Anomaly detection in PV fleet data via interpretable machine learning

Bennet Meyers
National Renewable Energy Lab
Golden, CO, 80401

Corentin Servouze
Stanford University
Stanford, CA, 94305

Aramis Dufour
Stanford University
Stanford, CA, 94305

***Abstract*—We present a data-driven methodology for inferring anomalies in photovoltaic (PV) power generation signals, based on a set of nearby generators. Our approach models correlations between nearby PV generators over a sliding time window. Our novel, white-box machine learning approach has three important components: (1) a time-dependent, multi-periodic quantile model of the individual power signals, which is used for marginal normalization of observed signals, (2) a linear regression model that predicts the normalized power output of a system at a given time based on a "neighborhood" of measurements across systems and time, and (3) a traditional binary classification algorithm. Model fitting is achieved via convex optimization, which provides globally optimal solutions in polynomial time.**

***Index Terms*—PV systems, anomaly detection, operations and maintenance, convex optimization, machine learning, interpretable models, artificial intelligence**

I. Introduction

The photovoltaic (PV) solar generation capacity in the United States has grown at an average of 22% annually over the last decade and accounted for 53% of new generating capacity in 2023, the first time in 80 years that a renewable energy resource was a majority of capacity addition [1]. Fast and accurate analysis of plant operating conditions from field data is critical for maximizing system energy yield providing a reliable operating capacity to serve the electrical grid.

In this paper, we study the problem of performing (near) real time anomaly detection on a PV fleet, *i.e.*, classifying operational issues in PV systems such as string outages or stuck trackers as they occur (rather than retrospectively in a historical data set). We assume access to time series of power production from a collection of PV generators that have correlated outputs. We present a supervised machine learning method for labeling the daily output of a given PV generator as containing a partial outage, given the daily output of the other systems in the fleet. We test the method on real PV power data from 9 rooftop PV systems in Orange County, CA, introducing known synthetic outages for both training and testing, and we show that it outperforms both a naive baseline and an off-the-shelf machine learning model.

II. Related work

A. PV outage and anomaly detection

There exists significant literature on detecting partial outages in PV system field data, based on both traditional modeling [2], [3] and machine learning approaches [4]–[6]. As described in [7], the "industry standard" solution to anomaly detection in utility power plants (PV and otherwise) is "advanced pattern recognition" (APR) software, which the authors note can be difficult to use effectively and is easily outperformed. These methods operate on a single PV system at a time (as opposed to working with a fleet of power generation signals) and tend to be concerned with classifying different fault categories.

B. Machine learning models for PV fleet data

Deep learning methods have recently been a popular choice for predicting PV generator output based on neighboring systems for purposes of forecasting, imputation, and anomaly detection [8], [9]. These methods employ deep neural networks with large numbers of parameters, which require special computational hardware to train [10]. The fundamental structure of the approaches taken in these papers—train a statistical model to predict a time-series based on neighboring time-series—is similar to the methods proposed in this paper.

C. Quantile transform

Quantile transforms are well known with popular implementations in packages such as sklearn [11]. Our method differs in two important ways: (1) we estimate time-varying quantiles instead of the quantiles of the bulk distribution, and (2) we base the transform on 11 quantile estimates (rather than 100s or 1000s) at the levels $[0.02, 0.1, 0.2, \ldots, 0.8, 0.9, 0.98]$.

D. Our contributions

We present a tightly scoped machine learning pipeline, designed for the task of detecting the partial loss of power production in a PV fleet. This method is comprised of distinct steps that are interpretable and auditable. We emphasize that the calculations carried out for this paper were all performed on standard laptops with no special compute hardware. The most computationally intensive portion of the work is the fitting of the quantile transform described in §III-B, which can take up to 5 or 10 minutes per PV system. However, this process may be carried out in parallel for large numbers of systems, enabling the scaling to very large fleets. The fleet is "coupled" through the linear regression model described in §III-C, for which highly efficient solution methods exist even for very large problems.

III. Method

We assume access to power time series measurements for a collection of $K+1$ PV generators. In this paper, we will describe a method for performing outlier or anomaly detection for one generator, based on the observed measurements from the remaining K generators, which we will refer to as the "target generator" and the "reference generators" respectively. Our proposed method is summarized as follows:

1) transform data to be marginally Gaussian
2) predict target system output from reference systems
3) classify residuals of prediction as containing partial outage or not

We propose a method for learning the model from training data that is expressive but quite interpretable. Our approach to machine learning is based on convex optimization [12], which guarantees globally optimal solutions in polynomial time.

A. Data preprocessing

We begin by on-boarding and cleaning the data with Solar Data Tools [13], [14], which flags days with major operational issues (full outages and similar) and performs basic data filling (linear interpolation during daytime and zero-filling at nighttime), described in more detail in [15]. Days flagged as having operational issue are removed from subsequent model training and testing.

Next, we perform a dynamic time dilation operation which removes nighttime values and standardizes the number of value in each day to be exactly 100, spaced evenly between sunrise and sunset, while maintaining the correct energy content of the signal, as described in [16]. This processes is demonstrated in figure 1. The number of points in the dilated days is an algorithm hyperparameter, and we generally find that a good rule of thumb is to not aggressively down-sample or up-sample the signal. We are using 5-minute data in this study, which has 288 measurements per day, meaning on average daytime covers have of that (more in the summer and less in the winter), or 144 measurements. In practice, we found that the classification performance explored in this research was not very sensitive to this parameter. A subroutine to carry out this dilation is available in Solar Data Tools.

B. Normalizing quantile transform

Nearby PV systems exhibit strong correlation, but they often do not have a simple linear relationship, as demonstrated in figure 2 (left). We note the "leaf shape" in the plot, defined by an outer loop and inner bar. The two systems have different orientations, one more pointed east and one more pointed west. On sunny days, the east-pointing system produces higher power in the morning relative to the west-pointing system, and this relationship is reversed in the afternoon, creating the outer loop. On overcast days, when the irradiance is mostly diffuse, the orientation of the systems is irrelevant, creating the inner bar.

We propose a transformation to the data that explicitly addresses this nonlinear, time-dependent structure. We fit smooth, multiperiodic quantiles to each power signal, as

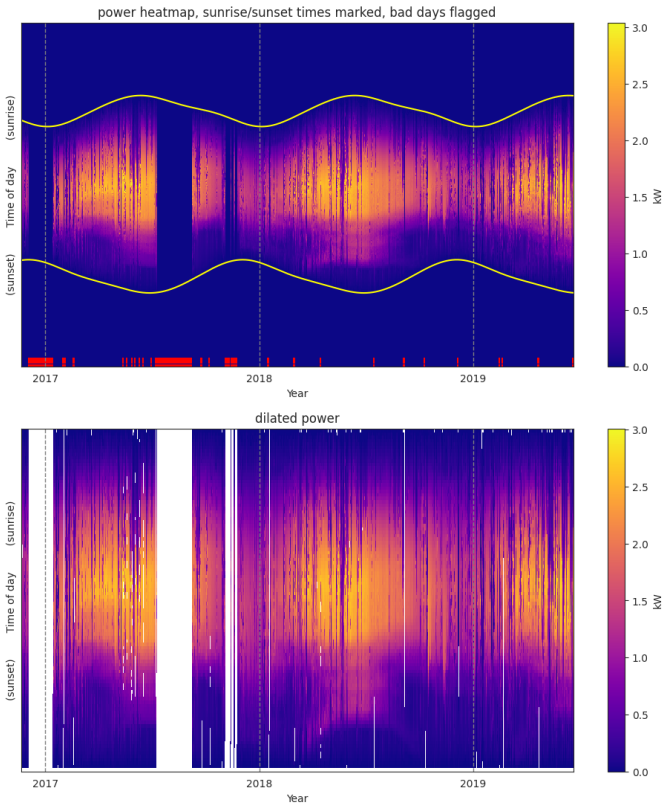

Fig. 1: An example power time-series before (top) and after (bottom) time dilation. The top plot is marked with the sunrise and sunset times estimated automatically with SDT (yellow) and the days flagged by SDT as having operational issues (red).

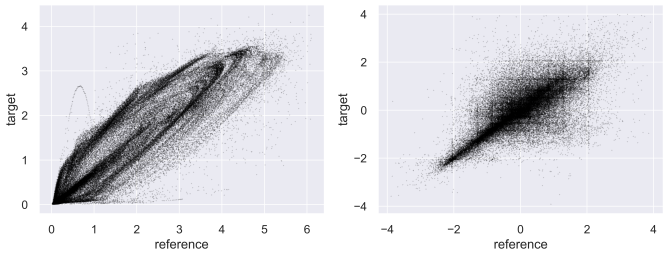

Fig. 2: Scatter plot of target data and data from one reference system before (left) and after (right) quantile normalization.

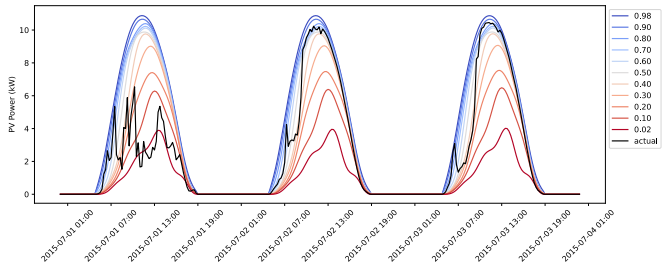

Fig. 3: An example of smooth, multi-periodic quantiles fit to PV power data.

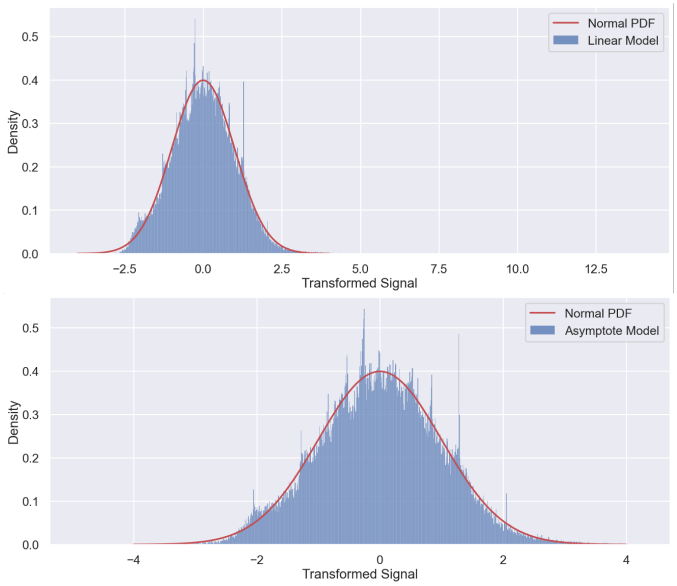

Fig. 4: The bulk distribution of transformed PV power data with linear extrapolation (top) and asymptotic extrapolation (bottom).

described in [16]. Figure 3 shows an example of quantiles fit to PV power time-series data. These quantiles describe the likelihood of observing a certain power output at a particular time on a particular day, *i.e.*, at the 0.4 quantile, there is a 40% probability that the power will be below that level. Note how the quantiles collapse to zero at night when there is no power production.

The transformation function is constructed as a linear interpolation between estimated quantile levels, resulting in a family of transfer functions indexed in time. Note that we only fit 11 quantiles levels, as opposed to 100s or 1000s, as is common for standard quantile transformation implementations. This significantly reduces the computation time (which scales with the number of estimated quantiles) and is justified by numerical experiments that show that increasing the number of quantiles has minimal impact on this problem.

We also need to handle extrapolating the transformation function outside the minimum and maximum fit quantiles, 0.02 and 0.98, respectively (we expect 4% of observed data to fall outside these levels). A natural choice might be linear extrapolation; choosing the slope of the closest internal segment (*i.e.*, the extrapolation below 0.02 would take the slope of the segment between 0.02 and 0.1). However, we find that this is a poor choice in practice for PV power data. We therefore propose an asymptotic extrapolation method. On the lower tail, we use a vertical asymptote at (or just below) zero, representing our prior knowledge that power values are not negative. On the upper tail, we use a horizontal asymptote at a value of 4.26, which corresponds to the quantile function of the normal distribution, evaluated at a quantile level of 0.9999. This represents our prior believe that we do not expect to observe values that have a probability of occurrence of less than 0.01%. In practice, we find that extrapolating with these asymptotic functions results in transformed data with tails that are more Gaussian, as shown in figure 4. The vertical and horizontal asymptotes are implemented by fitting a logarithmic and exponential function respectively. The logarithm barrier function with an asymptote at x_0 has the form

$$f(x) = \beta \log \alpha(x - x_0),$$

and the exponential barrier function with an asymptote at y_0 has the form

$$g(x) = y_0 + \alpha \exp(\beta x).$$

Each function has two free parameters, α and β, which can be solved for analytically by matching the value and slope of the transform function at the boundary quantile. Quantile fitting and data transformation was carried out with the `spcqe` Python package [17].

C. Statistical model

Given power time-series data from $K+1$ PV generators that have been dilated and transformed as described above, we construct features from the K reference systems to predict the standardized power of target system. We propose a linear model that is nonetheless expressive and accurate, while being robust to uncertain training data and fast to fit ($<$1 second on a standard laptop). The features are the measurements from the reference systems, not just at the time of prediction, but over a window of times as well, *i.e.*, leading and lagging values.

Let $y_{d,t} \in \mathbf{R}$ for $t = 1, \ldots, 100$ be the standardized power of the target system at each of the index points on some dilated day, d. Let $x_{d,t} \in \mathbf{R}^n$ be a vector of measurements from our reference systems on the same day. Specifically, we will use the standardized power measurements at index points $\{t{-}3, t{-}2, t{-}1, t, t{+}1, t{+}2, t{+}3\}$—*i.e.*, a window of seven values centered at the time index of interest— from each of the K reference systems, giving $n = 7K$. Our statistical model is

$$y_{d,t} = \theta_t^T x_{d,t} + \epsilon_{d,t},$$

where $\theta_t \in \mathbf{R}^n$ is a vector of coefficients (to be determined by model fitting) and $\epsilon_{d,t} \in \mathbf{R}$ is the residual error. We note that the described features only make sense for $t = 4, 5, \ldots, 96, 97$, so we will only fit $\ell = 94$ linear models. We can compactly represent all ℓ models by defining the following. Let $y_d \in \mathbf{R}^\ell$ be the vector of target values on day d, and let $\Theta \in \mathbf{R}^{\ell \times n}$ be a matrix containing of the coefficients as rows. Finally, let $X_d \in \mathbf{R}^{n \times \ell}$ be a matrix containing the reference features as columns. Then, our model is represented in vectorized form as

$$y_d = \mathbf{diag}\left(\Theta X_d\right) + \epsilon_d,$$

where $\epsilon_d \in \mathbf{R}^\ell$ is a vector of residual errors and $\mathbf{diag}(\cdot)$ is the standard matrix-to-vector diagonal operator, which returns the diagonal entries of a square matrix as a vector.

In this form, we make some observations about the structure of Θ. The rows of this matrix correspond with the coefficients for making a prediction point in time (*i.e.*, at index t). The

columns correspond to the coefficients for each of the K reference systems at a given lag in time relative to the prediction time. The first column corresponds to reference system 1 at a lag of -3 time steps, and the fourth column corresponds to a lag of zero in the same system, *i.e.*, the measured power from that reference at the time of prediction. We expect the model to change slowly over the course of the day because the contribution of a reference at a given lag should not change drastically from, *e.g.*, 10:00AM to 10:05AM. In other words, the columns of Θ should be *smooth*. Motivated by this prior belief and empirical testing, we directly impose a smooth structure on the columns of Θ as follows.

Let $B \in \mathbf{R}^{\ell \times q}$ be a basis matrix of Chebyshev polynomials (of the first kind) of order q (see, *e.g.*, [18]). Then, let $\Theta = B\Phi$, with $\Phi \in \mathbf{R}^{q \times n}$. Our statistical model then becomes

$$y_d = \mathbf{diag}\left(B\Phi X_d\right) + \epsilon_d,$$

with parameters Φ. We select a modest order, such as $q = 8$ and note that $q < \ell$, so this also imposes a *low-rank* structure on Θ in addition to column smoothness, while reducing the total number of parameters to estimate. We find in practice that this constraint increases the out-of-sample prediction accuracy of the model.

D. Model implementation and fitting

We segment the measured power data into training and test sets. With respect to the training set, we solve the ridge regression problem,

$$\text{minimize} \quad \textstyle\sum_d \| \mathbf{diag}\left(B\Phi X_d\right) - y_d\|_2^2 + \lambda \|\Phi\|_F^2, \qquad (1)$$

with variable Φ and where the sum is over all days d in the training set. λ is a weight on the ridge regularization term, which is set through cross-validation. Problem (1) is not only convex [12], but it is a linear least-squares problem and can be solved with standard methods [19]. We used the least squares module from NumPy [20], with solution times on the order of half a second on a standard laptop for a given value of λ.

E. Residual analysis

As a final step, we propose to train a supervised classification model to predict whether a day contains a partial outage based on the residuals between the actual and predicted power of the target systems over the course of the day. This is a classic binary classification problem on a modest ($\mathbf{R}^{96}$) feature space, so many possible off-the-shelf algorithms are available [21], [22]. In §IV we summarize the test performance of many common classifiers.

To formulate this task as a supervised learning problem, we need high-quality labeled data with and without outages. To accomplish this, we construct a partial outage generation model, which applies a random partial outage to the real daily output of a PV system. The process is described in the following.

F. Synthetic partial outage generation

We generate randomized partial outages as follows:

1) With 50/50 probability, select either (a) full-day outage or (b) partial-day outage
2) If partial-day outage, pick start and end times of outage uniformly between sunrise and sunset times
3) Select an outage level with uniform probability between 0 and 100% (no loss and full loss, respectively

We apply this outage generator to every day in the training and test data sets. The binary classifiers, therefore, learn to discriminate between days with a partial outage and those without. More details about the numerical experiments are given in §IV.

G. Baseline models

We also introduce two baseline models to which we compare the performance of our proposed method. The first baseline is the naïve "random predictor," which always always selects `True` or `False` with 50% probability. Our classes are balanced in training and testing, so this is equivalent to random guessing. This model always has an accuracy and f-score of (roughly) 50%; it is a sanity check for our procedure and results. The second baseline is an end-to-end (*e.g.*, "out of the box") XGBoost predictor [23]. This model is trained on the labeled partial outage data, with the features being the actual outputs of all $K+1$ systems. We apply only the *time dilation* portion of our pre-processing pipeline before end-to-end XGBoost training. Our purpose is to test our highly engineered approach to a standard machine learning tool, but we find that removing the night time data and standardizing day length significantly improves statistical models of this kind, and we allow the standard approach to use this "assist". (Note that using the entire 24-hour period forces a statistical model to learn to ignore roughly 50% of the features, when the sun is not shining.)

H. Intraday outage labeling

As a final processing step, we introduce a strategy that aims to identify the specific time periods within a day during with the failure was likely to have occurred. This step was added to anticipate possible operational needs of the solar industry, as such information could be useful for further troubleshooting or categorization of outage types (*i.e.*, extended versus intermittent events). We present this as a "bonus" to the main classification pipeline presented in this manuscript.

This method is based on a Hidden Markov Model (HMM) with two hidden states — failure and no failure. An HMM is a probabilistic model where the latent variable is assumed to follow a Markov chain, and the model emits an observable variable whose distribution depends on the current hidden state [24].

Using training data, we estimate the conditional densities $p(r_t|z_t)$, where r_t is the residual at time t, and z_t is the hidden state. These densities can be estimated either parametrically (using Laplace and Johnson's SU distributions) or non-parametrically (using Gaussian kernel density estimation),

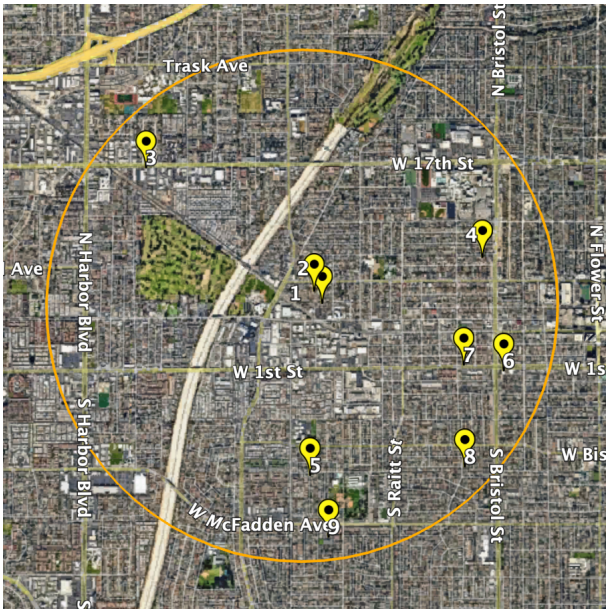

Fig. 5: Relative locations of the 9 residential PV systems selected for this study. A circle is drawn for reference with a radius of 1.25 miles.

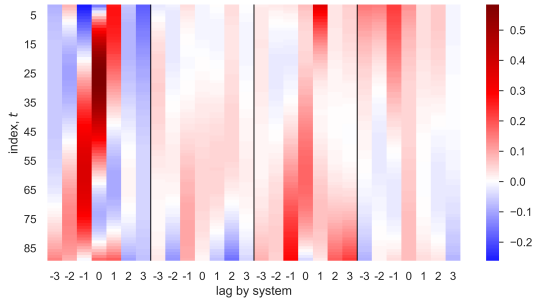

Fig. 6: Estimated Θ for $K = 4$. The coefficients corresponding to each target system are separated by black lines.

with little impact on the final performance of the model. At inference time, we compute the posterior distribution $t \mapsto p(z_t|r_{1:T})$ using the forward-backward algorithm [24, §A.5].

While the HMM can also be employed as the primary binary classifier to determine if an outage occurred on a day, our experiments show that standard machine learning classifiers perform better for this task. Instead, we tune the HMM method to find the most likely periods of an outage *given our prior belief that a partial outage did occur*. So, the HMM acts as a "postprocessor" that takes a prediction of a daily outage and estimates the most likely times for the outage to occur.

IV. Results

We demonstrate the proposed methods on real data from 9 residential PV systems in Southern California within roughly two miles of each other, with overlapping data covering a time period of June 13, 2017 to March 30, 2019, for a total of 655

method	accuracy	f-score	discrimination
ensemble	0.908	0.903	0.815
SVMrbf	0.906	0.900	0.813
PCA+LR	0.897	0.892	0.794
XGB	0.895	0.891	0.791
LR	0.891	0.887	0.781
PCA+QDA	0.876	0.870	0.755
End2EndXGB	0.857	0.842	0.714
PCA+LDA	0.841	0.813	0.683
LDA	0.819	0.797	0.642
QDA	0.772	0.791	0.550
Gaussian z-score	0.729	0.713	0.474
random guess	0.504	0.504	0.253

TABLE I: Table classification methods along with their accuracy and f-score.

consecutive days. A satellite image with the sites marked is shown in figure 5. After cleaning the data and removing days with known operational issues, we split the data sequentially into train and test sets, using an $80/20$ split with 405 days in the train set and 102 days in the test set. (We note that out-of-sample prediction is easier with a random, rather than sequential, hold-out set because of distributional shift. We choose the more difficult configuration.)

A. Linear model ablation study

To evaluate the efficacy of including a "neighborhood" of PV systems in the linear regression model, we perform a quick ablation study, using 1 of the 9 PV systems as the target. We fit three linear models on the test data using the $K = 1, 4$, and 8 closest neighbors. We find the root-mean-square error on the test data to be 0.632, 0.450, and 0.432, respectively, showing increased accuracy on out-of-sample prediction with more reference systems. (Note that after quantile normalization, the data are all mean-zero and unit-variance.) The parameter matrix Θ for the $K = 4$ case is shown in figure 6 as a heatmap over the entries of the matrix. The black lines delineate the grouping of coefficients associated with each of the 4 reference systems. The top rows correspond to morning predictions, and the bottom rows to afternoon predictions. The magnitude of the coefficients tell us the amount of "attention" the model is paying to the different reference systems and lag time. Interestingly, we find that the attention the model pays to the first reference system shifts over the course of the day from a lag of 0 to a lag of -2. Note how the values in each column change smoothly over the course of the day.

B. Classifier performance

We run 45 independent experiments as follows. We iterate over each of the 9 PV systems as the target and train 9 linear regression models on the remaining reference systems over the training data. For each target, we take 5 random samples from the outage generator for each day in the train and test sets. The classifiers are trained 5 times for each target, for a total of 45 trained classifiers, using cross-validation to tune algorithm hyperparameters as needed. Finally, The classifiers are tested on each of the 45 instances of test data, and the accuracy of methods is evaluated over all 45 experiments.

	predicted false	predicted true	*total*
real false	4,400	190	4,590
real true	659	3,931	4,590
total	5,059	4,121	9,180

TABLE II: Partial outage classification confusion matrix for the ensemble model.

	miss positive	detect positive	*total*
miss negative	0	190	190
detect negative	659	3,741	4,400
total	659	3,931	4,590

TABLE III: Partial outage pairwise discrimination results for the ensemble model.

We selected the following classifiers for testing, in increasing order of complexity: Gaussian z-scores, linear discriminate analysis (LDA), quadratic discriminant analysis (QDA), logistic regression (LR), support vector machines with a radial basis function kernel (SVMrbf), and XGBoost (XGB). Additionally, we tried applying principle component analysis (PCA) dimensionality reduction before applying these methods. Finally, we tested a bagged "ensemble" model containing PCA+LR, PCA+QDA, SVMrbf, and XGB, combined using a "soft-voting" procedure [25]. A summary of classifier performance is given in table I. The best performing model is the ensemble, marked in yellow. The two baselines are marked in red (random predictor) and orange (end-to-end XGBoost). In addition to the standard binary classification metrics, *accuracy* and *f-score*, we also include the *discrimination* fraction, *i.e.*, the fraction of outage/no-outage day pairs in which both days are correctly labeled. Because the partial outage data is synthetically constructed from real data with no outages, we can treat the data points as pairs rather than independent samples, and score the methods on their skill at separating these pairs. We find that the random predictor (red) is quite poor, with all proposed models significantly outperforming it. We also note that the end-to-end model (orange), which bypasses the proposed linear regression model and calculation of target residuals, performs reasonably well, outperforming a number of residual classification algorithms. However, XGBoost applied to the residuals outperforms XGBoost when used as an end-to-end model, indicating that the normalization and residual calculation steps in the pipeline are beneficial. For all methods, f-scores are lower than accuracy, and pairwise discrimination scores are lower than f-scores. Additionally, these differences are larger for lower performing methods. For example, the difference in accuracy and discrimination is 0.093 for the ensemble model, while the end-to-end XGBoost has a difference of 0.143. Random guessing only achieves a discrimination score of 0.253, half as much as the accuracy and f-score, a sanity check with basic probability theory (*i.e.*, likelihood of getting heads on two random coin flips).

We give the confusion matrix for the ensemble model in table II, with a total accuracy of 90.8%, a true positive rate of 85.6%, and a true negative rate of 95.9%. Table III summarizes the ability of the ensemble model to discriminate between the pairs of days, noting that the procedure never labels both days in the pair incorrectly. Figure 7 presents the distributions of the true positives and false negatives by *total loss*, which we define to be the loss fraction multiplied by the duration fraction. From this chart, we infer that the threshold of detection for the proposed method is roughly 5% total loss, below which we do worse than random chance. Figure 8 is a heatmap showing the fraction of actual positive outages detected by the proposed method, binned by the loss fraction and duration fraction (*i.e.*, of daylight hours). Blue colors show performance better than random chance, and red colors are worse than random chance. The region where the

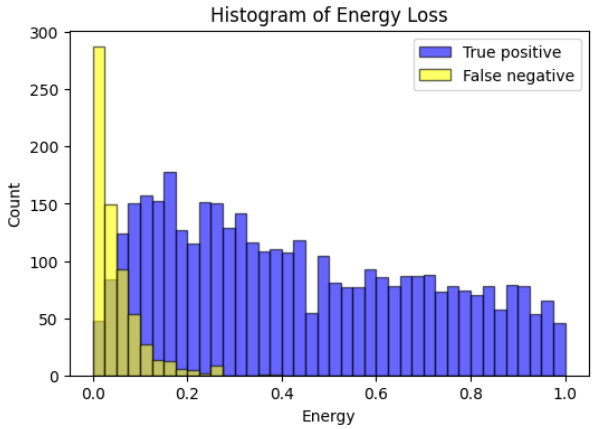

Fig. 7: The distributions of true positives and false negatives by *total loss*, *i.e.*, loss fraction times duration fraction. Each bin has a width of 0.025.

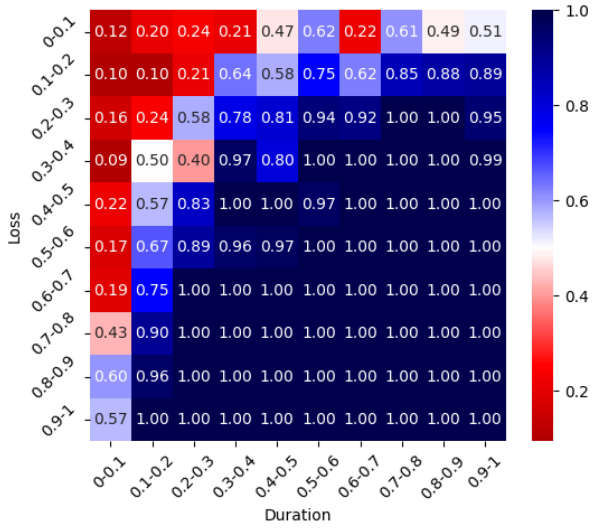

Fig. 8: A heatmap showing the fraction of correct labels on the partial outage test data, binned by loss fraction and duration fraction (of daylight hours).

Fig. 9: An easy pair of days to discriminate, 82.2% outage over 51.5% of the day.

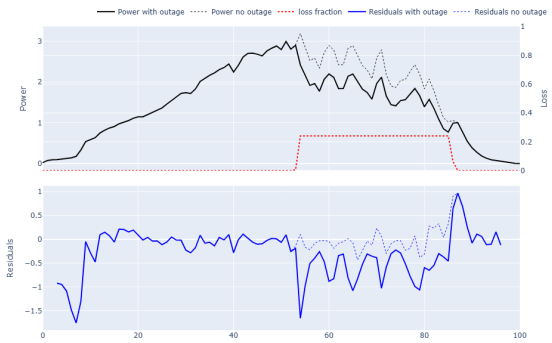

Fig. 10: A more difficult pair of days to discriminate, 24.1% loss over 34.7% of the day.

proposed method performs well has a curved boundary in loss and duration, showing a trade-off in detectability; very low losses are detectable when they have long durations, while short duration outages are detectable when the loss is large. When the loss fraction and duration fraction are both higher than 40% ($> 20\%$ total loss), the proposed method detects the outage with nearly perfect accuracy.

C. Case studies

In figures 9 and 10, we show two pairs of days from the test set that were correctly discriminated by the algorithm, *i.e.*, the "no outage" and "outage" conditions were correctly labeled. The first example is well above our threshold of detection, and most predictors correctly discriminated this pair. There is a large negative deviation in the residuals, with a large run of values around -4, and a human operator would have little difficulty identifying this partial outage condition directly from the power data. The second example is from the frontier of the region of detectability shown in figure 8; the bin has an accuracy of 97% but has neighbors that are below 50%. We consider this to be a more difficult example, that would challenge a human operator performing a visual inspection.

From tables II and II, we observe that the proposed method produces about 3.5 times as many false negatives as false positives; the algorithm is more likely to miss a small partial

Fig. 11: A example from the set of 190 false positive labels, incorrectly predicted to have a partial outage.

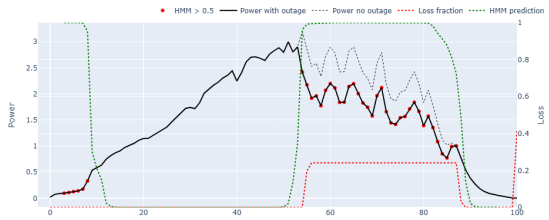

Fig. 12: Sub-daily labeling of partial outage times.

outage than falsely claim an outage that is not present. An example of a false positive is shown in figure 11. The roughly 4% of the "no outage" training days that were incorrectly labeled positive are typically intermittent cloud conditions with large power swings. Inspecting the predicted residuals, we note that they are similar to the expected residuals for short, low loss outage events. We conclude, therefore, that the proposed method could be improved by eliminating from the training data set short, low-loss events that are outside the region of detectability shown in figure 8.

D. Sub-daily labeling

Finally, we demonstrate the application of the HMM discussed in §III-H. Returning to the example shown in figure 10, the estimated time frame of the outage is shown figure 12. We mark the time periods with a red circle if the HMM probability of being in the partial outage state is over 50%. We observe that the method correctly identifies the synthetic outage in the afternoon, but the method also marks a partial outage in the morning, which was not part of the synthetic outage generation, which we interpret as a false positive. (It is worth noting, however, that during research and development, the proposed methods found real partial outages in the training data that were not previously identified, which were then manually filtered in later experiments. In other words, it is possible that there are real—*i.e.*, not synthetic—partial outages in the training data that were not added by us.)

V. Conclusion and Next Steps

We have presented a data-driven method for automatically detecting partial outages in PV systems, based on the output of

a collection of nearby generators. This approach is built from interpretable steps—a data-driven marginal transformation, a linear regression model, and classification of residuals. It nonetheless adapts to complex site conditions such as local shading and encodes rich information about the similarities between nearby systems, which dynamically change over the course of a day. We find that the proposed methods performs very well on a synthetic partial outage data set, outperforming an off-the-shelf implementation of XGBoost.

What we have presented here is a prototype method, that will require additional research and development to operationalize for industry. We suspect the methods described here can be improved through eliminating short-duration, low-loss outages from the training data set. Future work may also include exploring the effect of training set size on classifier performance and evaluating the efficacy of the proposed methods with a much shorter training period, on the order of weeks or months, *i.e.*, looking at the "cold start" problem.

We plan to make the training and test data curated for this work public and available to other researchers. Please reach out to the lead author if you are interested.

Acknowledgment

This work was authored in part by the National Renewable Energy Laboratory for the U.S. Department of Energy (DOE), operated under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number 38529. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

[1] Solar Energy Industries Association (SEIA), "Solar industry research data," 2024. [Online]. Available: https://seia.org/research-resources/solar-industry-research-data/

[2] A. Livera, M. Theristis, L. Micheli, J. S. Stein, and G. E. Georghiou, "Failure diagnosis and trend-based performance losses routines for the detection and classification of incidents in large-scale photovoltaic systems," *Progress in Photovoltaics: Research and Applications*, vol. 30, pp. 921–937, 8 2022.

[3] S. Sheppard, T. Cook, D. Fregosi, C. Perullo, and M. Bolen, "Field experience detecting PV underperformance in real time using existing instrumentation," in *2022 IEEE 49th Photovoltaics Specialists Conference (PVSC)*, vol. 2022-June. IEEE, 6 2022, pp. 0307–0313.

[4] Y. Zhao, B. Lehman, R. Ball, and J.-F. de Palma, "Graph-based semi-supervised learning for fault detection and classification in solar photovoltaic arrays," *2015 IEEE Energy Conversion Congress and Exposition*, vol. 30, pp. 1628–1634, 2015.

[5] Y. Zhao, Q. Liu, D. Li, D. Kang, Q. Lv, and L. Shang, "Hierarchical anomaly detection and multimodal classification in large-scale photovoltaic systems," *IEEE Transactions on Sustainable Energy*, vol. 10, pp. 1351–1361, 7 2019.

[6] F. Aziz, A. U. Haq, S. Ahmad, Y. Mahmoud, M. Jalal, and U. Ali, "A novel convolutional neural network-based approach for fault classification in photovoltaic arrays," *IEEE Access*, vol. 8, pp. 41 889–41 904, 2020.

[7] S. Sheppard, K. A. Dickey, S. Koskey, C. Teasley, C. Perullo, D. Fregosi, and W. Li, "Benchmarking a physics-based approach for anomaly detection at utility pv plants," in *2024 IEEE 52nd Photovoltaic Specialist Conference (PVSC)*. IEEE, 6 2024, pp. 0856–0858.

[8] A. M. Karimi, Y. Wu, M. Koyuturk, and R. H. French, "Spatiotemporal graph neural network for performance prediction of photovoltaic power systems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 15 323–15 330, 5 2021.

[9] R. Wieser, Y. Fan, X. Yu, J. Braid, A. Shaton, A. Hoffman, B. Spurgeon, D. Gibbons, L. S. Bruckman, Y. Wu, and R. H. French, "Large scale, data driven, digital twin models: Outlier detection and imputation," in *2024 IEEE 52nd Photovoltaic Specialist Conference (PVSC)*. IEEE, 6 2024, pp. 0902–0905.

[10] D. G. Widder, S. West, and M. Whittaker, "Open (for business): Big tech, concentrated power, and the political economy of open AI," *SSRN Electronic Journal*, 8 2023.

[11] The scikit-learn developers, "Sklearn quantile transformer," Jan. 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html

[12] S. Boyd and L. Vandenberghe, *Convex optimization*, 9th ed. Cambridge University Press, 2009.

[13] S. Miskovich and B. Meyers, "Solar data tools documentation," Jan. 2025. [Online]. Available: https://solar-data-tools.readthedocs.io

[14] B. Meyers, E. Apostolaki-Iosifidou, and L. Schelhas, "Solar data tools: Automatic solar data processing pipeline," in *2020 47th IEEE Photovoltaic Specialists Conference (PVSC)*, 2020, pp. 0655–0656.

[15] B. Meyers, "PVinsight (final technical report)," SLAC National Accelerator Laboratory (SLAC), Menlo Park, CA, Tech. Rep., 2021, SLAC-R-1155. [Online]. Available: https://doi.org/10.2172/1897181

[16] G. Ogut, B. Meyers, A. Dufour, and S. Boyd, "Time dilated Bundt cake analysis of PV output," in *2024 IEEE 52nd Photovoltaic Specialist Conference (PVSC)*. IEEE, 6 2024, pp. 877–883. [Online]. Available: https://ieeexplore.ieee.org/document/10749393/

[17] B. Meyers, S. Miskovich, A. Dufour, and G. Ogut, "spcqe python package," Jan. 2025. [Online]. Available: https://github.com/cvxgrp/spcqe

[18] E. W. Weisstein, "Chebyshev polynomial of the first kind," *MathWorld–A Wolfram Web Resource*. [Online]. Available: https://mathworld.wolfram.com/ChebyshevPolynomialoftheFirstKind.html

[19] S. Boyd and L. Vandenberghe, *Introduction to Applied Linear Algebra*. Cambridge University Press, 2018.

[20] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer New York, 2009.

[22] The scikit-learn developers, "Classifier comparison," *scikit-learn documentation*, 2025. [Online]. Available: https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785

[24] J. Daniel and J. H. Martin, "Hidden markov models," in *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 2025. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/

[25] The scikit-learn developers, "Sklearn soft voting classifier," Jan. 2025. [Online]. Available: https://scikit-learn.org/stable/modules/ensemble.html#weighted-average-probabilities-soft-voting