^* \cdot f_{IAM}(\theta_i^*)} + \gamma \right] G_s^* \quad (3)$$

The angle θ_i^* can be calculated using the python library pvlbl (William F. Holmgren & Mikofski, 2018) to implement the NREL model for the optimal tracker tilt angle (Anderson & Mikofski, 2020). Thus, from a given power production profile $x_h(t)$ of a normal functioning string, assuming it is optimally tracked and follows $\theta_i^*(t)$ at any moment t during the day, we can generate a faulty profile $x_f(t)$ of a string which is stuck at a fixed angle θ_0 using

$$x_f(t) = \left[(1 - \gamma) \frac{\cos \theta_0 \cdot f_{IAM}(\theta_0)}{\cos \theta_i^*(t) \cdot f_{IAM}(\theta_i^*(t))} + \gamma \right] x_h(t). \quad (4)$$

The faulty profiles for the training set are created by sampling different values of θ_0 out of a uniform distribution. In this way we cover a variety of possible tracker faults. Initial values for the model parameters b_0 and γ are estimated based on physical knowledge. The values are further calibrated using 10 samples of faulty profiles from the operational data (that are then excluded from the test set) and fitting them by minimizing the RMSE. This yields the estimated values of $b_0 = \gamma = 0.05$. However, as will be demonstrated below, the model parameters do not have to be calibrated very precisely,

as they will eventually be drawn from a uniform distribution around the estimated values in order to allow the synthesis of various fault characteristics.

It is important to note that the method we use here allows us to generate faulty profiles on clear days as well as on cloudy days, since the corrupted faulty profile is expressed in relation to the expected healthy profile of the same day, irrespective of the weather conditions. In addition, using real operational power profiles as a starting point for fault synthesis naturally covers a wide variety of realistic situations (related to environmental or operational conditions), thus augmenting the training data in a physical manner. Moreover, it allows us to avoid using meteorological information such as irradiance, which can suffer from inaccuracies. In other words, the physics informed approach we introduce does not involve a full model of the normal functioning system. Instead, it incorporates faults into the available operational data from the real healthy system. This approach accounts for various complex properties of the healthy data without the need for an accurate physical model that captures all of them.

Figure 2 shows two examples of faulty profiles generated from healthy profiles in the way described above. The profile in 2(a) was taken under clear sky and the one in 2(b) under cloudy conditions. The two plots show in dotted lines the reference profiles of the same day, representing healthy power production given the daily irradiance profile. The calculation of the daily reference power is explained below in subsection 2.2.

2.1.1. Improved fault simulation: additional power losses

A further step towards synthesizing realistic fault profiles can be achieved by modeling a certain amount of additional power losses in addition to tracker faults. These can result for example from small amounts of soiling or panel degradation that are assumed to reduce the power production by a fixed factor,

$$x_{fn}(t) = C_p \cdot x_f(t) \quad (5)$$

where the power factor C_p is a fraction between 0 and 1. When synthesizing the faulty profiles x_{fn} we draw the factor C_p out of a uniform distribution between 0.8 and 1, in order to model a variable small amount of performance losses. Allowing for a random modification of the daily profiles helps to augment the training data in a physics informed manner.

In Section 3 we compare the FD performance of two classifiers based on different models for the synthetic faulty profiles, thus corresponding to two data augmentation schemes:

Model A: the faulty profiles $x_f(t)$ are generated using Eq. 4 alone to simulate tracker faults.

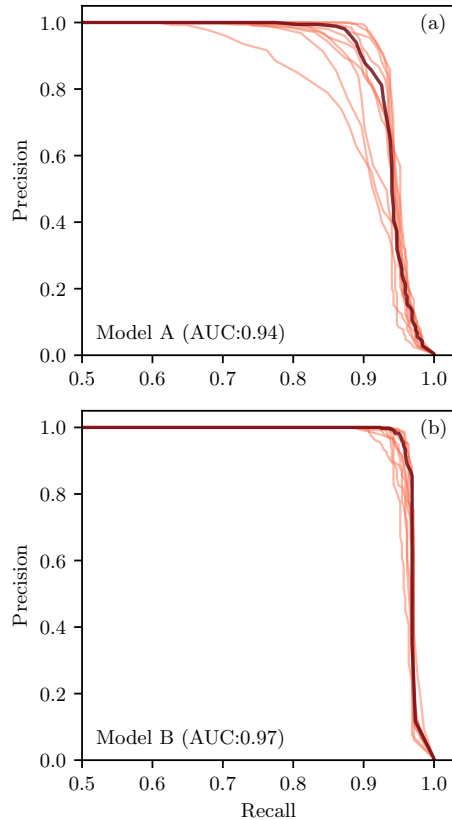


Figure 3. Precision-recall curves for two physics informed models. (a) Model A generates synthetic tracker fault data for training. (b) Model B includes additional power losses on top of tracker faults. The precision and recall are evaluated on one year of operational data including healthy periods and tracker faults. Each panel shows the results of 10 training runs and their median (thick curve). The AUC of the median curve is given in brackets.

Model B: the faulty profiles $x_{fn}(t)$ are generated using Eq. 4 followed by Eq. 5, thus simulating mixed effects of tracker faults with additional power losses. The healthy training profiles are modified using Eq. 5 only. We introduce additional stochastic elements into the entire training set by allowing the values of the constants b_0 and γ to be drawn from normal distributions around the empirical values of 0.05 when generating synthetic faulty profiles.

2.2. CNN for fault classification

The synthetic generation of faulty profiles allows us to generate an arbitrary amount of faulty samples. As a result, the fault detection task can be carried out by means of supervised classification methods, trained with balanced inputs from both classes. The synthetic faulty power profiles are used together with real healthy profiles to train a binary classifier. The training data is extracted from one year of operational data

of a PV plant with 624 strings. It includes 170'000 healthy daily profiles, each one of a single string, and 170'000 synthetic faulty profiles generated from the healthy profiles as explained above.

For the classification task we use a convolutional neural network (CNN) with three one-dimensional convolutional layers followed by two fully-connected layers. The convolutional layers enable time-correlated feature extraction from the time-series inputs. The network architecture was optimized using a grid search on a validation data set, resulting in 30'000 trainable parameters. The hyperparameters which were tuned this way are the number of convolutional layers, the number of filters and the learning rate.

The raw power data is pre-processed before feeding it into the CNN. The pre-processing includes calculating a reference for a healthy daily power profile for each day. The reference profile $x_r(t)$ is calculated by taking the 0.9 quantile over all strings in the plant at time t , such that $x_r = F^{-1}(0.9)$, where $F(x)$ is the empirical cumulative distribution function. For each daily profile $x(t)$ we then calculate the deviation from the reference profile,

$$x_d(t) = x_r(t) - x(t). \quad (6)$$

The input into the CNN is the deviation profile $x_d(t)$ rather than the raw daily profile $x(t)$. This allows an accurate detection of deviation from the normal power production irrespective of the weather conditions. In subsection 3.1.1 we demonstrate the advantage of the pre-processing step over using raw power profiles as inputs.

3. RESULTS

To evaluate the performance of the method we use test data of one year from the same operational PV plant used for training. During this year we manually labeled 417 daily profiles as clearly suffering from tracker faults, and 94'333 profiles as healthy. Other known fault types or unclear cases were filtered out from the training and the test data.

Figure 3 shows the healthy/faulty classification performance in terms of precision-recall plots. In panel (a) the synthetic fault profiles of the training data were generated using simulation model A, i.e without simulating additional power losses. Panel (b) displays the results when the CNN was trained with faulty profiles generated using model B, which includes small additional power losses on top of tracker faults as explained in Subsection 2.1.1. In each panel we display curves of 10 training runs along with the curve resulting from the ensemble median classifier. The runs differ from each other in the following stochastic elements: (i) random weight initialization (ii) random batch splits (iii) randomly drawn parameters of the physical model which generates the faulty profiles. This implies that the faulty training data is different in each of

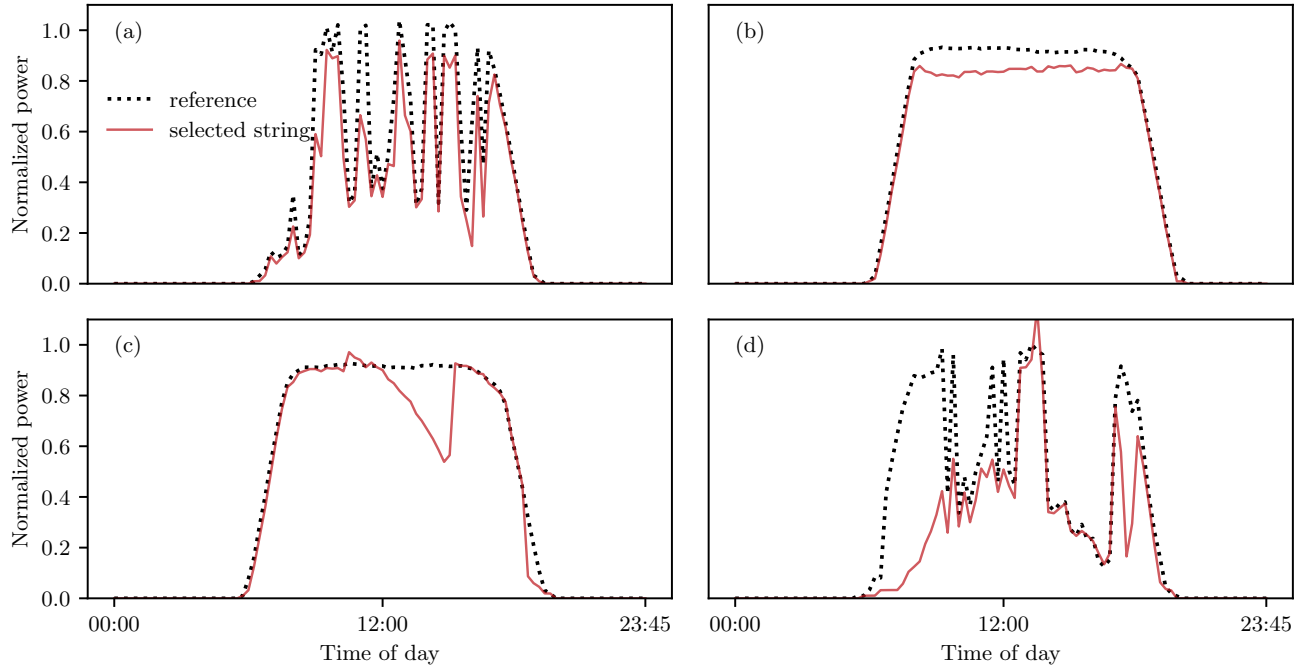


Figure 4. Performance comparison of two data-augmentation models. Four daily (normalized) power production profiles are shown as examples for the improved fault detection performance of model B compared to model A. (a) and (b) are days with no true tracker faults. Model A wrongly detected them as faulty with anomaly scores 0.88 and 0.95 respectively. Model B identified them as healthy with scores 0.005 and 0.02 respectively. (c) and (d) are days with true tracker faults that were not detected with model A (scores 0.02 and 0.09) but detect with model B (scores 0.91 and 1.0).

the 10 runs. In model B the healthy training data is randomly modified as well. A comparison of the two panels of Fig. 3 shows that accounting for possible added power losses in the data leads to a considerable improvement of the classification performance with an Area Under Curve (AUC) of 0.97 with model B compared to 0.94 with model A. Even more importantly, the classification results based on model B are significantly more reproducible, with a lower variability over multiple training runs, and is therefore more robust against stochastic effects which are typical for noisy operational data.

Figure 4 displays typical examples in which model B outperforms model A. The red solid power profiles in panels (a) and (b) were measured in strings with no tracker faults. The reduced power production observed in these strings compared to the plant reference for the same day (black dotted curve) results from root causes that are unrelated to tracker faults. However, using model A these profiles obtained an anomaly score close to 1 (in this case the classifier outputs were 0.88 and 0.95 for (a) and (b) respectively). This means that they would be classified as "tracker fault" with a very high confidence. On the other hand, model B correctly classified them as healthy with an anomaly score close to 0 (in this case 0.005 and 0.02 for (a) and (b) respectively).

panels (c) and (d) demonstrate the opposite case: the two red

curves correspond to strings suffering from true tracker faults that were clearly observed using the direct tracker positions (available in this case for labeling purposes, but not available in all operational plants). The two profiles were classified as healthy (anomaly scores 0.02 and 0.09 respectively) by model A, and as faulty (anomaly scores 0.91 and 1.0 respectively) by model B.

The four examples above demonstrate that model B, which is trained to recognize additional power losses, can distinguish such losses from losses due to tracker faults. It thus correctly classifies string power profiles with such losses as "healthy" with respect to tracker faults, and is not likely to generate false positives in this case. Moreover, it detects true tracker faults even under complex conditions, i.e in cases of combined losses due to tracker faults and other loss mechanisms (e.g cloudy weather or degraded performance). These two facts leads to a significantly improved FD performance and increased robustness of model B compared to model A.

3.1. Benchmarking

In order to justify the most important contributions of our physics informed model we compare the results with two benchmarks. The first one is aimed at evaluating the importance of the pre-processing step, in which we generate the deviation

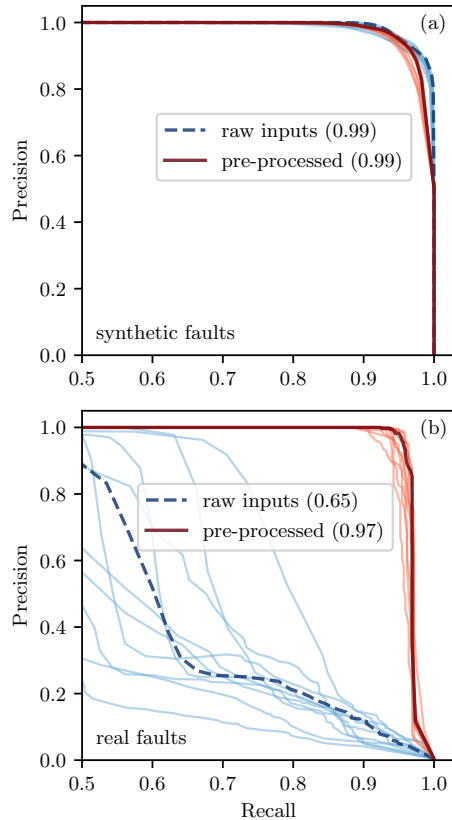


Figure 5. The effect of pre-processing the inputs using a daily reference profile. The fault classification performance of two models is evaluated in terms of their precision-recall curves. A physics informed (PI) classifier with raw power profiles as inputs is compared to a PI classifier with pre-processed profiles as inputs. The performance is evaluated on two test data sets: (a) with synthetic faults simulated using model B (b) with real annotated faults from operational data. Each panel shows the results of 10 training runs and their median (thick curves). The AUC of the median curve is given in brackets.

from a data-driven reference power profile. The second one evaluates the significance of incorporating physical information in the DL model.

3.1.1. Raw vs. pre-processed inputs

As inputs to the classifier CNN we use the daily difference $x_d(t)$ between the produced power $x(t)$ and a reference power production $x_r(t)$, as explained in section 2.2. It turns out that using the power production difference instead of the power itself is an important pre-processing step prior to the training. In order to demonstrate the benefit of this step, we compare two classifiers. The first one is trained to classify the raw power production profiles $x(t)$, without pre-processing the data using a daily reference profile. The second one is trained to classify $x_d(t)$, the power differences compared to

a daily reference power production calculated every day by taking the 0.9 quantile over all strings. In both cases we use model B for the synthetic generation of faulty profiles, such that the only difference between the models is in the pre-processing of the data. Figure 5 shows the results of the comparison with and without the daily reference, which are denoted as "pre-processed" and "raw inputs" respectively. In panel (a) we compare the precision-recall curves of the two models, evaluated on a data set with synthetically generated tracker faults. In panel (b) the performance is evaluated using a full year of operational data including real tracker faults. The performance on synthetic faults shows that both classifiers were trained properly to classify daily power profiles into healthy and faulty, independent of whether the data is pre-processed or not. However, the generalization of the two classifiers to real faults is strikingly different: our suggested model, using the pre-processed profiles performs almost as well on real faults ($AUC = 0.97$) as on simulated faults ($AUC = 0.99$). In contrast to this, the model which classifies the raw daily profiles, shows a very poor performance on real faults, with weak reproducibility, that is, very strong fluctuations between different training runs. This behavior suggests that training the CNN to classify the relative power production with respect to a plant-wide reference production helps to avoid over-fitting to the synthetic faults, and enables excellent generalization from synthetic to real data. A detailed analysis of the results provides an explanation to this fact: the "raw" classifier has a hard time to distinguish power losses due to tracker faults from power losses due to cloud coverage, whereas the classifier of the pre-processed relative profiles avoids this problem by accounting for cloud coverage in the reference profile.

We conclude that using a robust data-driven method to derive a daily power production reference, and training the CNN to classify the relative power profiles compared to this reference enable a highly accurate and reproducible tracker fault detection, even under bad weather conditions.

3.1.2. PIDL vs. a purely data-driven model

The augmentation of the training data based on a physical fault model allows us to implement supervised learning in the absence of true fault labels. A common alternative is a semi-supervised approach also known as normal state models. In this case a ML model is trained with normal (healthy) data exclusively and deviations from the predicted normal condition are monitored in real time in order to detect anomalies. A widely used normal state model is an autoencoder (AE) which is trained to reconstruct healthy signals. At prediction time, large reconstruction errors are associated with faulty conditions, thereby serving as anomaly or fault indicators (Zhou & Paffenroth, 2017). Such an anomaly detection algorithm requires no data under faulty conditions, and can be trained with healthy data only, making the physics informed fault

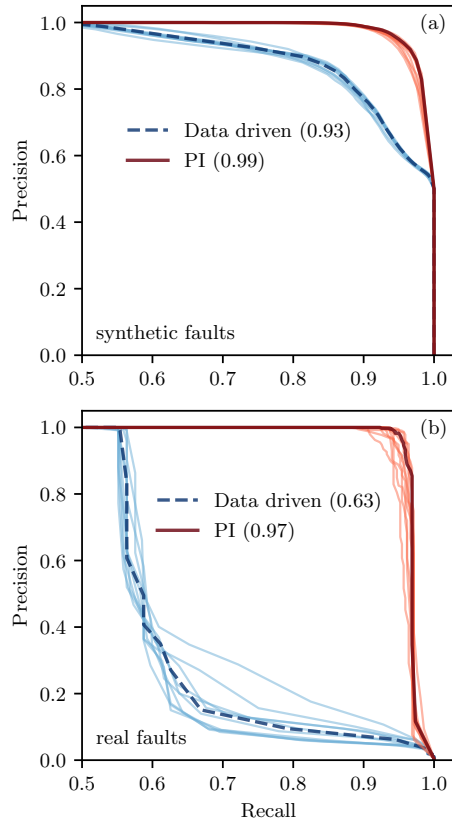


Figure 6. Physics informed vs. pure data driven models. The fault classification performance of two models is evaluated in terms of their precision-recall curves. A purely data-driven convolutional autoencoder is compared with a Physics Informed (PI) deep learning model on two test data sets: (a) with synthetic faults simulated using model B (b) with real annotated faults from operational data. For each model we show the results of 10 training runs and their median (thick curve). The AUC of the median curve is given in brackets.

synthesis obsolete. In the following we compare the pure data-driven approach based on Convolutional AE (CAE) with our suggested physics informed (PI) deep learning approach. We note that we chose an AE architecture with convolutional layers in order to compare to a model with similar feature extraction layers as in the PI classifier based on a CNN. We note that each one of the two model architectures was optimized individually on the validation data, resulting in a similar number of trainable parameters. Based on the observation that the pre-processed inputs improve the performance compared to the raw signals, we train both the PI and the data-driven models with the same pre-processed input data $x_d(t)$. In this way the two models differ primarily in the presence or absence of physical information about the fault mechanism in addition to the data. The CNN classifier utilizes this physical information whereas the CAE is fed with operational data only. This allows us to evaluate the benefit of augmenting the data

based on physical information. A detailed comparison with other classifiers and/or other anomaly detection approaches goes beyond the scope of this paper.

Figure 6(a) compares the precision-recall performance of the data-driven and the PI models, evaluated on the test data set with synthetic generated faults. The performance of the PI model, which has been trained with data drawn from the same distribution, is clearly better. Nevertheless, the purely data-driven CAE model achieves a good performance on this data. This situation is very different when comparing the performance of the two models on the operational test data containing one year of data from both healthy and faulty trackers, as shown in Figure 6(b). In this case the data-driven performs poorly whereas the PI model performs almost as well as with synthetically generated faults. This implies that supplementing the data-driven model with physical information about the fault mechanism allows the model to generalize much better from synthetic to real faults. Moreover, the outcome reproducibility of the PI model in multiple training runs is considerably higher. Its higher FD performance together with its robustness against stochastic effects make the PI model very attractive for operational deployment.

For a practical deployment of the PI framework, an operational point along the PR curve must be selected. This amounts to setting a specific threshold for the detection of anomalies, which must be done irrespective of the method in use. Considering the fault detection task at hand, a threshold which corresponds to a very high precision is often favored from a practical point of view. Such a threshold ensures a low false positive rate, thereby avoiding a situation in which technicians are sent to a remote location unnecessarily due to a false alarm.

In Figure 7 we compare the performance of various models at such a high precision operational point. For each model we fixed a classification (or fault detection) threshold that guaranteed a precision score of 0.99 on the test data. We then compare the resulting confusion matrices of three models: (i) the data driven CAE (ii) the PI model trained with faults generated with model A (iii) the PI model trained with faults generated with model B. In panels (a)-(c) we compare the confusion matrices of the models on a test set with synthetic faults, where healthy and faulty profiles are equally represented. As expected, for a given precision score (corresponding to a given false positive ratio) both PI models obtain a lower fraction of false negatives (FN) than the purely data driven model. This means that the PI models are capable of detecting faults which are missed by the data driven model. Interestingly, the PI model A performs better on the synthetic faults than the PI model B, which accounts for stochastic effects in the training data. This is, however, no longer the case when the models are tested on operational data with real faults, see panels (d)-(f). In the latter case, the stochastic ef-

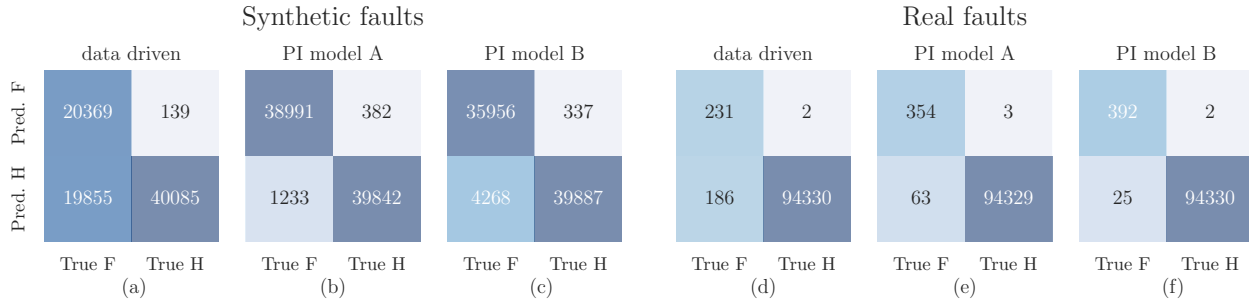


Figure 7. Confusion matrices for model comparison. The confusion matrices for three models: (i) a purely data driven model (ii) a PIDL model trained with faults generated using model A (iii) a PIDL model trained with faults generated using model B. The models are compared on two test sets: a balanced test set with synthetic faults in (a)-(c) and a test set of one year of operational data with real faults in (d)-(f).

fects inserted into model B improve the generalization capabilities of this model. This is in contrast to model A, whose performance is considerably worse when tested on real faults, implying that this model over-fits the synthetic training data. For a fixed precision score of 0.99, the PI model B reduces the missed detections to $FN = 25$, compared to 63 with PI model A, and 186 with the data driven model.

The test data we used in this research originates from a single PV plant. In future research we will test the accuracy and robustness of our approach on data from other plants, as well as extend the outcomes to plants that are monitored at inverter level only. Potential improvements of the simulation model include accounting for the effect of clouds more accurately (e.g by exploring higher values of the parameter γ which are more appropriate for cloudy days). An obvious future goal is the extension of this algorithm to be able to diagnose all common fault mechanisms in utility scale PV plants.

4. CONCLUSIONS

We developed a physics informed deep learning framework for the automatic detection of tracker faults in utility scale PV plants. A classification CNN is fed with training data containing real healthy daily power profiles of single strings and augmented by synthetic faulty profiles from a physics-based simulation of tracker faults. We demonstrate the accurate and robust fault detection enabled by this algorithm on test data from a real operational PV power plant. Moreover, we demonstrate the importance of two key ingredients of our approach, which allow for generalization of the task to the operational data. The first one is a pre-processing step of the input data, using a purely data-driven reference for normal power production. The second one is the combination of a physics informed model and a deep-learning algorithm. We showed that a PIDL model significantly outperforms a purely data-driven DL model, both in accuracy and in robustness. In future research the approach will be extended to include additional fault types, and its validity will be tested on data from

multiple plants.

ACKNOWLEDGMENT

This research was funded by Innosuisse - Swiss Innovation Agency under grant No. 55018.1 IP-ICT. The Authors would like to thank Manuel Arias Chao for his valuable comments to the manuscript.

REFERENCES

- A.F. Souka, H. S. (1966). Determination of the optimum orientations for the double-exposure, flat-plate collector and its reflectors. *Solar Energy*, 10.
- Anderson, K., & Mikofski, M. (2020). *Slope-aware backtracking for single-axis trackers* (Tech. Rep.). National Renewable Energy Lab.(NREL), Golden, CO (United States).
- Chen, Z., Chen, Y., Wu, L., Cheng, S., & Lin, P. (2019). Deep residual network based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions. *Energy Conversion and Management*, 198, 111793.
- Chine, W., Mellit, A., Lughy, V., Malek, A., Sulligoi, G., & Pavan, A. M. (2016). A novel fault diagnosis technique for photovoltaic systems based on artificial neural networks. *Renewable Energy*, 90, 501–512.
- Daliento, S., Chouder, A., Guerriero, P., Pavan, A. M., Mellit, A., Moeini, R., & Tricoli, P. (2017). Monitoring, diagnosis, and power forecasting for photovoltaic fields: A review. *International Journal of Photoenergy*, 2017.
- Fink, O., Wang, Q., Svensen, M., Dersin, P., Lee, W.-J., & Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence*, 92, 103678.
- Gao, W., & Wai, R.-J. (2020). A novel fault identification method for photovoltaic array via convolutional neural

- network and residual gated recurrent unit. *IEEE access*, 8, 159493–159510.
- Guerriero, P., Piegari, L., Rizzo, R., & Daliento, S. (2017). Mismatch based diagnosis of pv fields relying on monitored string currents. *International Journal of Photoenergy*, 2017.
- Haque, A., Bharath, K. V. S., Khan, M. A., Khan, I., & Jaffery, Z. A. (2019). Fault diagnosis of photovoltaic modules. *Energy Science & Engineering*, 7(3), 622–644.
- Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422–440.
- Li, B., Delpha, C., Diallo, D., & Migon-Dubois, A. (2021). Application of artificial neural networks to photovoltaic fault detection and diagnosis: A review. *Renewable and Sustainable Energy Reviews*, 138, 110512.
- Mansouri, M., Trabelsi, M., Nounou, H., & Nounou, M. (2021). Deep learning based fault diagnosis of photovoltaic systems: A comprehensive review and enhancement prospects. *IEEE Access*.
- Mellit, A., Tina, G. M., & Kalogirou, S. A. (2018). Fault detection and diagnosis methods for photovoltaic systems: A review. *Renewable and Sustainable Energy Reviews*, 91, 1–17.
- Pillai, D. S., & Rajasekar, N. (2018). A comprehensive review on protection challenges and fault diagnosis in pv systems. *Renewable and Sustainable Energy Reviews*, 91, 18–40.
- Racharla, S., & Rajan, K. (2017). Solar tracking system—a review. *International journal of sustainable engineering*, 10(2), 72–81.
- Rausch, R. T., Goebel, K. F., Eklund, N. H., & Brunell, B. J. (2005). Integrated In-Flight Fault Detection and Accommodation: A Model-Based Study. In *Volume 1: Turbo expo 2005* (pp. 561–569). ASME. doi: 10.1115/GT2005-68300
- Skomedal, Å., Øgaard, M. B., Selj, J., Haug, H., & Marstein, E. S. (2019). General, robust, and scalable methods for string level monitoring in utility scale pv systems. In *36th european photovoltaic solar energy conference and exhibition*.
- Triki-Lahiani, A., Abdelghani, A. B.-B., & Slama-Belkhdja, I. (2018). Fault detection and monitoring systems for photovoltaic installations: A review. *Renewable and Sustainable Energy Reviews*, 82, 2680–2692.
- William F. Holmgren, C. W. H., & Mikofski, M. A. (2018). pvlib python: a python package for modeling solar energy systems. *JOSS The journal of open source software*.
- Zhou, C., & Paffenroth, R. C. (2017). Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (pp. 665–674).