Performance Comparison of Clipping Detection Techniques in AC Power Time Series

Preprint

Kirsten Perry, Matthew Muller, and Kevin Anderson

National Renewable Energy Laboratory

Presented at the 48th IEEE Photovoltaic Specialists Conference (PVSC 48)
June 20-25, 2021

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-5K00-78954
October 2021

Performance Comparison of Clipping Detection Techniques in AC Power Time Series

Preprint

Kirsten Perry, Matthew Muller, and Kevin Anderson

National Renewable Energy Laboratory

Suggested Citation
Perry, Kirsten, Matthew Muller, and Kevin Anderson. 2021. *Performance Comparison of Clipping Detection Techniques in AC Power Time Series: Preprint*. Golden, CO: National Renewable Energy Laboratory. NREL/CP-5K00-78954.
https://www.nrel.gov/docs/fy22osti/78954.pdf.

NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Contract No. DE-AC36-08GO28308

Conference Paper
NREL/CP-5K00-78954
October 2021

National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, CO 80401
303-275-3000 • www.nrel.gov

NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) 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 Solar Energy Technologies Office (SETO) Agreement Numbers 34348.

Cover Photos by Dennis Schroeder: (clockwise, left to right) NREL 51934, NREL 45897, NREL 42160, NREL 45891, NREL 48097, NREL 46526.

NREL prints on paper that contains recycled content.

Performance Comparison of Clipping Detection Techniques in AC Power Time Series

Kirsten Perry[1], Matthew Muller[1], Kevin Anderson[1]
[1] NREL, Golden, CO, USA

***Abstract*—In this research, a variety of methods were developed to detect clipping periods in AC power time series. AC power data streams associated with 36 unique systems across the United States were collected, and data points representing clipping periods were manually labeled by experts. Using this data set for training and validation, novel logic-based and machine learning (ML) approaches were developed to classify time series values as clipping or non-clipping. These approaches were compared to the RdTools method for detecting clipping periods. The logic-based and ML XGBoost approaches achieved F-scores of 85.0 and 77.6, respectively, when cross-validated against the manually labeled data, as compared to the current RdTools approach (F-score of 56.4), indicating a significant improvement at detecting clipping periods. Additionally, the effects of each clipping filter when evaluating system degradation rates were assessed, using 31 unique systems across the United States. Results indicate that estimated system degradation rate can vary based on the type of clipping filter used, by up to 0.6% degradation rate for some cases.**

***Index Terms*—machine learning, clipping, photovoltaic, solar, modeling, rdtools**

I. Introduction

Due to a significant decrease in solar module prices over the past decade, developers have increased installations' inverter loading ratio (ILR), resulting in a higher DC-to-AC ratio. By increasing a system's DC-to-AC ratio, developers can increase system output outside of peak irradiance windows. Consequently, this design decision has led to an increase in clipping in solar projects. Clipping prevents inverter overload by operating the array in a reduced efficiency state when normal operation would exceed the inverter's power conversion limits. Generally, clipping signals manifest as a flat line at or near the peak of an AC inverter's production capacity. Clipping thresholds can remain constant for an AC power time series, or may vary. Some causes of variance include temperature derating and dynamic plant control. Some example clipping profiles, taken from physical AC power streams, are shown in Fig. 1.

Best practices for PV degradation analysis include the removal of clipped points from AC power or energy time series [1]. A few approaches have been explored to mask clipping periods in AC inverter time series, which are described presently.

One approach, given in the RdTools Python package [2], is to filter power or energy data using a percentile cutoff [1]. The default clipping filter in the RdTools package is set to the 98th quantile, where any data in the AC power time series greater than 99% of the 98th quantile is assumed as clipping, and filtered out of the time series before running a degradation analysis. Although this approach is simple and intuitive, it fails to handle edge cases where the clipping cutoff threshold varies over the course of the time series. Furthermore, this approach filters out time series data during periods of peak irradiance, where un-clipped systems may be performing optimally.

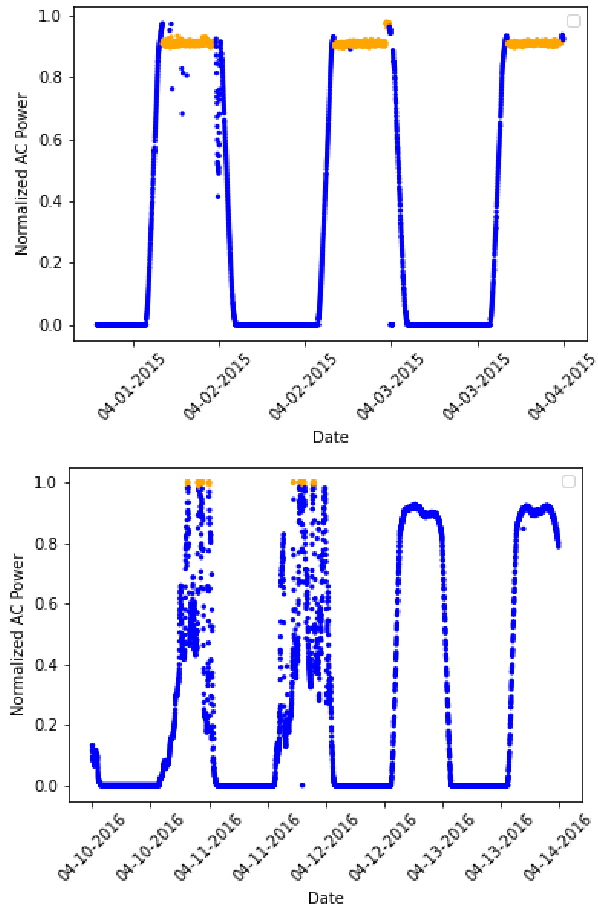

Fig. 1. Example AC power time series data, taken from actual field PV systems. Clipping periods are labeled in yellow and non-clipping periods are labeled in blue.

Another approach developed for clipping detection uses fuzzy system logic [3], where features such as actual plant power, the time series gradient mean, and the time series gradient range are taken as inputs to evaluate the degree of clipping in a solar plant. This method was validated using labeled data from a PV plant located in Spain, and was approximately 90% accurate. This research only explored algorithm performance at a single location; in contrast, our work handles multiple configurations, data frequencies, and locations, so it is more universally applicable when applied to field data.

In this paper, we present a series of logic-based and ML clipping filters and compare their performance to the RdTools clipping filter performance using manually labeled and simulated clipping validation data. We plan to publicly release these algorithms via the RdTools package, as well as the manually labeled data that we used during the validation process.

Additionally, we assess the effects of applying each clipping filter when performing system degradation analysis, using physical PV system data. By doing this, we demonstrate that degradation estimates vary by clipping filter used.

II. Methodology

A. Datasets

Thirty-six AC power time series, representing 31 physical PV systems and approximately 2.13 million rows of data, were selected with the intent of building a training and validation set. Data with a wide range of clipping behaviors was selected, including varying clipping signal behavior (noisy vs. clean signals), averaging interval (ranging from 1-minute to hourly), and mounting configuration (single-axis tracking vs. fixed tilt). Estimated clipping periods for each time series were manually labeled by two experts with experience identifying clipping signals in data. The datasets used for this study were provided via the National Renewable Energy Laboratory's PV Fleet Performance Data Initiative [4]. A summary of the manually labeled data set distribution by data averaging interval and mounting configuration is provided in Tables I and II, respectively. Predominantly, data streams are averaged at 1-minute and 15-minute intervals, with one data stream averaged at a 30-minute interval and two data streams averaged at a 60-minute interval. Fifteen of the data streams are associated with single-axis tracking systems, and twenty-one of the data streams are associated with fixed tilt systems.

A subset of the manually labeled data was selected and artificial clipping signals were inserted into these time series. This simulated clipping data set was generated to test algorithm performance during ideal clipping situations, where all data are clipped at a constant AC power threshold. Two different approaches were taken for inserting artificial clipping signals. First, some data sets were down-sampled at different frequencies, and time periods labeled as 100 percent clipping were marked as clipping periods. For the second approach, clipping at a variable threshold above the 70th percentile was inserted in the time series, by setting all values greater than or equal to the threshold as the associated threshold value. Each time series could have up to 5 different clipping thresholds across the series, with location and number of thresholds randomly selected. A summary of the simulated clipping data, broken down by averaging interval and mounting configuration, is provided in Tables III and IV, respectively. Overall, the simulated clipping data set consists of twenty-three individual test cases, with the vast majority of the data sets averaged at 1-minute, 15-minute, and 30-minute intervals. Seventeen of the test cases are associated with a fixed-tilt mounting configuration, and six of the test cases are associated with a single-axis tracking mounting configuration.

TABLE I
Distribution of Manually labeled Data, by Averaging Interval

Frequency (Minutes)	Number Data Streams	Number Readings
1	15	951387
15	18	1048464
30	1	64232
60	2	61364

TABLE II
Distribution of Manually labeled Data, by Mounting Configuration

Mounting Configuration	Number Data Streams	Number Readings
Fixed Tilt	21	1237417
Single-Axis Tracking	15	888030

B. RdTools Approach

Each AC power time series described in the Datasets subsection was run through the RdTools clipping filter. This filter calculates the 98th quantile of the AC power time series, and any values that are greater than 99% of the 98th quantile are classified as clipping.

Any clipping periods identified by the RdTools clipping filter were compared to the labeled training sets, to quantify filter performance. Filter accuracy, precision, recall, and F-score were calculated using the following three equations, respectively:

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$\text{precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F-score} = \frac{2 \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}} \tag{4}$$

where TP is the number of true positives, FP is the number of false positives, FN is the number of false negatives, and TN is the number of true negatives. Accuracy represents the number of correctly classified data points divided by total data points. Recall, or true positive rate (TPR), is the ability to correctly detect positive cases. Precision, or positive prediction value (PPV), is the fraction of correctly identified positives given the total positive predictions. F-score is the harmonic mean of precision and recall. The higher the F-score, the more precise and robust the classifier.

TABLE III
DISTRIBUTION OF DATA WITH SIMULATED CLIPPING PERIODS, BY AVERAGING INTERVAL

Frequency (Minutes)	Number Data Streams	Number Readings
1	5	484783
5	1	199993
15	9	390142
30	6	203501
60	2	61322

TABLE IV
DISTRIBUTION OF DATA WITH SIMULATED CLIPPING PERIODS, BY MOUNTING CONFIGURATION

Mounting Configuration	Number Data Streams	Number Readings
Fixed Tilt	17	1162365
Single-Axis Tracking	6	177376

C. Logic-Based Approach

A new logic-based filter was developed for detecting clipping periods in time series. This filter identifies clipping periods via the following steps:

1) The averaging interval of the power or energy time series is determined. If the averaging interval is more frequent than once every 10 minutes, then the time series is aggregated to 15-minute intervals, using the mean.
2) The maximum and minimum readings are calculated across a moving window for the time series. For tracked systems with a data averaging interval more frequent than once every 30 minutes, a window of 5 readings is used. For all other systems, a window of 3 readings is used.
   Once the maximum and minimum moving values are determined, the following equation is used to calculate the maximum rolling range of each value:
   $$range_{max} = \frac{x_{rolling\,max} - x_{rolling\,min}}{\frac{x_{rolling\,max} + x_{rolling\,min}}{2}} * 100 \quad (5)$$
   where $x_{\text{rolling max}}$ is the rolling maximum over the past n values, and $x_{\text{rolling min}}$ is the rolling minimum over the past n values.
3) An initial boolean clipping mask is derived, where values with a max rolling range (see (5)) of less than 0.2 are marked as clipping. When an individual data point is determined as clipped per (5), all data points within the given p-length rolling window are also set as clipped, where p is 3 or 5 reading, as described in Step 2. The value 0.2 is an empirically-derived cutoff.
4) Any high-frequency time series that were aggregated up to 15-minute intervals are sampled back to their original frequency. Clipping labels are forward-filled in the data.
5) Additional logic is added to address noise in high frequency time series sets. For time series data sets with a averaging interval of 10 minutes or more frequent, the daily mean and standard deviation for labeled clipping values is determined. A maximum and minimum daily threshold for setting clipping values is derived as 2 standard deviations above and below the mean, respectively. For time series data sets with a averaging interval less frequent than every 10 minutes, the daily maximum and minimum thresholds for clipping are set as the maximum and minimum values where clipping is detected over the course of that day, respectively.
6) Once the daily maximum and minimum clipping thresholds are determined, all data values between these two thresholds are updated as clipping periods.
7) An overall clipping threshold is applied to the data, based on the classified clipping periods derived in the previous steps. The following is calculated:
   $$threshold_{clipping} = \frac{x_{p99} - clippedx_{p99}}{\frac{x_{p99} - clippedx_{p99}}{2}} \quad (6)$$
   In addition to the previously identified clipping points, all values in the time series greater than the overall clipping threshold are classified as clipping.
8) Accuracy, precision, recall, and F-score, described in (1)-(4), are calculated, where the logic-based model predictions are compared to the ground truth labels in the training and validation data set.

D. Supervised Machine Learning Approach

The labeled time series data described in the Datasets section are used to train a series of black-box ML models to classify clipping periods. For each individual data stream, the data is min-max normalized, via the following equation:

$$x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (7)$$

By performing min-max normalization, data streams with different units and clipping thresholds can be compared simultaneously. Features are then derived from the min-max normalized data. Features used in the model include:

- The min-max normalized AC power time series
- The simple moving average of the time series, where a 5 reading-window is used when the averaging interval is more frequent than once every 10 minutes, a 3 reading-window is used when the averaging interval is between 10 minutes and an hour, and 2-reading window is used otherwise
- The forward- and backward-calculated derivatives for the min-max normalized values and the rolling average
- The maximum rolling range of the min-max normalized values, as shown in Equation 5
- The mounting configuration (fixed tilt vs. single axis tracking), converted to a categorical variable
- The mode, or most common, time series sampling frequency
- The daily maximum of the time series, as well as the percent of the daily max each individual reading is
- After generating classifier predictions for a data set, additional override logic is introduced. Specifically, all

values between the maximum and minimum predicted clipping periods for each day are set as clipping, where predicted clipping periods must be 80 percent or greater of the associated daily maximum value.

A random forest classifier model and an extreme gradient boosting, or XGBoost, classifier model were used to train the model. Python's open-source scikit-learn [5] package was used to build the random forest model, and the open-source XGBoost package [6] was used to build the XGBoost model.

Cross-validation was used when evaluating model performance. For each individual data stream, the model was retrained with the data stream omitted from the training set, and model performance was evaluated on the omitted data stream. In all, the model was regenerated 59 times, for each of the 59 data streams (manually labeled and simulated data). After the model had generated clipping classifications for each data stream, the overall accuracy, precision, recall, and F-score were calculated.

E. Applying Algorithms to Field Data

The degradation rates of 31 different systems in the PV Fleets Initiative were calculated using the RdTools sensor-based methodology [4]. Systems comprised a series of different solar fleet owners, as well as diverse geographic locations, sampling frequencies, and system configurations. When performing the degradation analysis, all parameters remained constant, except the clipping filter used on the data. Five different experiments were run to assess the role the clipping filter has on degradation results: no clipping filter applied to the data (the placebo); the RdTools filter applied; the logic-based algorithm applied; the Random Forest algorithm applied; and the XGBoost algorithm applied. Once the RdTools sensor-based degradation methodology was run on the systems, the results for each experiment were compared.

III. Results

A. Algorithm Performance on the Training and Validation Set

Table V displays the accuracy, precision, recall, and F-score for each of the different clipping algorithms, based on performance on the manually-labeled data set and simulated clipping data set, respectively.

Using F-score as a benchmark since it accounts for both classifier precision and robustness, the best overall performance on the manually labeled data set was achieved by the logic-based clipping filter, followed by the Random Forest and XGBoost clipping filters, respectively. All ML and logic-based filters performed better on the manually labeled data set, compared to the standard RdTools clipping filter (F-score of 56.4). For the simulated data set, the Random Forest classifier algorithm had the best overall performance with an F-score of 93.6, followed by the RdTools filter (F-score of 92.8) and the logic-based filter (F-score of 92.4), respectively. It is unsurprising that the RdTools filter scores significantly better on simulated data than on physical data. The simulated data sets focus largely on ideal conditions where clipping occurs at a consistent threshold at the time series maximum, which the

TABLE V
Clipping Filter Performance on labeled Clipping Data

Label	Algorithm	Accuracy(%)	Recall	Precision	F-Score
Manual	RdTools	92.1	46.1	72.6	56.4
Manual	Logic-based	96.5	89.4	81.0	85.0
Manual	Random Forest	95.2	79.4	77.6	78.5
Manual	XGBoost	94.8	81.4	74.1	77.6
Simulated	RdTools	98.0	89.1	96.9	92.8
Simulated	Logic-based	98.1	95.6	89.5	92.4
Simulated	Random Forest	98.5	90.9	96.5	93.6
Simulated	XGBoost	97.3	81.8	95.4	88.1

RdTools filter is constructed specifically to detect. However, under field conditions, clipping signals do not necessarily manifest this ideal behavior, as the clipping threshold may vary over time and include noise. All of the algorithms perform better on the simulated data than the manually labeled data, likely because simulated data represents idealized clipping behavior, in opposition to the real-world behavior present in the manually labeled data set.

Table VI illustrates algorithm performance by mounting configuration (fixed tilt vs. single-axis tracking), and Table VII illustrates algorithm performance by data averaging interval, for each of the data sets. In examining the results from Table VI, algorithms generally perform similarly for fixed-tilt systems and single-axis tracking systems, when F-score is used as a benchmark. One notable exception is the RdTools filter, which performs significantly better on fixed tilt data set compared to single-axis tracking data set, for the manually labeled data (an F-score of 67.7 vs. an F-score of 51.6, respectively). AC power data for a tracking system manifests as a valley-like signal around maximum daily output, as shown in Figure 2. Although this signal is distinct from a clipping signal, it is likely to be confounded as a clipping signal using the RdTools' filter's simple logic.

Algorithm performance varied across averaging interval for the manually labeled data set. For 1-minute, 15-minute, and 60-minute manually labeled data, the logic-based filter had the best performance, with F-scores of 81.1, 90.1, and 90.3, respectively. The random forest algorithm had the best performance on the 30-minute manually labeled data, with an F-score of 90.6.

B. Degradation Variability for Physical Systems, based on Clipping Algorithm Used

In total, 31 systems, representing multiple fleet owners across the NREL PV Fleets initiative, were evaluated using the RdTools methodology to determine sensor-based degradation rates [4]. Systems represented a wide geographic area across the United States, with multiple configurations, module technologies, and data sampling frequencies. Particularly, we focused on how estimated degradation rates varied for each system, based on which clipping filter was used. Fig. 3 shows the difference in sensor-based degradation rates, where the results that used the logic-based, XGBoost, and RdTools filters are subtracted from the results where no filter was used, for

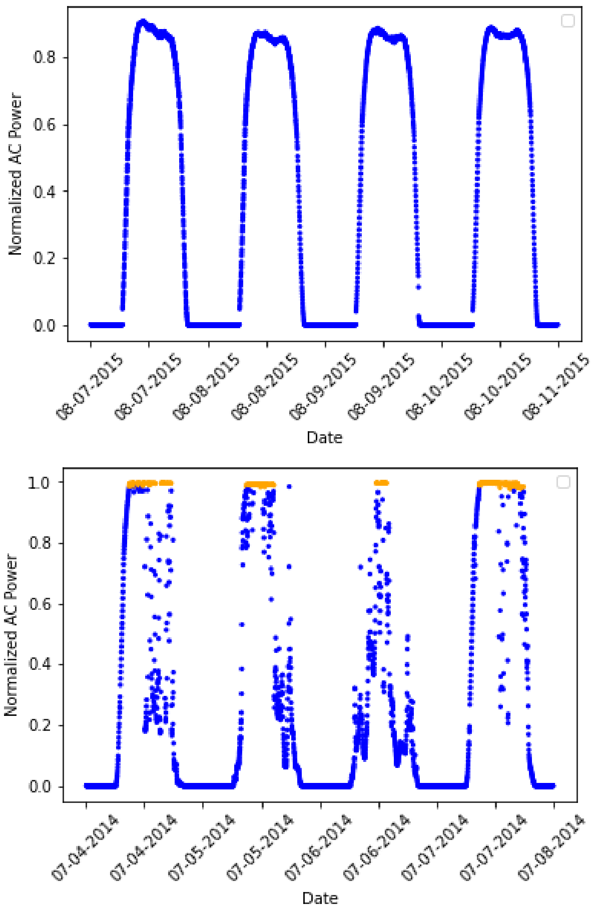

Fig. 2. Tracking system behavior vs. fixed tilt system behavior, respectively, where clipping periods are labeled in yellow and non-clipping periods are labeled in blue.

each respective system. Data is color-coded by system, and system degradation rate is calculated as the mean degradation across all AC power inverters associated with the system. Based on these results, the type of filter affects degradation estimates; in some systems, this can be up to 0.6%/year.

Figure 4 displays the individual distributions of the clipping filter variation, where the system degradation rate with each respective filter applied is subtracted from the system degradation rate when no filter is applied. The median difference in degradation estimates for the XGBoost, Random Forest, Logic-Based, and RdTools filters was 0.05%, 0.05%, 0.04%, and 0.06%, respectively. Likewise, the standard deviation of the differences for the XGBoost, Random Forest, Logic-Based, and RdTools filters was 0.11%, 0.12%, 0.10%, and 0.20%, respectively. These results indicate that the degradation distributions for the XGBoost, Random Forest, and logic-based filters are similar, with a tighter distribution than the RdTools degradation results.

IV. Future Work

In addition to performing degradation analysis on physical systems, we plan to the same process on simulated data with known degradation rates, to evaluate degradation accuracy when each of the four treatment types was used. By doing this, we can further demonstrate that improving clipping filtering on AC power data streams results in more accurate degradation estimates for a system.

TABLE VI
Clipping Filter Performance By Mounting Configuration

Labeling	Mounting	Algorithm	Accuracy (%)	F-Score
Manual	Fixed Tilt	RdTools	97.0	67.7
Manual	Fixed Tilt	Logic-based	98.3	82.2
Manual	Fixed Tilt	Random Forest	97.9	76.6
Manual	Fixed Tilt	XGBoost	97.8	76.9
Manual	Single-Axis Tracking	RdTools	85.3	51.6
Manual	Single-Axis Tracking	Logic-based	93.7	83.1
Manual	Single-Axis Tracking	Random Forest	91.5	79.1
Manual	Single-Axis Tracking	XGBoost	90.6	77.9
Simulated	Fixed Tilt	RdTools	98.1	92.6
Simulated	Fixed Tilt	Logic-based	98.1	90.8
Simulated	Fixed Tilt	Random Forest	98.6	92.2
Simulated	Fixed Tilt	XGBoost	97.2	83.7
Simulated	Single-Axis Tracking	RdTools	96.6	94.0
Simulated	Single-Axis Tracking	Logic-based	98.2	96.6
Simulated	Single-Axis Tracking	Random Forest	98.2	96.7
Simulated	Single-Axis Tracking	XGBoost	98.2	96.8

TABLE VII
Clipping Filter Performance By Averaging Interval

Labeling	Frequency (min)	Algorithm	Accuracy (%)	F-Score
Manual	1	RdTools	85.3	49.8
Manual	1	Logic-based	92.9	81.1
Manual	1	Random Forest	90.6	76.7
Manual	1	XGBoost	89.7	75.5
Manual	15	RdTools	97.8	76.4
Manual	15	Logic-based	99.3	90.1
Manual	15	Random Forest	98.9	85.8
Manual	15	XGBoost	99.0	86.8
Manual	30	RdTools	98.7	63.2
Manual	30	Logic-based	99.4	73.1
Manual	30	Random Forest	99.8	90.6
Manual	30	XGBoost	99.7	89.6
Manual	60	RdTools	94.0	43.5
Manual	60	Logic-based	98.7	90.3
Manual	60	Random Forest	98.5	88.4
Manual	60	XGBoost	98.3	87.2
Simulated	1	RdTools	97.5	93.2
Simulated	1	Logic-based	96.8	92.0
Simulated	1	Random Forest	98.8	96.9
Simulated	1	XGBoost	96.1	89.7
Simulated	15	RdTools	97.1	88.7
Simulated	15	Logic-based	99.5	98.2
Simulated	15	Random Forest	96.7	81.7
Simulated	15	XGBoost	96.0	77.1
Simulated	30	RdTools	99.3	96.8
Simulated	30	Logic-based	99.7	98.5
Simulated	30	Random Forest	99.7	94.7
Simulated	30	XGBoost	99.5	91.6
Simulated	60	RdTools	91.3	76.5
Simulated	60	Logic-based	99.2	98.1
Simulated	60	Random Forest	99.5	98.8
Simulated	60	XGBoost	99.4	98.6

V. Conclusions

This research presents a series of new filters that can be used to successfully detect clipping periods in AC power time series data from solar installations. These filters, which will be released in the RdTools Python package, allow solar analysts

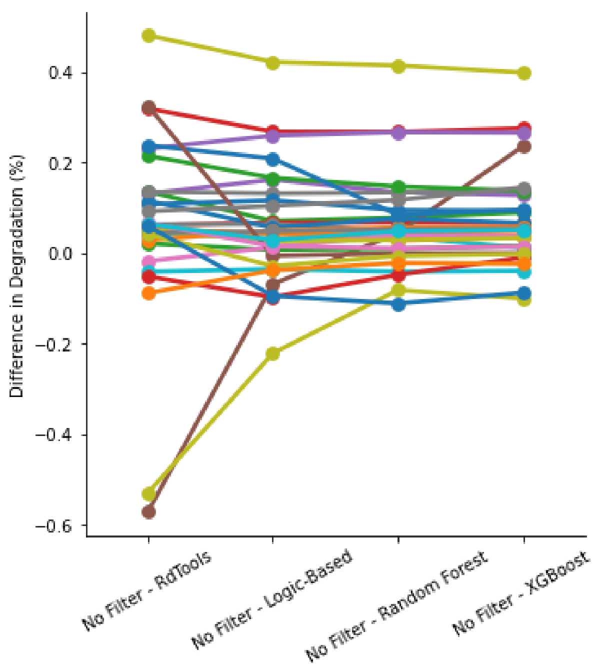

Fig. 3. Scatterplot of the difference in sensor-based degradation rates when no filter is used vs. when each respective filter is used, color-coded by individual system

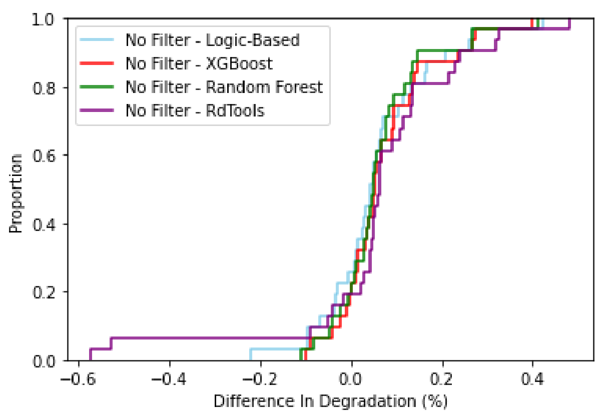

Fig. 4. CDF of the difference in degradation rates when no filter is used vs. when each respective filter is used

to automatically and consistently identify and remove clipping periods from their time series data. All filters developed in this research show a significant improvement at detecting individual clipping periods in time series, when compared to the current RdTools method, with the logic-based filter having the best performance overall (F-score of 85.0 on manually labeled data vs. RdTools F-score of 56.4 on manually labeled data). Each filter's performance was tested on a diverse set of physical AC power streams, including manually labeled and simulated clipping periods. This labeled, anonymized data will be released publicly on NREL's DuraMAT DataHub.

In the second part of this paper, we demonstrate that estimated system degradation varies when different clipping filters are used to mask AC power time series data. In our analysis of 31 physical systems, we show that sensor-based degradation rates vary based on the clipping filter used on a system-by-system basis, in some cases by up to 0.6%. Additionally, system results show that the degradation distributions for the newly developed XGBoost, Random Forest, and logic-based filters are similar, with a tighter distribution than the RdTools degradation results. We plan to further investigate if improving clipping filter performance leads to more accurate system degradation results, using simulated data with known degradation rates.

Acknowledgment

This work was authored by Alliance for Sustainable Energy, LLC, the manager and operator of the National Renewable Energy Laboratory for the U.S. Department of Energy (DOE) 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 Solar Energy Technologies Office (SETO) Agreement Numbers 34348. 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] D. C. Jordan, C. Deline, S. R. Kurtz, G. M. Kimball, and M. Anderson, "Robust PV degradation methodology and application," *IEEE Journal of Photovoltaics*, vol. 8, no. 2, pp. 525–531, Mar. 2018. [Online]. Available: https://doi.org/10.1109/jphotov.2017.2779779

[2] M. Deceglie *et al.*, "NREL/rdtools: Version 2.1.0-beta.1," 2020. [Online]. Available: https://doi.org/10.5281/ZENODO.4307010

[3] F. S. Fernández, M. A. Muñoz-García, and S. Saminger-Platz, "Detecting clipping in photovoltaic solar plants using fuzzy systems on the feature space," *Solar Energy*, vol. 132, pp. 345–356, Jul. 2016. [Online]. Available: https://doi.org/10.1016/j.solener.2016.03.013

[4] C. Deline, R. White, M. Muller, K. Anderson, K. Perry, M. Deceglie, L. Simpson, and D. Jordan, "PV fleet performance data initiative program and methodology," in *2020 47th IEEE Photovoltaic Specialists Conference (PVSC)*. IEEE, Jun. 2020. [Online]. Available: https://doi.org/10.1109/pvsc45281.2020.9300583

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] 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: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785