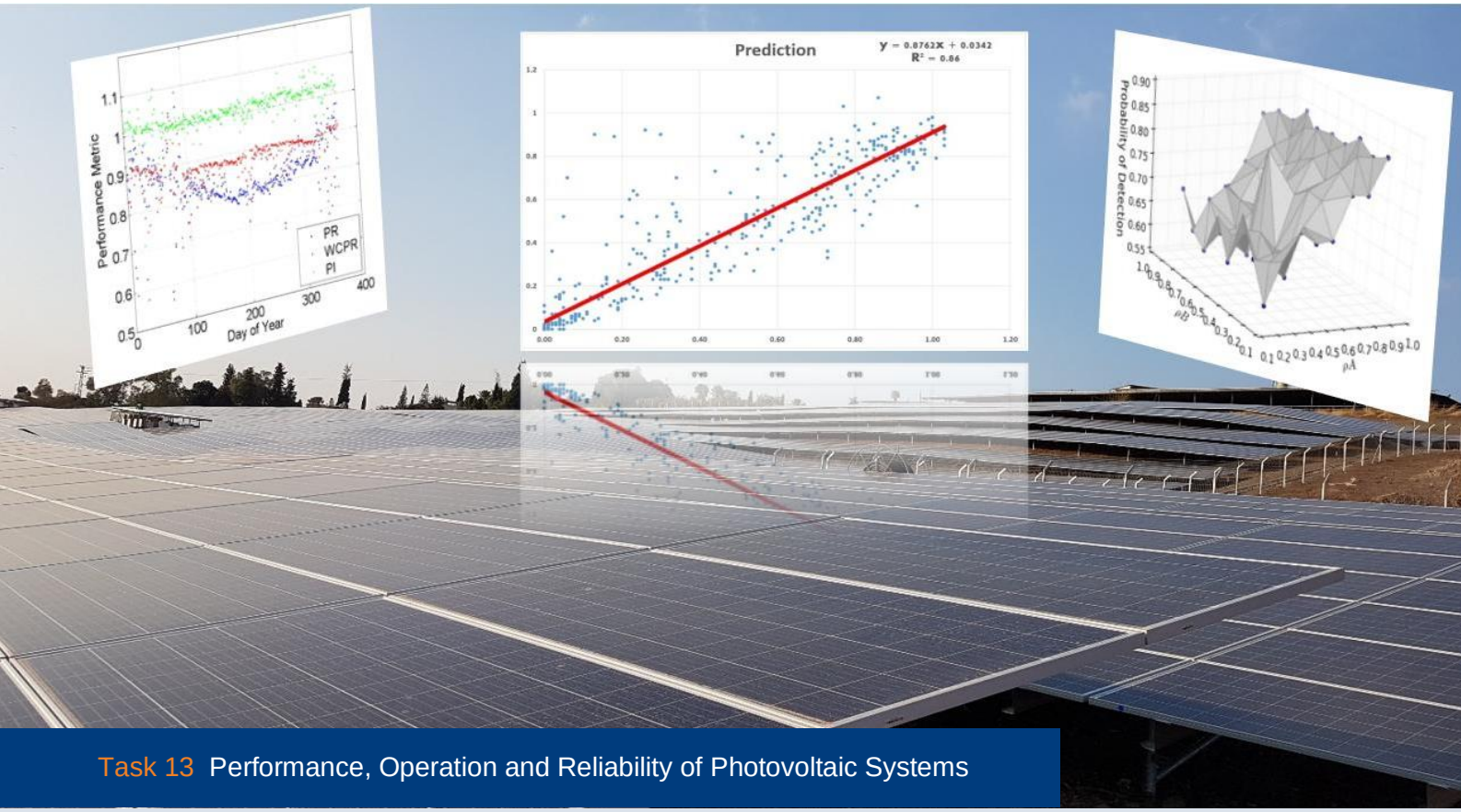
Technology Collaboration Programme
by iea

International Energy Agency
Photovoltaic Power Systems Programme



Task 13 Performance, Operation and Reliability of Photovoltaic Systems

PVPS

The Use of Advanced Algorithms in PV Failure Monitoring

2021

Report IEA-PVPS T13-19:2021

What is IEA PVPS TCP?

The International Energy Agency (IEA), founded in 1974, is an autonomous body within the framework of the Organization for Economic Cooperation and Development (OECD). The Technology Collaboration Programme (TCP) was created with a belief that the future of energy security and sustainability starts with global collaboration. The programme is made up of 6.000 experts across government, academia, and industry dedicated to advancing common research and the application of specific energy technologies.

The IEA Photovoltaic Power Systems Programme (IEA PVPS) is one of the TCP's within the IEA and was established in 1993. The mission of the programme is to "enhance the international collaborative efforts which facilitate the role of photovoltaic solar energy as a cornerstone in the transition to sustainable energy systems." In order to achieve this, the Programme's participants have undertaken a variety of joint research projects in PV power systems applications. The overall programme is headed by an Executive Committee, comprised of one delegate from each country or organisation member, which designates distinct 'Tasks,' that may be research projects or activity areas.

The IEA PVPS participating countries are Australia, Austria, Belgium, Canada, Chile, China, Denmark, Finland, France, Germany, Israel, Italy, Japan, Korea, Malaysia, Mexico, Morocco, the Netherlands, Norway, Portugal, South Africa, Spain, Sweden, Switzerland, Thailand, Turkey, and the United States of America. The European Commission, Solar Power Europe, the Smart Electric Power Alliance (SEPA), the Solar Energy Industries Association and the Cop- per Alliance are also members.

Visit us at: www.iea-pvps.org

What is IEA PVPS Task 13?

Within the framework of IEA PVPS, Task 13 aims to provide support to market actors working to improve the operation, the reliability and the quality of PV components and systems. Operational data from PV systems in different climate zones compiled within the project will help provide the basis for estimates of the current situation regarding PV reliability and performance.

The general setting of Task 13 provides a common platform to summarize and report on technical aspects affecting the quality, performance, reliability and lifetime of PV systems in a wide variety of environments and applications. By working together across national boundaries we can all take advantage of research and experience from each member country and combine and integrate this knowledge into valuable summaries of best practices and methods for ensuring PV systems perform at their optimum and continue to provide competitive return on investment.

Task 13 has so far managed to create the right framework for the calculations of various parameters that can give an indication of the quality of PV components and systems. The framework is now there and can be used by the industry who has expressed appreciation towards the results included in the high-quality reports.

The IEA PVPS countries participating in Task 13 are Australia, Austria, Belgium, Canada, Chile, China, Denmark, Finland, France, Germany, Israel, Italy, Japan, the Netherlands, Norway, Spain, Sweden, Switzerland, Thailand, and the United States of America.

COVER PICTURE

Low budget static ground-based PV system takes advantage of southern slope for high ground cover ratio built on natural unlevelled land. Photo by Mike Green

ISBN 978-3-907281-07-9 Task 13 Report The Use of Advanced Algorithms in PV Failure Monitoring

INTERNATIONAL ENERGY AGENCY

PHOTOVOLTAIC POWER SYSTEMS PROGRAMME

IEA PVPS Task 13

Performance, Operation and Reliability of Photovoltaic Systems

The Use of Advanced Algorithms in PV Failure Monitoring

Report IEA-PVPS T13-19:2021

September 2021

ISBN 978-3-907281-07-9

AUTHORS

Main Authors

Shimshon Rapaport, Green Power Engineering Ltd., Israel
Mike Green, Green Power Engineering Ltd., Israel

Contributing Authors

Carolin Ulbrich, PVcomB, Helmholtz Zentrum Berlin für Materialen und Energie GmbH, Berlin, Germany

Paolo Graniero, PVcomB, Helmholtz Zentrum Berlin für Materialen und Energie GmbH, Berlin, Germany and Freie Universität Berlin, Berlin, Germany

Atse Louwen, Eurac Research, Institute for Renewable Energy, Bolzano, Italy

Editors

Mike Green, Green Power Engineering Ltd., Israel
Ulrike Jahn, VDE Renewables, Alzenau, Germany

This report is supported by

Green Power Engineering Ltd., Israel

Arava EC&T Ltd., Israel

Eurac Research, Bolzano, Italy

PVcomB, Germany

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The Editors of this report wish to acknowledge the initial work undertaken by Dan-Eric Archer of EMULSIONEN EKONOMISK FORENING that enabled the continued research and writing of this report.

Paolo Graniero acknowledges the support of the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS).

Eurac Research acknowledges that the activities for this report were carried out in the framework of the project PV 4.0: Utilizzo di logiche Industry 4.0 e Internet of Things nel settore fotovoltaico, funded by the European Regional Development Fund PO FESR EFRE 2014-2020 Provincia autonoma di Bolzano- Alto Adige, under contract No1128.

The Authors thank David Moser of Eurac Research for his guidance.

The Editors thank Arava EC&T Ltd. for support in completing this report.

LIST OF ABBREVIATIONS

AC	Alternating Current
AI	Artificial Intelligence
ADC	Analogue to Digital Converter
ANN	Artificial Neural Networks
ANOVA	Analysis of Variance
AR	Array Ratio
CNNs	Convolutional Neural Network algorithms
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DC	Direct Current
DIP	Digital Image Processing
IEA	International Energy Agency
IV	Current vs Voltage (as in an IV-curve)
I_{sc}	Short Circuit Current
kNN	K Nearest Neighbors
LSSVN	Least Square Support Vector Machines
MAE	Mean Absolute Error
ML	Machine Learning
MPP	Maximum Power Point
MPPT	Maximum Power Point Tracker
MSE	Mean Squared Error
MWp	Mega Watt peak
O&M	Operations and Maintenance
PID	Potential Induced Degradation
PV	Photovoltaics
RAE	Relative Absolute Error
RMSE	Root Mean Squared Error
RSE	Root Squared Error
SaaS	Software as a Service
SPM	Statistical Performance Monitoring
SVM	Support Vector Machines
V_{oc}	Open Circuit Voltage

EXECUTIVE SUMMARY

This report provides an introduction to the emerging field of Statistical Performance Monitoring for photovoltaic (PV) systems and a survey of the development of these fault detection systems and their applications.

This survey found four primary methods used for identifying faults: (i) identifying faulty electrical signatures, (ii) comparing historical performance to actual performance, (iii) comparing predicted performance to actual performance and (iv) comparing the relationships between different PV systems or subsystems. The four approaches used for identifying faults include applying machine learning algorithms, statistical tests, specifying computational rules and generating simulations using models.

As shown in Figure 1, from the research papers studied, it shows that Asia is leading the world in studying and developing PV fault detection systems followed by Europe. The popularity of different parameters used by fault detection systems by developers include current and/or voltage (AC or DC) (25%), irradiance (19%), temperature (17%) and IV curve data (12%).

The study also found clear machine learning algorithm preferences. Among the papers studied artificial neural networks are the most popular (30%), followed by K Nearest Neighbors (10%), fuzzy systems (8%) and support vector machines and linear regression (7%).

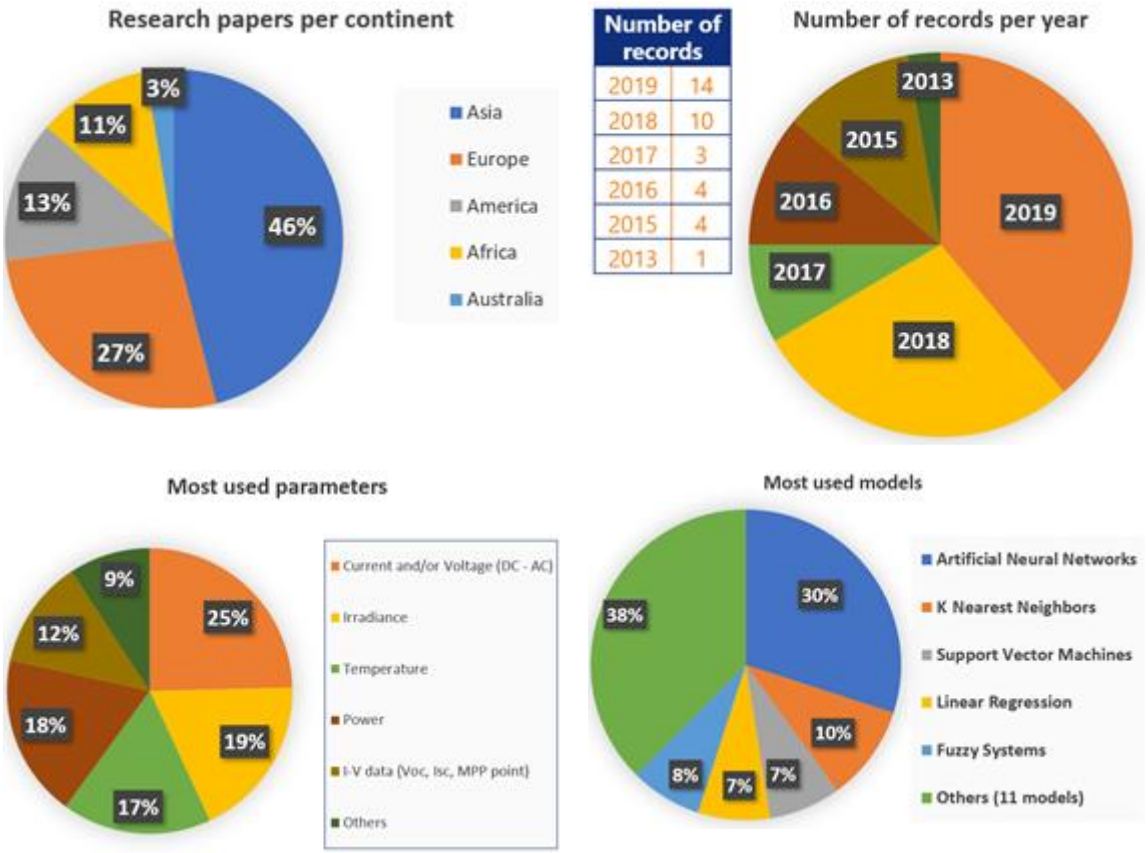


Figure 1: Overview on the analyzed literature sorted by continent, number of records per year, parameters used by fault detection systems and algorithms used.

In addition to explaining the statistical algorithms in effect and studying the approaches used for identifying faults, this paper also reviewed the different sources of data used by PV fault detection systems. Research has found that PV fault detection input data comes from a variety of devices and sources including sensors connected at the site, commercial weather stations, inverters, optimizers and IV curve tracers. Depending on the device system architecture, different parameters are available at different frequencies and accuracies.

It appears from this study that a machine learning training strategy using training data close in time to testing data provides better results and that performance data and environmental data seem to be on par with each other for some machine learning algorithms regarding accuracy of the outcome.

In comparing 8 of the 22 of the summarized algorithms in a head-to-head competition where each was fed the same data from a live PV system it was found that different algorithms have very different sensitivities.

1 INTRODUCTION

Photovoltaic (PV) energy is generated via an interaction of photons and electrons within an absorber material sealed and encapsulated in a PV module. The electrons are captured as Direct Current (DC) and converted to Alternating Current (AC) in an electronic power inverter, then the generated electricity is fed into the grid. Due to this solid-state process with few if any access points, the monitoring of PV systems beyond counting the energy produced has traditionally been of questionable value since the primary influencer in the system are the weather conditions making performance monitoring reliant solely on the existence of expensive calibrated irradiance sensors in order to enable any type of performance monitoring.

As the share of solar PV in terms of its contribution to overall electricity generation is strongly increasing in many countries, the reliability of PV electricity generation is becoming more important. National grid managers require high availability and a high level of predictability from PV energy suppliers. This demand is particularly difficult to meet in countries where the share of PV energy produced by small rooftop systems is high compared to utility grade PV power stations. Small systems are often not monitored at all while large systems are often equipped with monitoring instruments that fall short of performance monitoring that is capable of doing more than simply recording the energy production and alarming on gross data base discrepancies.

It is not surprising then that statistical performance monitoring based on artificial intelligence (AI) principles is becoming common. Since statistical performance monitoring and AI represent complex scientific topics, this document is intended to serve as a primer and reference guide, aiming to provide an introduction to machine learning algorithms and their applications in detecting PV system faults for a broad audience. In order to understand the trends in the research that is focused on enabling PV system performance monitoring, an extended literature study was performed. Out of the relevant publications identified, over 30 fault detection research papers have been used for preparing this document. 22 of these papers are described in this report in more detail.

The target audience for this report includes PV customers, PV industry personnel, inverter manufacturers, solar industry O&M companies, testing equipment developers and research institutions. Given the varying background knowledge of the target audience, this report aims at providing an easy to comprehend introduction to those in the field who are untrained in statistics just beginning to learn of PV statistical performance monitoring as well as a practical reference guide for experts actively researching the current state of PV fault detection system developments.

In order to make this report comprehensible to a larger audience, this report avoids introducing abstract mathematical equations or in-depth technical concepts. Instead, the paper emphasizes the general concepts related to machine learning algorithms and their applications to PV fault detection systems. All the papers and sources are referenced clearly to enable further study.

The study begins with Chapter 2 defining and describing the types of faults that can afflict a PV system. These are the faults that cause loss of energy production capability and are the targets of the statistical performance monitoring system. After the faults have been defined for continued reference, the next chapter, Chapter 3, outlines the general concepts defining the methods for identifying these faults using various statistical tools.

The next logical step in discussing statistical algorithms is to understand the data necessary for this process. Chapter 4 discusses different sources of data, uncertainties and approaches towards filtering noise and identifying corrupted datapoints.

With the basic concepts laid out, the statistics primer begins with explanations on statistical testing in Chapter 5. The following chapter, Chapter 6, concentrates on explaining the various machine learning algorithms found most common amongst those researchers working on PV fault monitoring.

After explaining the various algorithms, Chapter 7 presents a case study based on data of a PV campus with a number of arrays. We examine exemplarily the veracity of data set types and a number of the algorithms explained in the previous chapter.

Chapter 8 summarizes 22 papers on PV fault monitoring algorithms available for viewing online. The chapter provides a concise overview of a variety of cutting-edge fault detection systems.

The final chapter, Chapter 9, applies a number of the reviewed algorithms on a real data set and summarizes the differences between them.

2 TYPES OF FAULTS

PV failure monitoring attempts to identify physical faults through analysis of monitored digital data produced by a PV plant or module. The most general effect of faults is loss of produced energy, caused by one or more independent faults. Many algorithms work on ascertaining that a drop in energy production is caused by a fault and not the end result of a cloudy day or another uncontrollable cause, while other algorithms attempt to ascertain the individual fault responsible for this drop in energy production. One algorithm studied here attempts to find the signature of a DC arching event, a fault with greater impact on the wellbeing of the PV plant than low energy production.

This chapter lists and describes the faults discussed within the context of the publications studied for this report.

2.1 Degradation

Degradation is a general term referring to solar modules' inherent reduction in efficiency possibly due to a variety of malfunctions within the solar module. Common causes of degradation include hotspots, Potential Induced Degradation (PID), cracks in the solar module, solar module delamination, bubbles in the solar module, yellowing/browning of the solar module Ethylene-vinyl acetate layer, to name but a few causes. All solar modules undergo some degradation over the operational lifetime. Accordingly, manufacturers typically provide warranties to accommodate for degradation greater than expected.

Identifying and subsequently proving that an operational PV plant is showing degradation rates that are higher than manufacturer's guarantees, however, may be difficult. Degradation typically occurs gradually, and so is not apparently visible when analyzing short term data. Since the manufacturers' warranty relates to annual degradation, the data set being analyzed must span at least two years [1].

2.2 Shading

A variety of assumption and methods are applied in an effort to identifying shading. Zaki et al. [2] differentiate between PV systems where, under shading conditions, bypass diodes are closed and not closed.

Green et al. [1] identify shading by assessing the energy production patterns of the PV system over time. The authors identify faults in systems by comparing actual PV system output to modelled PV system output. The PV fault detection system learns to identify consistent patterns of PV system underperformance identifying the reduced performance as shading.

2.3 Hot spots

Hot spots can occur for a variety of reasons including as a consequence of shading, solar cell cracks and a variety of other solar module malfunctions. Identifying hotspots is an important task since hotspots typically grow and can spread within a solar module eventually leading to failure.

The common method used today for identifying hotspots involves using a thermal imaging camera to search for them manually. In small systems maintenance personnel may examine each array individually. In large scale systems drones are used to carry the thermal cameras. By closely examining the drone's footage records, hotspots are identified. In a study presented

by Vidal de Oliveira et al. [3], hotspots are identified in aerial thermal images by applying digital image processing and convolutional neural network algorithms.

2.4 Inverter clipping

Inverter clipping occurs when the solar module DC power is larger than the rated inverter AC output. In such a case the inverter limits the DC power production to the inverter's power limits [1]. Not all instances of inverter clipping are faults. Inverter clipping is at times designed into a system to enable higher yield during the morning and evening at the expense of curtailing during peak sun hours.

2.5 String faults

String faults occur when a string stops producing power for a variety of reasons such as when the DC fuse protecting the string is blown. DC string faults can be identified when the power output of the system suddenly decreases by an amount closely equal to the power generated by one string [1].

2.6 Soiling

The term soiling combines several sources of power losses, from snow and dirt to dust and other particles that cover the surface of a PV module [4]. Soiling can be studied and predicted to a certain extent, recording the reduction in solar energy production in relation to the frequency of rain episodes during different seasons. By studying the PV systems reduction in efficiency in relation to absence of rain or between cleaning of the PV modules, producers can approximate the effect of soiling on a system and, accordingly, advise system owners on optimal times to wash their system in order to optimize performance [1].

2.7 Ground faults

Ground faults occur when there is an unexpected connection, or reduced insulation, between the PV system and the electrical grounding, resulting in current leaking to the ground thereby reducing the PV system's efficiency and creating a safety hazard. A typical cause of ground faults is damage to the insulation of the current carrying conductors transmitting the PV system electricity. To prevent ground faults from occurring, national electrical codes typically require the installation of ground fault detection and interruption devices that detect excessively leaking current to the ground. Typical solar inverters are also equipped with insulation testing circuits that detect for ground leakage [1].

2.8 Line-Line faults

A line-to-line fault occurs when two points of different potential in a PV system are short circuited, resulting in an overcurrent in the faulty circuit. Line-Line faults can occur due to short circuits between different modules in the same string or neighboring strings. Overcurrent protection devices are typically required by both national wiring codes and standardized and accepted practices for designing PV systems. The overcurrent protection devices are designed to trigger at a given current level. The short circuit capability of a PV string is very low as a percent of operating current. At low irradiation levels the current may not trigger the protection device [5].

2.9 DC arc faults

DC arc faults occur when a high-power discharge suddenly occurs between two conductors. DC arc can occur in series among neighboring conductors in a string or parallel between parallel strings. DC arcs are severely damaging to a PV system usually causing destructive fires. Since DC arcs are transient by nature they tend to be challenging to detect. Current methods for identifying arcs in PV systems include spectrum analysis of the PV systems current and voltage waveforms. In addition, arc fault circuit interrupters, installed on individual strings, can protect a PV system from arc faults [6].

2.10 AC overvoltage

Due to high resistance in the distribution grid relative to solar PV peak capacity in the nearby area, voltage may increase over the inverters' set parameters for overvoltage shut down. The cause may also be more local, too high resistance in wires between inverter and the grid connection. With compliance to the relevant national standard the inverter disconnects within the defined time span in a situation with more than the allowable voltage on the AC side. As this happens, voltage drops and after the defined reconnection time has passed, the inverter turns on again and voltage starts rising. It can take a while for wires to heat up again so the voltage may not instantly reach the high voltage again, it can take a few seconds or minutes or more before it shuts down again.

3 METHODS FOR IDENTIFYING FAULTS

A variety of methods are used by different fault detection developers to identify and classify faults including:

- Identifying electrical signatures
- Comparing historical data to current PV system behavior
- Comparing a simulated PV system to actual performance
- Comparing performance of different components or subsystems

In implementing the methods listed above, a number of approaches are used:

- Applying statistical tests to infer faults
- Applying machine learning algorithms to predict and classify faults
- Specifying instructions and rules to be programmed into a fault detection system, that specify when data hint at a fault occurrence
- Generating simulations from models

In some cases, a combination of two or more methods and/or approaches are used by a fault detection system. For example, a fault detection system method may identify electrical signatures that indicate faults by comparing neighboring array performance. More than one approach can also be used for implementing the method such as by the use of machine learning algorithms and statistical tests.

3.1 Identifying electrical signatures

There are a variety of methods used for identifying electrical signatures including identifying abnormal data patterns being received from inverters. A trivial, but simple, example of an electrical signature for identifying faults involves identifying a string inverter disconnect. In such a scenario an electrical signature can be defined by a sudden decrease in AC power equal to the amount of power provided by the string disconnected.

Another method is by cataloguing the electrical signatures of previously identified faults and generating an alert when similar electrical signatures reappear. Cataloguing electrical signatures may involve a researcher manually studying data containing faults or may involve inputting PV system datasets into machine learning algorithms that automatically identify faults and categorize them. In cases where researchers are manually studying faults, faults may be generated intentionally in a laboratory to gain an in depth understanding of the fault's electrical behavior.

One challenging aspect of identifying faults in PV systems, by the method of identifying electrical signatures, involves the uncertain behavior of PV fault detection systems under different environmental and electromagnetic conditions. For example, an electrical signature that may clearly indicate a fault under certain environmental conditions may not be a fault under different environmental conditions. To illustrate, an electrical signal may be an actual fault when the system is completely exposed to the sun with no shading. Yet the same electrical signature, under cloudy environmental conditions, may imply normal system behavior. Therefore, when a PV fault detection system identifies a suspicious electrical signature, it may apply additional fault detection analysis techniques, such as to compare historical PV system performance to determine if the electrical signal was generated during similar environmental and electromagnetic conditions in the past.

A popular type of electrical signal being used in PV fault detection systems are IV curves given that they contain meaningful information about the DC side of the solar systems state of health. When IV curves are used, different parameters of the IV curve are compared to an expected IV curve. For example, Rabhi et al., in their paper "Real Time Fault Detection in Photovoltaic Systems," study and compare the slopes of the open-circuit voltage (V_{OC}) to the maximum power point and short-circuit current (I_{SC}) to the maximum power point and compare it to an expected value under such conditions. In some cases, statistical tests are used to classify a fault by applying certain cut-off criteria such as the number of standard deviations the parameter is from an expected value (computing the level of significance). If the parameter deviates by a significant amount the fault detection system categorizes the electrical signal as a fault.

3.2 Comparing present with historical performance

Another method used for identifying faults involves comparing past system performance to current performance. In its most simple form, this method is typically used by novice PV system owners, intuitively, when they first suspect that their system is not performing as expected. In such cases, the system owners compare their electricity bill, or the PV energy produced, to the PV performance during similar times in the past. When a large deviation between historical performance and current performance is identified consistently, owners become concerned with their systems health.

Similarly, fault detection systems compare historical performance to present performance. However, in contrast to the intuitive approach of PV system owners, analysis of historical data by a statistical fault detection method is done by machine learning algorithms and statistical tests based on additional parameters other than just the produced energy parameter. Furthermore, PV system performance may be compared to performance on any time frame and in a continuous manner ranging from hours to days to months. Fault detection systems assess the system health by identifying anomalies in system performance compared to performance in the past and accordingly quantify the system's current health state and what faults might exist in the system.

3.3 Comparing predicted energy with produced energy

This method involves comparing the amount of energy a system is expected to produce with the PV system's actual performance. When the system's performance is significantly less than expected the fault detection system classifies the PV system as faulty. In most cases weather data is included in the energy prediction algorithms. Weather data may be sourced from commercial weather stations or received from sensors installed on-site. The prediction system consists of a machine learning model, and in some cases a photovoltaic model (such as those based on single diode or double diode model). Input parameters, such as past electrical behavior and weather data, are input into the model which then generates predictions.

One inherent challenge with this method is knowing the accuracy of the predictions. Since PV system performance is influenced by numerous parameters which are constantly changing, it is not always possible to know how accurate the prediction system is. Because of this difficulty, prediction systems that monitor PV performance consider how PV performance compares to predictions over time before concluding that the PV system is underperforming. Figure 2 presents a block diagram of the general method used for identifying faults using the prediction method.

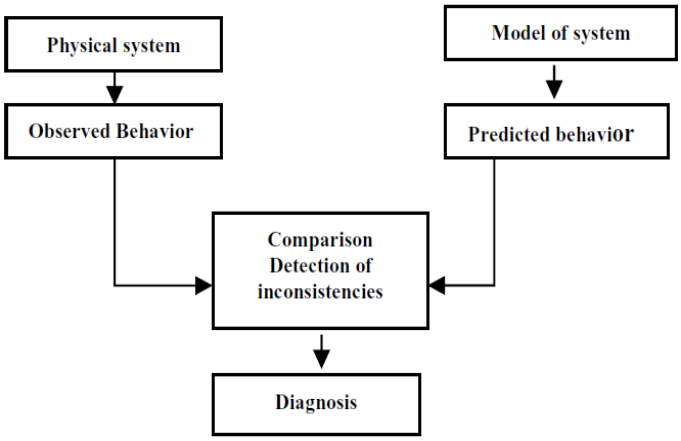


Figure 2: Block diagram of a PV fault detection system using the prediction method.

The prediction method for identifying faults provides a variety of additional advantages for the PV market by expanding the possible applications of the fault detection system. In addition to assessing the health of the PV system, predictions of system performance can be used to provide day ahead or hour ahead PV system production predictions to grid managers, renewable energy power plant owners (that are required to report to grid managers how much energy their system will produce ahead of time) and energy traders. In addition, future peer-to-peer (P2P) PV markets will require PV energy prediction forecasts for efficient energy trading.

3.4 Comparing performance of different components

Comparing performance of different systems is another popular method for identifying faults. System comparison can be on any level of granularity ranging from comparison of neighboring PV systems behavior to comparing the performance of neighboring subarrays or inverters within the same system. Once a relationship is established between neighboring systems, when the relationship begins to deviate the system can identify faulty behavior. Depending on how the deviation occurs may signify different types of faults.

Vergura et al. [7] compare neighboring system performance in identifying low-rate degradation faults. By applying a variety of statistical tests including ANOVA, Kruskal-Wallis test, Mood's median test, homoscedasticity's test and the normal probability test. They propose a method for identifying faults by comparing different systems with the same design conditions (e.g. same shading characteristics, hardware, string design). After establishing expected system relationships between different PV array parameters, faults are identified when those relationships begin to change. Depending on the parameter relationships that change different types of faults are identified.

3.5 Using statistical tests to infer a fault

While statistical tests are used in almost all methods of fault identification, such as at the core of machine learning algorithms, in some cases statistical tests are exclusively applied for identifying faults. In applying statistical tests, estimators are computed from data received from the PV system. Estimators are statistical parameters that have a statistical distribution of what an expected system value should be. Once estimators are generated, when new data is received from a PV system, their parameter is evaluated within the distribution. A variety of evaluation methods are used for determining if the received parameter indicates a fault. A simple example is a statistical test evaluating the number of standard deviations a parameter differs from the estimators mean. In this method, when a parameter is computed from incoming PV system

data, the standard deviation of the parameter is computed relative to the estimator distribution. If the parameter falls above a certain standard deviation threshold the statistical test indicates that the computed parameter is highly unlikely and classifies the computed statistic as a fault.

3.6 Statistical performance monitoring for drone mounted infrared thermal cameras

The use of infrared thermal cameras mounted on drones is an important method of identifying defective modules. A drone is capable of scanning and recording the thermal footprint of tens of thousands of modules a day. It is not surprising that a number of algorithms have been developed to analyze this vast amount of data collected by the drone in a fast and more accurate manner than is possible by human resources. In one study presented by Kirsten et al. [3], a method is proposed for detecting and classifying faults by analyzing aerial IRT images by applying Digital Image Processing (DIP) and Convolutional Neural Network algorithms (CNNs).

4 DATA USED IN FAULT DETECTION SYSTEMS

Data is the core of the fault detection system and an important feature in characterizing such systems. Some of the fault detection systems studied used real world data from live systems. In such cases, data was collected from inverters or a variety of sensors such as irradiation, temperature, wind speed and direction. Other data sources included IV curve tracing instruments and PV module optimizers. In some cases, data was collected from a controlled laboratory setting. Some fault detection systems used PV system data from an external database while other fault detection systems use simulated data made by generating datasets based on a solar energy system model (e.g., the single and the double diode solar cell model).

4.1 Inverter data

Inverters are a primary source of information for a significant majority of fault detection systems. However, different inverter manufacturers offer different parameters for collection by the monitoring system, as shown in Table 1. Furthermore, the parameter accuracies and the frequency of the data supplied are not standard among inverter manufacturers. Because of inverter manufacturers unique levels of accuracy and digital processing, different inverters may perform significantly better than others for a given fault detection system.

Table 1: Variance in type and quantity of available parameters offered by different Inverter manufacturers

Inverter #1 available parameters:			
AC current L1	AC current L2	AC current L3	AC energy
AC voltage L1	AC voltage L2	AC voltage L3	AC power
AC frequency L1	AC frequency L2	AC frequency L3	DC voltage
Energy from grid	Power factor	Reactive power	
Inverter #2 available parameters			
AC current L1	AC current L2	AC current L3	DC voltage
AC voltage L1	AC voltage L2	AC voltage L3	Ground Fault Resistance
AC frequency L1	AC frequency L2	AC frequency L3	Inverter temperature
AC power L1	AC power L2	AC power L3	Reactive power
Apparent power L1	Apparent power L2	Apparent power L3	Total Energy AC
Power factor L1	Power factor L2	Power factor L3	
Inverter #3 inverter available parameters			
Energy	AC current	AC power	DC power
AC current L1	AC current L2	AC current L3	Power control
AC voltage L1	AC voltage L2	AC voltage L3	Inverter temperature
AC power L1	AC power L2	AC power L3	DC voltage MPPT 1
DC power MPPT 1	DC power MPPT 2	DC power MPPT 3	DC voltage MPPT 2
DC current MPPT 1	DC current MPPT 2	DC current MPPT 3	DC voltage MPPT 3

4.2 Optimizer data

Solar power optimizers are sometimes connected to individual modules to optimize the solar energy coming from a PV array by preventing an inefficient solar module from reducing the efficiencies of the rest of the neighboring solar modules in the string. Optimizers may also provide owners with monitoring capabilities for individual solar modules. While optimizers may be helpful at identifying faulty solar modules and increase string efficiencies, they also increase the cost and complexity of the system and, in the case where optimizers malfunction, may reduce system efficiencies. It may be difficult to identify inefficient optimizers since there are no devices monitoring them. Below is a list of available parameters for two optimizer manufacturers.

Table 2: Overview of parameters available for several

Optimizer #1 available parameters per module:				
Module current	Module energy	Module voltage	Optimizer voltage	Module power
Optimizer #2 optimizer available parameters per module:				
Module voltage	Module current	Received signal strength indication	Optimizer voltage	Optimizer power

4.3 IV curve tracer data

The ability to analyze IV curves of individual solar modules or even strings can be a significant advantage in identifying solar module faults. IV curves are generated when the IV curve sensors implement a voltage test involving measuring the output current of a solar module for a range of voltages between 0 Volts and the open-circuit voltage over a short period of time under a known irradiation level. Depending on the characteristics of the IV curve, such as the value of the short-circuit current, open-circuit voltage, maximum power point and other parameters, a solar module can be assessed for its general health. Some PV fault detection systems apply analyses of IV curves to develop electrical signatures that indicate specific faults [8].

While individual solar module IV curve instruments can be useful in identifying faults, they are rarely found in PV systems given their cost and the complexities of installing and maintaining an extra device for each module. However, IV curve sensors are extremely useful in studying the nature of solar modules and their resulting electrical signatures under different conditions.

It is not surprising to see a trend among string inverter manufacturers in adding IV curve monitoring of strings to their proprietary portal providing a significant advantage to system owners.

4.4 Weather data

Weather data, such as irradiation, amount of cloud coverage, temperature and wind speed, humidity, precipitation, and other variables can all be valuable sources of information in monitoring PV systems. Weather sensors typically found on a PV site used for collecting weather data include pyranometers, solar irradiance cells, ambient temperature sensors, back module temperature sensors, wind speed and direction sensors. However, weather sensors are usually only available at utility and some commercially sized PV plants (typically sites larger than 1MWp) and are rarely seen on residential and small commercial PV systems given their relative expense. Because of this many fault detection developers purchase weather data services from commercial weather station services.

Table 3: Partial list of external sources for receiving weather data

Organization maintaining database	Link	Cost
SolarVu Energy Portal [9]	http://gcc.solarvu.net/	Unknown
Dataport Research Program [10]	https://www.dataport.de/who-we-are/	Free
Australia National Weather Services [1]	https://pv-map.apvi.org.au/	Free
Wunderground [1]	https://www.wunderground.com/	Paid

4.5 Uncertainty in PV data

To better understand the value of PV system data it is important to understand the flow of data from the field to the fault detection system database. For this purpose, in this section, the data chain is dissected and analyzed to identify potential data errors at each level of the data conversion and manipulation process.

The data chain begins in the field, with solar irradiation inducing an electrical current and voltage in the modules. The behavior of PV current and voltage hold the clues as to the health of the solar modules. However, unless optimizers or IV curve sensors are installed at the site, the inverter is the first point of parameter measurement. The data sensed and measured by the inverter is then transmitted to a database through various paths depending on the inverter manufacturer. A data logger is used for collecting the information from the inverter and the manner in which the data is received.

The DC voltage at the inverter's input terminal and the current flow into the inverter are sensed by voltage and current sensors connected to a microcontroller. The microcontroller utilizes the current and voltage sensors to apply one of a variety of MPPT software algorithms that optimize the inverters energy conversion efficiency. AC values, obtained by the inverter's conversion process are then used by the system operator for operational purposes including the calculation of system efficiencies and troubleshooting maintenance issues. Yet since revenue metering is not needed for the inverter's operation, there is no incentive for inverter manufacturers to invest in accurate AC monitoring hardware and software beyond what is needed for increasing the inverters efficiency, leaving the revenue metering for the system integrator. As a consequence, no standards exist for the AC power metering performed by inverters and the accuracy of AC values is typically unknown. Even in the case where the accuracy is stated, no standard for the measurement of the stated accuracy exists. Depending on the manufacturer, the physical equipment and data quality requirements of the system, the DC and AC electricity accuracies may be very different. Similarly, no standard for measuring DC values exists for inverter manufacturers. Whereas understanding the inverter's stated MPPT accuracy or frequency can enable an understanding of the necessary consequent accuracy of DC values, no such mechanism exists on the AC side. In some cases, inverter manufacturers do not publicize MPPT frequencies or accuracies.

The core electrical values of DC current and voltage behave as a direct function of solar irradiation and module temperature. The most prolific method for measuring the DC values involves the use of an Analogue to Digital Converter (ADC) device. The ADC is an electronic circuit connected to the voltage source to be measured that transforms the measured entity into a digital value. An ADC will return a discrete binary number corresponding to the analogue DC voltage sensed. As the current and voltage values change in response to the solar irradiation so do the output bits of the ADC. The ADC and the mechanism of transmitting the DC

voltage and current to the ADC contain accuracy errors. Typical ADC accuracies range from 0.25-1.5% maximum error.

Collecting sensor data using randomly sampled values of sensor input as opposed to averaged values for a clear day can be quite different both from a macro and micro point of view. Figure 3 presents pyranometer spot values sampled every 3 minutes from 08 AM in the morning to 4 PM in the afternoon versus 10-minute-average values at 10 seconds sampling interval on a clear summer day in August 2021. As can be observed from Figure 3, there may be a dramatic difference between data sampling depending on the sampling process.

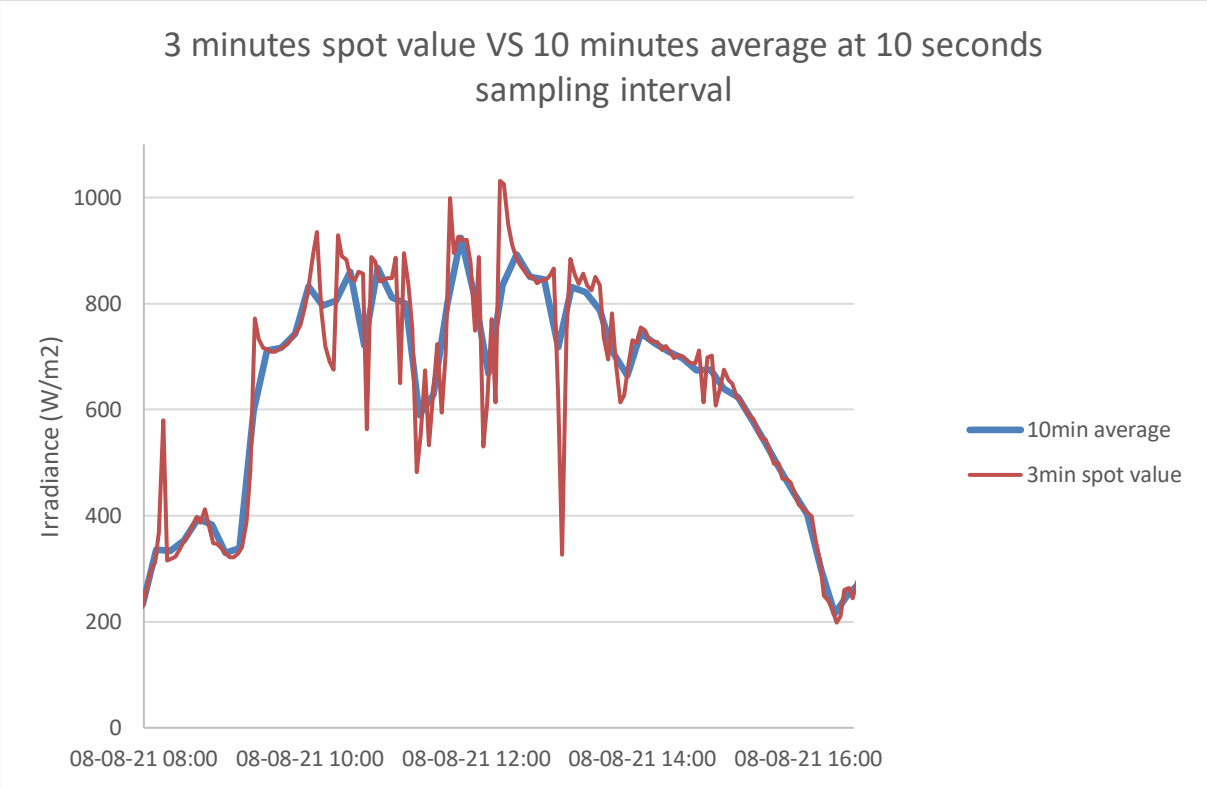


Figure 3: Minute sampling of solar irradiance

After the inverter stores the data in internal registers the data is transmitted to data loggers. In some cases, inverter manufacturers supply proprietary data loggers that collect and send the data to a proprietary web site. Some inverter manufacturers depend on third party monitoring companies to supply data loggers and monitoring web sites, while other inverter manufacturers enable connecting directly to the owner's computer desktop. Additional sources of uncertainty include dealing with missing data and the number of significant figures to be used in data collection.

The data transmitted by the inverter is not necessarily what reaches the data base. The data logger, an intermediary device responsible for transmitting the inverter data to the portal or system host, may be configured to apply a variety of digital processing techniques to the data being received from the inverter. A review of data acquisition protocols shows that averaging performed in the data logger is on 1-minute values, as per industry standard IEC61724; however, the number of values used in the final averaged parameter is unknown. Consequently, parameters received at a database may be direct transferals from the inverter, an average of a number of sampled values processed by a datalogger or a random value picked by the datalogger during specified time intervals.

4.6 Filtering noise and corrupt data

One aspect of fault detection systems is the ability to filter out noise from the data being input into the fault detection system. In a paper published by University of Oslo faculty Asmund Skomedal et al. [11], it was found that filtering noise from PV data may improve fault detection accuracy by 2-5 times. The process of filtering noise is typically crucial for the accuracy of the fault detection system since corrupt data will inherently cause the fault detection system algorithms to identify faults that do not exist or overlook underlying faults. The effect of noise on fault detection performance depends on a variety of factors such as the types of data input into the fault detection system, the algorithms used, the statistical models used and the data acquisition system that transmitted the data to the database.

Noise can occur for a variety of reasons. Some causes of corrupt data include communication failures between the data logger and sensors, misalignment or misconfiguration of measurement sensors, glitches in the software or communication protocol, system outages, sensors operating outside of their specified operating conditions and noise effects by the environment the sensors are situated in [12].

A standard practice for PV monitoring systems is only to collect data above or below certain thresholds as a method of eliminating noise. For example, observations including irradiation measurements below 20 W/m^2 are often removed from a dataset, since under such irradiation conditions, irradiation sensors tend to be highly inaccurate and overly sensitive to environmental conditions. Another method for eliminating noise involves identifying faulting sensors by comparing sensors performance. However, in the case of comparing multiple sensor measurements to each other in identify faulting sensors, it is difficult to determine which sensor is accurate and which is inaccurate, which is not trivial. Therefore, a standard industry practice is to send sensors to be calibrated as per O&M protocols decided upon at the beginning of plant operation or in cases where the sensor is suspected of malfunctioning.

While filtering data is an important aspect of fault detection systems, the process of filtering data brings a variety of challenges. For example, how can the filtering process differentiate between noise and abnormal PV system behavior since under both circumstances data may be abnormal and could be perceived as noise? Conversely, in cases where data above and below certain thresholds are removed from the dataset, the fault detection system may be overlooking faults present in the filtered data. To avoid filtering data containing potential system faults, observations above and below certain thresholds are only removed when under such conditions the PV systems production is minimal and identifying faults during these periods will not entail losing significant amounts of energy.

5 STATISTICAL TESTS

Statistical tests are a group of mathematical computational methods that make conclusions about test statistics, such as the average and standard deviation of a data population or how different data populations differ. There is a variety of different test statistics used depending on data characteristics such as how the data is distributed (such as bell-shaped curve, uni-modal, bi-modal), if the data involves categorical or continuous data and other data characteristics. There are numerous statistical tests that can be used in drawing conclusions about data, including the following partial list of methods:

- Independent T-Test
- Mann-Whitney Test
- Paired T-Test
- Analysis of variance (ANOVA) test
- Kruskal-Wallis Test
- Friedman Test
- Chi-Squared Test
- Cohen's Kappa
- Proportion Z-Test

5.1 Hypothesis testing

Statistical tests are used in hypothesis testing. Hypothesis testing is a statistical way to evaluate the likelihood between at least two conflicting theories or hypotheses. In the context of fault detection, the two possible hypotheses could be fault or normal for the given state of a PV system. When studying PV field data, hypothesis testing can be used to determine if the data in question indicates a fault and what type of fault may have occurred. The null hypothesis is the condition we usually assume as normal, as opposed to the alternative hypothesis that needs evidence to be considered true. Conceptually, this is similar to a court trial: the null hypothesis, in this case, would correspond to the position of the defendant, innocent until proven guilty, while the alternative hypothesis, guilty, must be grounded with enough evidence (summarized in the test statistics) to be accepted as true. We reject the null hypothesis in favor of the alternative hypothesis when statistical test results indicate a high level of significance. A variety of statistical tests can be used in evaluating data for identifying PV faults.

A test statistic is any sample statistic (a function of the data) that is used to decide whether to reject the null hypothesis or not. For a test statistic to be valid, its sampling distribution under the null hypothesis must be unbiased. It is then possible to compute the p-values, that allow us to determine how likely or unlikely an outcome of an experiment is, considering the null hypothesis is true.

One way to distinguish between statistical tests is based on the assumptions we make about the distributions of the data. Usually, parametric tests can be used when the data is distributed in certain shapes, while non-parametric tests make more lenient assumptions about the distribution (or shape) of the data and therefore are more general. Parametric tests provide more confidence in the results of the tests.

5.2 Analysis of variance (ANOVA)

Analysis of variance (ANOVA) [13] is an example of a parametric statistical test used to determine if two or more sample population means are equal. To apply the ANOVA statistical test, the data of the two sample populations is aggregated into one sample and the mean is computed. The ANOVA test then measures and compares the difference between the individual means and the aggregated mean. If the difference is statistically significant, indicated by its p-

value, we reject the null hypothesis of equal means and accept the alternative hypothesis that the two samples come from different distributions.

The ANOVA test assumes that:

1. The datasets being analyzed come from a normal distribution.
2. The data observations are independent, meaning that any collected observations are not influenced by other observations in the dataset.
3. The variances of the datasets being tested are equal.

A variety of statistical tests can be used to ensure that a dataset meets the required ANOVA assumptions before applying this test.

ANOVA can be used as part of a fault detection system by testing different parameter data to see if their means are equal or not. If we consider several arrays with very similar conditions (located in the same geographic location, having the same system design, subject to the same environmental conditions throughout the day, we can assume that the external conditions (temperature and irradiance) are the same for all arrays; therefore, we expect the different arrays to produce approximately the same amount of energy. In such a case, ANOVA can help in assessing whether there is an array that is underperforming compared to the others. We can collect the energy produced by each array over a given time, for example, one month. After checking that the assumptions for applying ANOVA are satisfied, we can compare the different distributions.

The result of applying ANOVA is a p-value that we can interpret as the probability that the means of all the samples are equal. For example, assuming a p-value of 0.01, it can be assumed that the probability of observing the statistical test results is only 1% making the alternative hypothesis more likely.

An application of a variety of statistical tests including the ANOVA test is presented by Silvano Vergura [14] in his research paper titled “A Statistical Tool to Detect and Locate Abnormal Operating Conditions in Photovoltaic Systems.” In his paper, Vergura compares DC current, DC voltage, AC current and AC voltage of different subarrays in order to identify faults in subarrays. If there is a dramatic difference between two subarray means, when they should be equal, the fault detection system identifies that the PV system is underperforming. To verify if the data received from the PV system can be analyzed using the ANOVA test, Vergura applies the Hartigan’s Dip test, Moods median test and the Kruskal-Wallis test to determine if the variances are equal – a requirement for using the ANOVA test. In cases where the dataset is not normally distributed or the variances of the different subarray parameters are not equal, Vergura applies the Kruskal Wallis test or Mood’s median test.

Note that via this method, it is not possible to identify the cause of a fault, but only to locate a fault in a given set of arrays. Such a technique is easy to implement and requires minimal data (only energy production data).

5.3 Bootstrapping

Bootstrapping is a resampling method used for estimating the probability distribution of estimators such as the mean or the correlation coefficient of a population, by sampling with replacement from a dataset. By randomly collecting a certain percent of the total observations from a dataset, with replacement, repeatedly, and for each sample computing the statistical parameter in question, the dataset parameter’s distribution can be estimated. For some statistics, bootstrapping is inherently biased, such as in computing the variance of a population. For parameters with a suspected bias, adjustments need to be made to the resulting bootstrap

distribution by adding the difference between the original sample data and the bootstrap sampling distribution data.

To illustrate how bootstrapping can be applied, consider a pyranometer collecting irradiation data on the windowsill of a building. The pyranometer sits in the shade for half the day and at midday it is suddenly exposed to direct sun. The resulting data appears exponential and therefore not bell shaped preventing the use of standard statistical parametric tests. Instead, non-standard tests for determining population estimates need to be applied. The mean of this data can be calculated using bootstrap. By resampling from the collected irradiation data many times, and computing a statistic for each bootstrap sample, a confidence interval estimating the average of the irradiation is established. Depending on how well the sampled data reflects the actual data determines the accuracy of the resampling distribution created by bootstrapping; see [15] for a detailed description.

6 MACHINE LEARNING ALGORITHMS

Arthur Samuel, an American machine learning pioneer, defined machine learning as a subfield of computer science that gives 'computers the ability to learn without being explicitly programmed'. Before the advent of machine learning, computers needed to be provided explicit rules in order to categorize cases, with a minimal ability to generalize computational analysis to observations or situations never seen before. In contrast to traditional computer programs, machine learning programs provide computers with general instructions instead of explicit rules [16]. In this document, we consider machine learning to be an application of artificial intelligence (AI).

The ability to identify faults and even predict them requires sophisticated data analysis and decision-making algorithms. While some PV experts can analyze data manually, by visually observing system behavior of various variables contained in real PV datasets, the ability for computers to replace humans in this task is a primary goal of fault detection systems for increasing the speed and accuracy of PV fault identification. The way fault detection systems develop the skill necessary to replace humans at this task and gain the ability of identify and predict faults independently, is through a variety of computational processes provided by the field of machine learning. In developing PV fault detection systems, a variety of machine learning principles are explored to identify the algorithm that provides the best results for predicting PV faults. This is done by splitting PV fault detection data into mutually exclusive sets and training different machine learning algorithms to accurately detect PV faults hidden in the data. The training data provides the computer examples of observations and outcomes which the computer can learn. The test data is then used to test the computer's ability to generalize the predictions to previously unseen data.

Machine learning algorithms can be categorized according to the task that they attempt to solve:

1. **Regression/estimation** is used to predict continuous values (rather than discrete values). Predicting solar PV energy production is considered a regressive task since energy can be any continuous decimal number.
2. **Classification** is used to classify data into categories. An example of classification can be a PV fault detection system that classifies different types of faults in a dataset depending on different patterns identified by the classifier.
3. **Clustering** is used for segmenting data into homogenous groups. Clustering can be used to find faults in a system.
4. **Association pattern mining** is used for finding items or events that co-occur in a dataset. Association pattern mining can be used to determine if conditions in a PV system are occurring at the same time and thus identify the state of the PV system and if it is performing optimally.
5. **Anomaly detection** is used to identify abnormal and unusual cases. For example, anomaly detection algorithms can identify abnormal PV system behavior, that is, faults.
6. **Sequence mining** is used for identifying upcoming events given a sequence of current events. Sequence mining can be used for predicting future faults.
7. **Dimensionality reduction** is used to reduce the size of data.
8. **Recommendation systems** are usually used to predict people's preferences with others who have similar tastes and recommends new items to them accordingly, not an inconceivable parallel to fault analysis.

Another way of classifying machine learning algorithm is by categorizing them as supervised or unsupervised learning algorithms. Machine learning algorithms are supervised if the data contains the outcomes of the observations in the training and testing datasets. Supervised

data allows computers to approximate or classify future unknown observations. In contrast, unsupervised learning involves providing the computer training data that does not include the resulting outcome of the data. The computer must draw independent conclusions on the dataset's outcome. Since unsupervised learning provides the computer less information than supervised learning, unsupervised learning techniques typically involve more complex algorithms since the computer is expected to predict outcomes without knowing the consequence of previous observations. Unsupervised machine learning techniques include dimensionality reduction, density estimation, market basket analysis, and clustering. Dimensionality reduction plays an important role in unsupervised learning algorithms by reducing the dimensions of the data, thereby making computations faster.

One confusion that arises in the study of machine learning is in understanding the difference between artificial intelligence, machine learning and deep learning. AI is the general technological development of computers for the purpose of making them intelligent. By writing programs that provide instructions similar to how humans process information, machines develop human like decision making abilities. Machine learning is a sub-branch of AI that deals with the statistical aspects of making machines intelligent by teaching the computer to solve problems by training the computer on numerous scenarios. After the machine is taught a variety of different scenarios, using statistics and probability, that computer generalizes the results to solve problems not included in the scenarios provided during training, with some probability of success. Deep learning is an advanced field of machine learning that involves a deeper level of automation in contrast to general machine learning techniques, for details see [17].

6.1 Regression

Regression is a machine learning method used to teach computers to predict a continuous dependent variable (a variable that can be any real number) by inputting into the regression algorithm independent data features, also known as explanatory variables. Regression models can be categorized into simple regression and multiple regression. Both types of regression can be linear or nonlinear. There are numerous regression models such as ordinal regression, Poisson regression, fast forest quantile regression, linear regression, polynomial regression, lasso regression, stepwise regression, ridge regression, Bayesian linear regression, neural network regression, decision forest regression, boosted decision tree regression and K nearest neighbors' regression [17].

6.1.1 Simple linear regression

Simple linear regression is used when there is only one independent variable for predicting the dependent variable. When more than one independent explanatory variable is present, the process is called multiple linear regression. Two advantages of using linear regression are that it can predict response variables very fast given its simple computation process. In addition, simple linear regression does not require parameter tuning, it is easy to understand and highly interpretable. Figure 4 illustrates the use of a linear regression model. As can be observed, the random variable x_1 predicts the income of an individual based on their age. The linear regression model was constructed applying a computation on the data. The blue dots represent data that was used for generating the linear regression model while the red dot represents a new data-point, the age value, and we can check how well the regression model predicts the income.

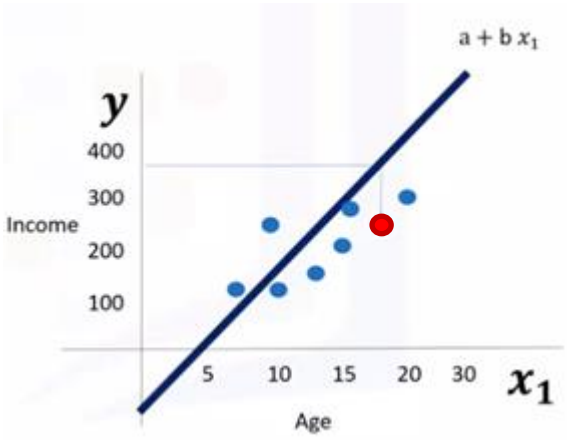


Figure 4: An illustration of simple linear regression. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

Linear regression is a statistical method of studying the relationship between two variables by generating a linear function. The linear function is created as follows,

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \tag{1}$$

where $\hat{\beta}_1$ and $\hat{\beta}_0$ are defined as

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_1}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2}$$

$$= r\frac{s_y}{s_x} \tag{2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \tag{3}$$

And (see [18])

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right) \tag{4}$$

r is called the correlation coefficient and provides a measure of how similar two variables behave with respect to each other. Note that r does not imply that one causes the other. Parameters s_x and s_y are the standard deviation of x and y, respectively, which measure the spread of the x and y variables. Besides providing a measure of correlation between two variables, linear regression allows us to predict the outcome of x values that have not been observed before. Note that $\hat{\beta}_0$ and $\hat{\beta}_1$ and ϵ_i are estimators and can only be approximated, for details see [17].

6.1.2 Multiple linear regression

Multiple linear regression is an extension of simple linear regression and involves predicting a dependent variable's behavior based on more than one independent variable. It is used when there is a linear relationship between each of the input parameters and the output parameter. The linearity relationship between the inputs of each variable with respect to the output can be verified in a variety of ways including using scatterplots and computing the correlation coefficient between each of the input variables and the output variable. If for all variables the correlation coefficient is 0.7 or greater, it is safe to assume that there is linear tendency. In cases

when linear regression provides inaccurate results, nonlinear models should be considered to model the data.

Regression can also be used to determine the strength of the effect that one of several independent variables has on the dependent variable. For example, after collecting sufficient amounts of data on a PV system containing temperature, voltage and irradiance, regression can be used to determine how much an increase of temperature by one degree may have on the energy output of the PV system, holding voltage and irradiance constant. Applying regression allows for determining which of the independent variables are meaningful in predicting the output variable and which only affect the outcome slightly.

The general form of a multiple linear regression model is

$$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \tag{5}$$

There are various methods to estimate the beta regression parameters β_i, all of which aim to minimize the error between the prediction and the actual value observed. The two most common methods used to minimize the error between the predicted and actual outcome of a multiple linear regression model are the ordinary least squares and optimization approaches.

Ordinary least squares estimates the values of the coefficients by minimizing the mean square error using linear algebra. The disadvantage of this method is that it can take substantial time to optimize the regression model. A general rule of thumb is to use the ordinary least square method for data sets with less than 10000 observations.

A second method for minimizing the multiple linear regression model error is to apply a variety of optimization algorithms. For example, gradient descent is an optimization algorithm that begins optimization using random values for each coefficient, calculates the errors, and iteratively modifies the coefficients to reduce the error. Gradient descent is a good choice for large datasets given that it is less computationally intense and can process datasets relatively fast. There are numerous additional optimization algorithms used for optimizing multiple linear regression models.

When implementing multiple linear regression, it is important not to add too many input variables since it can cause the model to become overfit, making the model sensitive to noise in the data. Such a model is too complicated and not general enough for new data observations. When applying data including categorical inputs, these inputs should be converted to discrete numbers since multiple linear regression input data must only be numerical. For example, if the inverter fan being ON or OFF is a variable used for predicting energy output of a PV system, ON can be converted to the number one and OFF can be converted into the number zero for regression modelling purposes. [17]

6.1.3 Non-linear regression

In cases when observed data on a scatterplot is not linear, a non-linear regression model should be used. One type of non-linear regression model is polynomial regression, where the relationship between the input variables and output variable can be modelled as an n^{th} degree polynomial.

One example of polynomial, non-linear regression model is

$$\hat{y} = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 \tag{6}$$

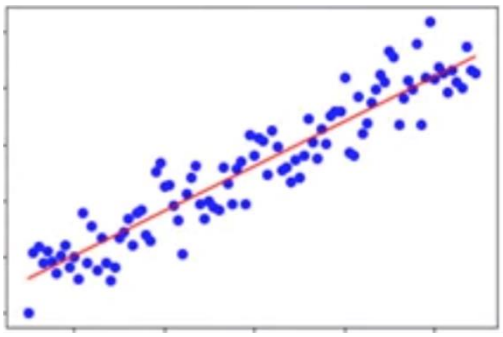


Figure 5: An Illustration of linear regression. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

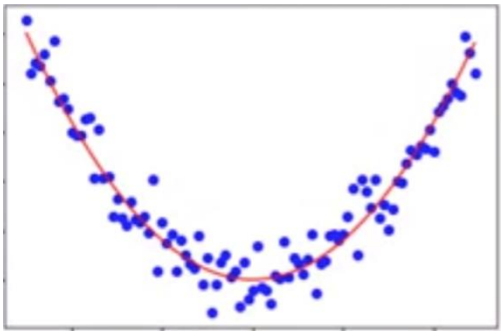


Figure 6: An Illustration of quadratic (parabolic) regression. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

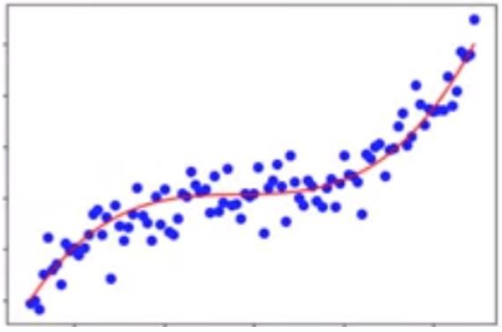


Figure 7: An Illustration of cubic regression. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

Using substitution, polynomial nonlinear regression can be converted into a multiple linear regression problem and consequently the least squares optimization algorithm specified above can be used.

In cases when polynomial regression does not suffice, numerous additional nonlinear models are available, including exponential models, logarithmic models and logistic models to name a few. Below are a number of non-linear regression models that can be used on different data distributions,

$$\hat{y} = \theta_0 + \theta_2^2 x \tag{7}$$

$$\hat{y} = \theta_0 + \theta_1 \theta_2^x \tag{8}$$

$$\hat{y} = \log(\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3) \tag{9}$$

$$\hat{y} = \frac{\theta_0}{1 + \theta_1^{(x-\theta_2)}} \tag{10}$$

6.1.4 Regression trees

The regression tree is a statistical method for modelling nonlinear multivariate datasets with continuous numbers for prediction purposes by subdividing data into partitions. Regression trees allow for modelling datasets in a piecewise manner rather than modelling the entire data set as a whole. This method is effective when the dataset, as a whole, is not easily modelled. By partitioning the data into carefully selected subsets, regression can be applied.

The general steps for modelling data applying a regression tree are as follows:

1. Select a value k which specifies the smallest number of observations in a partition.
2. For each attribute compute the sum of squares for all possible partitions.
3. Select the attribute and partition with the smallest sum of squares to be the primary node.
4. Repeat step 2 and step 3 excluding any data that has already been included in the regression tree until all nodes are at least size k or the sum of squares per node is minimized.

In the contribution 'Improving Efficiency of PV Systems Using Statistical Performance Monitoring,' Mike Green and Eyal Brill [1] apply regression trees for predicting the amount of energy a PV system may produce. By identifying relationships in data partitions of different weather and inverter data, the authors predict how much energy is expected to be produced. When the system does not meet energy production expectations, the PV monitoring system alerts that the system may not be performing as expected.

6.1.5 Half-sibling regression

The half-sibling regression method is a statistical regression method used to remove the confounding effects of a confounding variable of both an independent and a dependent variable. A confounding variable is a variable that affects both the independent and dependent variable being studied. A general illustration of how unobserved confounding variables can affect the study of input and output variables is presented in Figure 8.

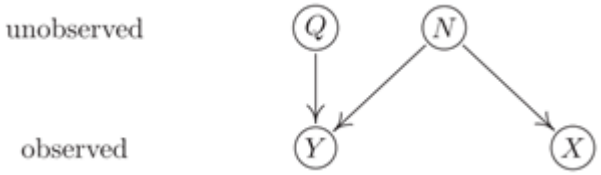


Figure 8: General illustration of the relation between unobserved and observed variables. Source: Iyengar et al. [10].

If Y and X represent the observed datasets of two neighboring PV systems, and X is independent of Q, half-sibling regression allows for estimating Q.

Since X and Q are independent of each other, X alone cannot provide information about Q. However, Since X and Y are both dependent on N, X can provide information about N and how N influences Y. Specifically, by trying to predict Y given X we obtain information about how N affects Y. By discovering how N affects Y, N's effect on Y can be removed which allows for determining how Q affects Y.

The following conditions must hold in order to apply half sibling regression:

1. X should be independent of Q
2. Y can be easily predicted by X such that a simple function class solves the regression problem f(N). Typically, this requires that N affects both X and Y in a similar way. However, it does not require that f(N) be linear.

Half-sibling regression has been innovatively applied by Iyengar et al. [10] in developing the Solar Clique method. In applying half-sibling regression to public PV datasets, the developers categorize causes for energy reduction into two categories: transients and anomalies. Transient causes for energy reduction include temporary factors such as weather and shading. Anomalies cause ongoing reductions in energy production and are due to system malfunctions such as faulty hardware or bird droppings. Transient causes are further classified as common or local factors. Common factors include causes for power reduction that affect the region where the neighboring PV sites are located. Local factors include transient causes of power reduction that are unique to the site being monitored, such as shading. A unique characteristic of transient local factors is that they are site specific and do not affect other sites. In summary, Solar Clique's key challenge in designing a solar fault detection system is the systems' ability to differentiate power reductions of the monitored system due to transient factors versus anomalies.

Table 4: Explaining transient causes versus anomalies.

Causes of energy reduction		Description	Example
Transient: temporarily affect energy production	Local	• Factors causing energy reduction unique to monitored site • Reduce power at fixed periods during the day	Shading
	Common	• Factors causing energy reduction affecting neighboring PV sites	Weather
Anomalies		• Factors causing prolonged energy reduction • Require corrective action to restore optimal PV performance	Hardware malfunction, bird droppings

General methodology

First, the authors define the following variables:

Y: power generated by a monitored system

X: power generated by a set of neighboring systems

C: common factors affecting both systems

L: site specific local factors affecting the system including transients and anomalies

Note that the values of *C* and *L* are unknown. The goal of the fault detection system is to determine the anomalies in *L*.

Note a number of characteristics illustrated by the Figure 8:

X, the power data from neighboring PV systems being monitored, is independent of L. In other words, the power behavior of neighboring PV sites is independent of the factors that temporarily affect the monitored PV site power output. Secondly, since X and Y are dependent on C, there is a correlation between them. Furthermore, when probabilistically conditioning X given

Y, Y becomes a collider random variable, meaning that both X and C influence the resulting solution. This implies that X can provide an estimate of L. Through mathematical manipulations the authors find the following equation to hold true

$$\hat{L} \coloneqq Y - E[Y|X] \tag{11}$$

First step: multiple half-sibling regression models are built to predict the energy of the monitored PV system (Y) based on the neighboring PV sites performance (X). This is done applying bootstrapping methods to obtain an estimator of L and its statistical parameters such as its standard deviation.

Second step: time series decomposition techniques at a weekly resolution are used to separate the local transients from the local anomalies, creating the anomaly estimator *A*. To create the anomaly estimator, it is assumed that the local transients do not vary much on a daily basis. Since shading, determined by the sun, is fairly consistent throughout the year, the anomaly estimator can be used to filter occurrences of shading easily.

Third step: Using the derived estimator *A*, days are flagged as anomalous when three conditions hold:

1. The deviation of $\hat{A}_t$ should be statistically significant
2. The anomaly exists for an extended period.
3. The anomaly occurs during the day.

Thus, an anomaly can be defined as:

$$anomaly = \left(\hat{A}_t < -4\sigma_t\right) \wedge \ldots \wedge \left(\hat{A}_{t+k} < -4\sigma_t\right) \quad \forall_t \in T \tag{12}$$

Iyengar et al. [10] applied half-sibling regression for identifying PV system faults.

6.1.6 Evaluation metrics in regression models

Once a regression model is chosen, it is necessary to determine how well the model fits the data. While there are numerous approaches to evaluating regression models, three popular approaches are:

- **Train and test on the same dataset.** one method for evaluating how well a model fits a dataset is by creating a regression model using the entire dataset and then inputting a subset of the dataset into a model and comparing the predicted values to the actual values. This method typically demonstrates a high training accuracy, but can result in overfitting. Overfitting occurs when the models are capturing noise from the dataset, because the model is essentially being constructed to perfectly reproduce the training dataset, rather than to make (generalized) predictions on unseen data. This method typically has a low out-of-sample-accuracy, which is the accuracy of the model on new data inputs that the model has not been trained on.
- **Train and test split.** This method is used when the dataset is split into data used for training the model and data used for testing the model. Using this method provides a higher out-of-sample-accuracy by fitting the regression model more effectively to the nature of the dataset. This method is more realistic for real world problems. After testing the model on the test data, this data should be included in the training dataset for the deployed model.
- **K-fold cross validation.** This is a more advanced method for determining the accuracy of a model by splitting a dataset into mutually exclusive subsets. The data can be split multiple times and the accuracy of the model computed on different subsets of data.

Using this method, multiple accuracy values are generated depending on the number of data splits created. The resulting computed accuracies are then averaged to obtain a relatively precise model accuracy score.

There are various metrics used for evaluating the accuracy of a given model such as the mean absolute error (MAE), mean squared error (MSE), and the root mean squared error (RMSE). An error is defined as the difference between an estimated value and the regression line generated by the model [17].

- **Mean Absolute Error ($\boldsymbol{MAE}$)**. It is the simplest type of error to understand, since it is just the average error of all *n* tested data points (*n* just being the number of data points tested), and is computed as follows

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{13}$$

  y_j being the value measured at the j'th measurement and $\hat{y}_j$ being the value predicted by the model.

- **The Mean Squared Error ($\boldsymbol{MSE}$).** Taking the square of the difference between measured and predicted values, larger differences are emphasized. This is the rationale behind the mean squared error metric

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{14}$$

- **The Root Mean Squared Error ($\boldsymbol{RMSE}$).** While MSE emphasizes large errors, taking square root of the MSE makes it is easier to intuitively understand the error since the error metric is now in the same dimensions as the response variable,

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \tag{15}$$

- **The Relative Absolute Error ($\boldsymbol{RAE}$)**. Also known as the residual sum of squares normalizes the total absolute error by dividing the total absolute error,

$$RAE = \frac{\sum_{j=1}^{n}|y_j - \hat{y}_j|}{\sum_{j=1}^{n}|y_j - \hat{y}|} \tag{16}$$

- **Root squared error ($\boldsymbol{RSE}$)**. RSE is widely adopted by the data science community since it is used for computing the very popular statistic RSE or R^2,

$$RSE = \frac{\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}{\sum_{j=1}^{n}(y_j - \bar{y})^2} \tag{17}$$

6.2 Classification

Classification is a supervised learning approach used to categorize unknown observations into discrete categories or "classes", implying that the target attribute is a categorical variable, assuming discrete values. By training a model of labelled observations based on a variety of attributes, the model can then categorize new observations. There are a variety of classification algorithms, some of which are listed below:

- Decision trees
- Naive bayes
- Linear discriminant analysis
- K Nearest Neighbors
- Logistic regression
- Neural networks
- Support vector machines

6.2.1 K Nearest Neighbors (kNN)

The kNN algorithm classifies an element by determining the k closest neighbors to the element. The algorithm is easy to implement and can be used both for classification and regression tasks. The basic idea of this algorithm is to compare a given data point x_{new} with the k training data points x_i that are closest to it. There are a variety of metrics that can be used for determining the nearest neighbors, such as the Euclidean distance. When classifying a new data point, x_{new} is assigned the most common class among its k nearest neighbors. kNN can also be used to predict continuous regression variables by assigning to x_{new} the average or median value of its k nearest neighbors.

The steps of the kNN algorithm are as follows:

1. Pick a value for k.
2. Calculate the distance between all training data points and the data point being classified.
3. Identify the k data points closest to the data point being classified.
4. Classify the unknown data point according to the majority of the K nearest data points. Alternatively, compute the mean or median of the data points when approximating a continuous variable.

One method of computing distances between the data points being classified and its k Nearest Neighbors is using Euclidean distances. Euclidean distances are computed as follows,

$$Dis\,(x_1, x_2) = \sqrt{\sum_{i=0}^{n} (x_{1i} - x_{2i})^2} \tag{18}$$

When choosing a value for k, it should not be too small or it may result in an overfit model, since outliers will have a more significant effect on the model making it overly sensitive to noise. Overfit models are not generalizable for new classification cases. On the other hand, the k value should not be too large to allow the model to learn local patterns. When deciding on the optimal k-value for the kNN model, the data should be split in a training dataset and a testing dataset. Then the model should be trained and tested for a variety of values of k in order to determine the optimal value for the kNN classifier model.

The main drawback of this approach is the need to compute the distance between each training data point and x_{new}. This can be computationally intense, requiring increased energy, time,

and computer processing power, on large datasets. An advantage of k Nearest Neighbors is its simple implementation and the clear interpretability of results. [17]

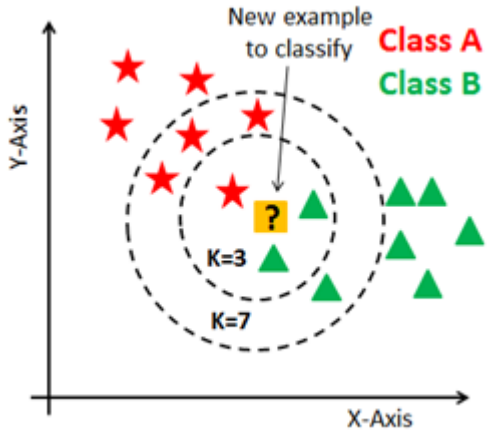


Figure 9: Illustration of the kNN algorithm. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

Figure 9 illustrates of the kNN algorithm for different k values. As can be observed by class A utilizing a k value of 3 versus class B utilizing a k value of 7, depending on the k-value used, data points may be classified differently.

6.2.2 Logistic regression

Logistic regression is a method of classifying categorical data. Logistic regression requires that the data provided be linearly separable, meaning that when the data is plotted a line can be drawn that completely separates the two sets from each other. Logistic regression utilizes an inequality to compute the probabilities in categorizing observations. For a two-dimensional dataset the logistic regression characteristic equation is in the form,

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 > 0 \tag{19}$$

The line separating the two datasets, called the decision boundary line, does not need to be linear. As long as the decision boundary line can be modelled using an inequality, the dataset can be modelled using logistic regression.

One advantage of logistic regression is that it can provide the probability that an observation belongs to one class or the other rather than simply categorizing an observation as belonging to one class or the other. Furthermore, logistic regression can provide insight into how significant an attribute is based on the magnitude of the logistic regression coefficients θ_i found by optimizing the linear decision boundary line [17].

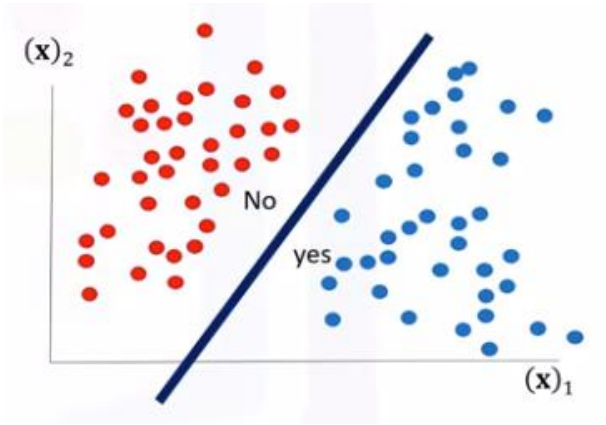


Figure 10: A logistic regression model used for predicting the category of input data. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

Logistic Regression applications in PV fault detection systems

An application of logistic regression applied to PV fault detection systems can be found in Jia Fan et al. [19]. The published detection system is able to identify the occurrence of an arc fault in the system based on four characteristic variables extracted from both the frequency domain and the time domain. The data represents the variation of the characteristic vector at a given instant. The data is collected in an experimental setting and used to train a logistic regression model. The authors highlight how the ability of logistic regression to return not only a class label (arc fault or normal state) but also the probability of such classes, is useful to decide which actions to perform.

6.2.3 Artificial Neural Networks

Artificial Neural Networks (ANNs), also known as connectionist systems, are a class of machine learning algorithms developed in the late 1950s that model the data structure and machine learning algorithms in a method similar to the method assumed to be used by the human brain. Figure 11 portrays the general algorithmic structure of ANNs containing an input layer, two hidden layers and an output layer. However, ANN algorithmic structures can have any number of hidden layers. The first layer contains the raw data provided in its most basic form. The hidden layers serve a variety of purposes to deconstruct the input data for analysis. The output layer then determines to which outcome the input data is most similar, based on patterns of the deconstructed data.

There are many variants of neural networks such as convolutional neural networks popularly used for image recognition purposes and long short-term memory network used for speech recognition.

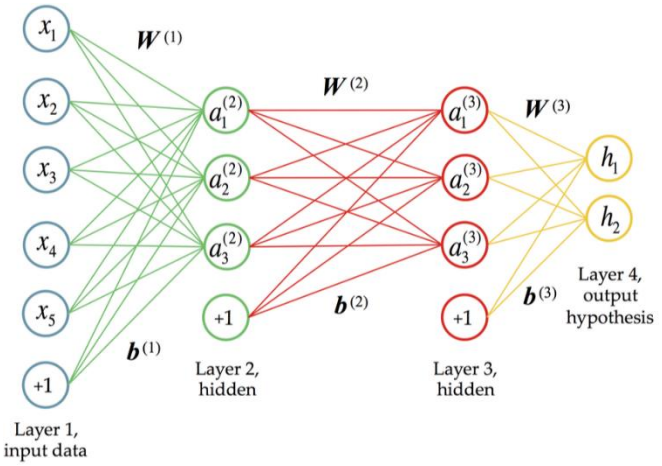


Figure 11: Illustration of a four-layer artificial neural network. Source: [20]

ANNs are becoming increasingly popular machine learning algorithms given their ability to train computers to implement complex tasks. Prevalent in cutting edge technologies, ANNs show up frequently in projects developing algorithms for PV fault detection applications.

6.2.4 Support Vector Machine algorithms

Support Vector Machines (SVMs) is a supervised classification algorithm that can identify patterns within data by finding a separator between different types of data. Once SVMs are applied to a dataset and patterns are identified, SVMs can be used to predict future outcomes by associating new observations with categorized patterns. The general steps applied for implementing SVM algorithms to a dataset are as follows:

1. Map the data available into a higher dimensional space so that it is easier to separate the data using hyperplanes.
2. Separate the data in the high dimensional space using hyperplanes.

To illustrate mapping data into a higher dimension, consider Figure 12 and Figure 13 below.

To transform data to a higher dimensional space we use a mathematical technique called kernelling by inputting the data into a kernel function. Kernel functions can be linear, polynomial, radial basis or a sigmoid function. The kernel function chosen depends on the characteristics of the dataset being classified. In some cases, different kernel functions are used for the classification process and then compared to determine which kernel function performs better. The illustration below illustrates mapping a one-dimensional dataset into a two-dimensional space. We see that after mapping the dataset into a higher dimensional space we can easily identify a separator as a linear hyperplane, illustrated by the black line separating the blue from red data points in the parabola.

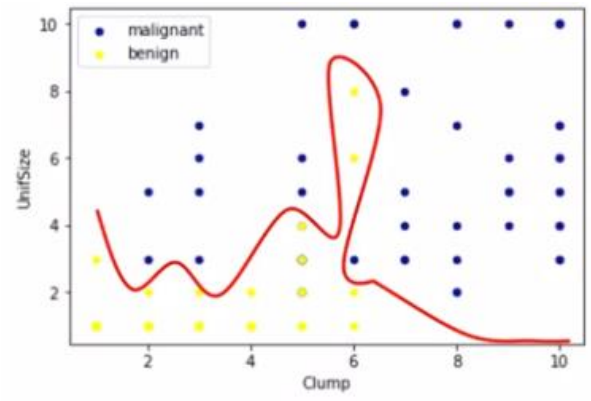


Figure 12: Data before being mapped in a higher dimensional space. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

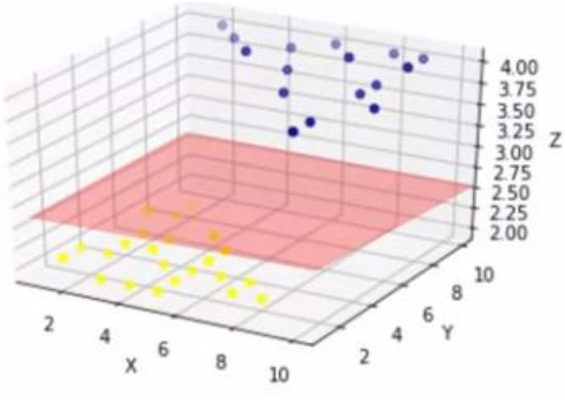


Figure 13: Data after being mapped to a higher dimensional space. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

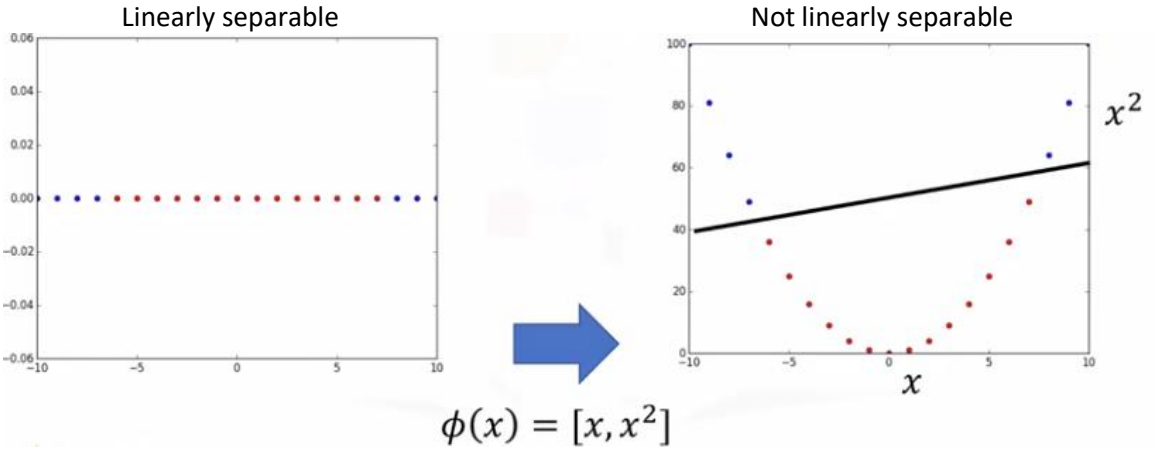


Figure 14: Illustration of a support vector algorithm. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

While the process of finding an optimal separator for classification, after mapping the data into a higher dimensional space, involves several considerations, and can be a complex process, there are a number of objectives in determining the hyperplane. The hyperplane should provide a maximum separation between the two classes; therefore, determining the optimal hyperplane is an optimization problem that may be solved using a variety of mathematical techniques such as gradient descent.

SVM provides several advantages in comparison to other machine learning classification techniques. Firstly, SVM may be highly accurate with high dimensional data. Secondly, SVM algorithms only use a subset of data points, known as support vectors, in order to classify new observations making the SVM algorithm memory efficient. Disadvantages of SVM are that the algorithm has a tendency to overfit if the number of data attributes is larger than the number of observations. In addition, the SVM algorithm does not provide probability estimations upon classifying, which may be a desirable feature. Also, SVM algorithms are not very

computationally efficient making it inefficient for large datasets (datasets with more than 1000 observations). [17]

SVM applications in PV fault detection systems

One example of using SVM for fault detection and diagnosis is given in Jiamin Sun et al. in their paper "Fault diagnosis model of photovoltaic array based on least squares support vector machine in bayesian framework" [30]. In this work the authors build a multiclass classifier that uses real output electrical parameters (open-circuit voltage, short-circuit current, maximum-power voltage and current) and parameters from the equivalent circuit representation of the PV array to distinguish between several states of the system: short circuit, open circuit, abnormal aging, normal state. Such a classifier acts both as fault detection, separating normal from faulty states, and as fault diagnosis, assigning the eventual fault to one of the classes mentioned above. The multiclass SVM classifier results from the aggregation of several two-class SVM classifiers: trained classifiers from pairs of classes. The class receiving the majority vote from this pool of classifiers is the one assigned to the observation being evaluated. The cited work also describes a tuning method of the SVM algorithm using a least square loss function and a radial basis function kernel.

6.2.5 Evaluating classification models

To compute the accuracy of classifiers, the dataset must be split into a training dataset and a testing dataset. While there are a variety of methods for evaluating classifiers, this paper will describe the Jaccard index, F1-score, and log loss classifier evaluation methods.

The Jaccard index, illustrated in Figure 15, computes the ratio of the number of correctly classified observations and the total number of observations tested. When the predictions are 100 percent accurate, the Jaccard index is one, and when there are no correct predictions the Jaccard index is 0.

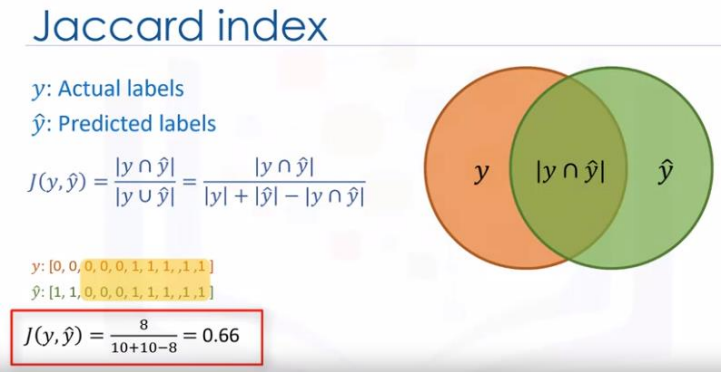


Figure 15: An illustration on applying Jaccard's Index for evaluating classification methods. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

A second method of evaluating classifiers is by using a confusion matrix, which is illustrated in Figure 16. The confusion matrix's rows show the actual labels while the columns show the predicted labels. The top left quadrant specifies the number of labels predicted as TRUE that are in fact TRUE and the data points in this quadrant are called true positives. The top right quadrant specifies the number of labels predicted as FALSE that were in fact TRUE. The data points in this quadrant are called false negatives. The bottom left quadrant specifies the number of data points that were classified as TRUE that were actually FALSE. The data points in this quadrant are called false positives. The bottom right quadrant specifies the number of data

points classified as FALSE that were actually FALSE and the data points in this quadrant are called true negatives. Summing the diagonal of the confusion matrix provides the total number of correctly predicted observations.

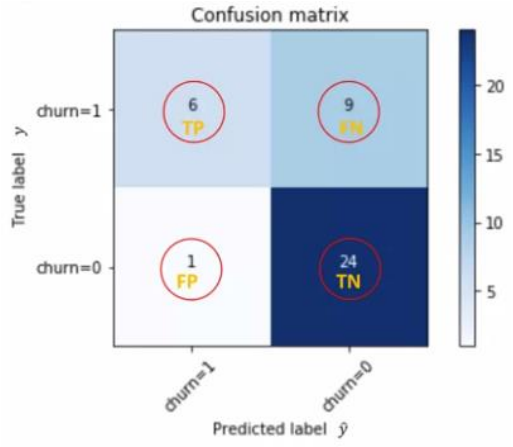


Figure 16: An illustration of a confusion matrix. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

Utilizing the confusion matrix, it is possible to compute the F1-score of the model, which is a measure of the accuracy of the classifier, as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (20)$$

When the classifier uses probabilities to classify an observation, a metric called log loss can be used to evaluate the accuracy of the classifier. The log loss of a given observation is given as follows,

$$LogLoss = -\frac{1}{n}\sum(y \times \log(\hat{y}) + (1-y) \times \log(1-\hat{y})) \quad (21)$$

where y is the actual value and $\hat{y}$ is the estimated value. The log loss value ranges between zero and one. The more accurate the classifier the smaller the log loss value.

Classification applications in PV fault detection systems

An example used for applying classification algorithms in identifying PV faults involves analyzing inverter data. Inverter data, typically collected from PV systems, includes AC power as well as a variety of other parameters such as AC voltage and AC current. A computer can be programmed to classify the AC power based on the variety of inverter parameters. Upon classifying the data, the computer learns, based on a constructed confidence interval, what the AC power should be depending on varying inverter parameters. When new data arrives at the inverter the classifier can identify if data falls within the expected confidence interval, and if the AC power deviates from its expected value how extreme the deviation is. When the AC power deviates overtime with larger and larger deviations from the confidence interval a fault can be predicted. Upon discovering what caused the fault, the machine learning classifier can learn how to categorize the type of fault that occurred [17].

6.3 Clustering

As stated in [21]: “A cluster is a group of data points or objects in a dataset that are similar to other objects in the group, and dissimilar to data points in other clusters.” Clustering is a method of segmenting groups of unlabeled data into categories based on characteristics they share. By partitioning data into mutually exclusive groups based on unique characteristics, clustering provides instructions for computers so that they can divide data into groups based on similar characteristics. In general, clustering can be used for a variety of purposes such as:

- To explore and analyze the data, to better understand it and gain insights about it
- To summarize data
- To identify outliers and remove noise
- To find duplicates in datasets
- To pre-process data before forecasting tasks, for data mining or before being input into additional algorithms

There are numerous types of clustering algorithms, some of which are:

- **Partitioned-based clustering algorithms,** which are relatively efficient and can be used on large datasets. Partitioned-based clustering algorithms include:
  - K-means algorithm
  - K-medians algorithm
  - Fuzzy c-Mean algorithm
- **Hierarchical clustering algorithms,** which create clusters by branching out the data in tree like structures, are relatively intuitive and are typically used on smaller datasets. Hierarchical clustering algorithms include:
  - Agglomerative algorithms
  - Divisive algorithms
- **Density based clustering algorithms,** which create arbitrary shaped clusters and are especially effective when analyzing spatial clusters and datasets containing noise.
  - Density-Based Spatial Clustering of Applications with Noise (DBSCAN algorithm)

6.3.1 K-Means clustering

k-Means, typically used on medium or large sized datasets, is a type of clustering algorithm that divides data into k non-overlapping (mutually exclusive) subsets (or clusters) used on unlabeled data based on similarities between different data attributes. Objects within clusters are similar. Objects belonging to different clusters are dissimilar. The way the k-Means algorithm partitions data is by computing the dissimilarity between different observations and then grouping data based on how dissimilar they are. There are a variety of ways of computing dissimilarity of observations in a dataset. A simple yet popular method for computing dissimilarities is by applying Euclidean distances. The dissimilarity between two observations is computed as shown in Figure 17.

Customer 1		
Age	Income	education
54	190	3

Customer 2		
Age	Income	education
50	200	8

$$\mathrm{Dis}(x_1, x_2) = \sqrt{\sum_{i=0}^{n} (x_{1i} - x_{2i})^2}$$
$$= \sqrt{(54-50)^2 + (190-200)^2 + (3-8)^2} = 11.87$$

Figure 17: Example of computing the Euclidean distance. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

In addition to Euclidean distances, there are also other methods of computing dissimilarity between observations, such as cosine similarity and average distance.

The general algorithm used for k-Means clustering is:

1. Choose a value for k which specifies the number of clusters and centroids in a dataset. Centroids are placed in the dataset; a variety of methods are used for determining where the centroids should be initialized in the dataset. For example, one method involves choosing k random data points in the dataset.
2. For each observation in the dataset compute the distance between the observation and the k centroids. Consequently a "distance matrix" is created specifying, for every observation, the values of the distances from the k centroids.

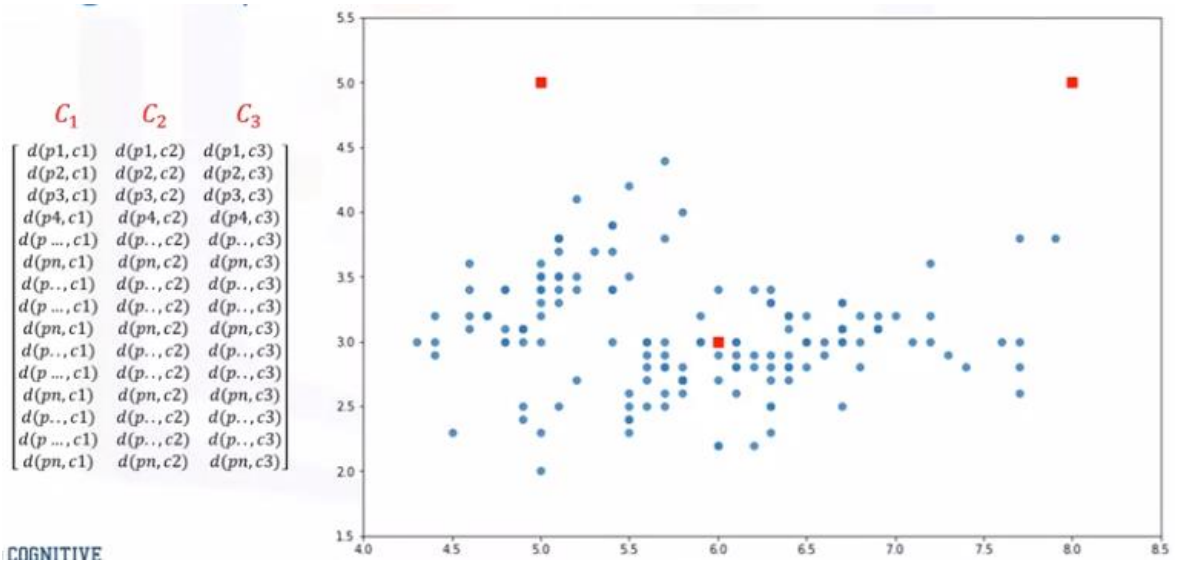


Figure 18: An illustration for computing the k-means clustering method. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

3. For each data point assign the centroid closest to that data point.
4. Evaluate how effectively the data was clustered by applying the sum of the squared differences between each point and its centroid for each cluster according to

$$SSE = \sum_{1}^{n} \left(x_i - C_j\right)^2 \qquad (21)$$

5. Compute the mean of the data points in the cluster.
6. Update the centroid center to be the mean of each cluster.
7. Step 2 to step 6 are iteratively implemented until the centroid reaches a minimal value error resulting in the densest clusters.

The resulting algorithm may only provide a locally optimal cluster and is not guaranteed to be a global optimum cluster. Therefore, when implementing the k-Means algorithm, the algorithm should be run on multiple initial points in hopes of finding the global optimum.

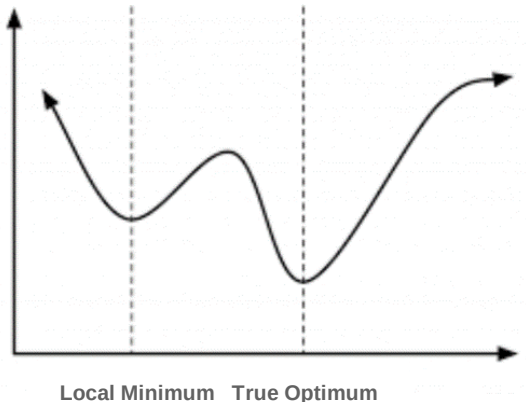


Figure 19: Illustration for contrasting local minimum with the global minimum. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

Choosing an optimal k value is a hard and frequent problem in large datasets since it depends on the initial points used when running the k-Means algorithm. Furthermore, since the shape and scale of the dataset may be ambiguous it may not be possible to know if optimum points are in fact the global optima. [17]

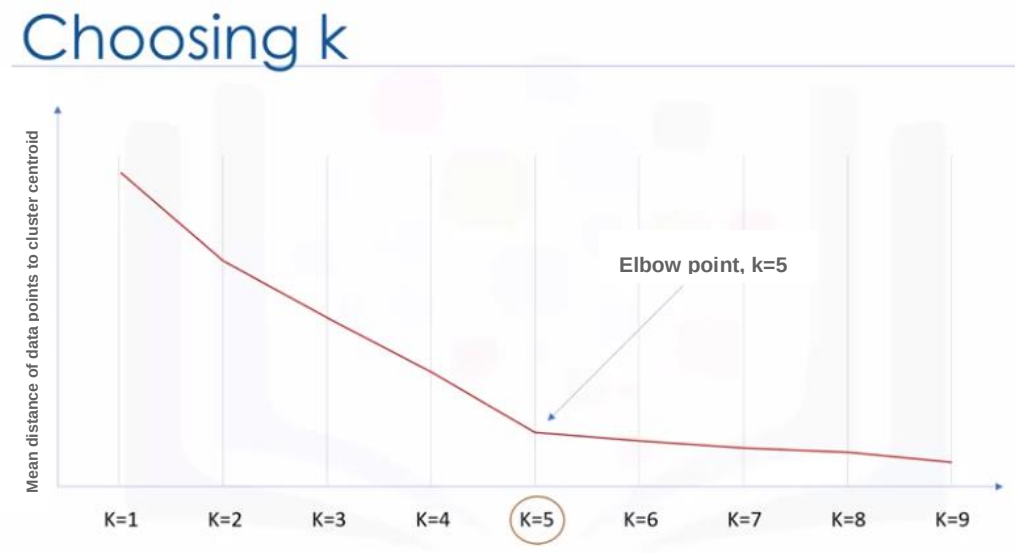


Figure 20: Choosing the correct value of k for the k-means clustering method. Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation.

One method for identifying the optimum k value is called the elbow method. First, one computes the average distance between all data points in a cluster and the centroid for several values of k. Since the average distance between data points and the centroid decreases with the number of centroids applied in the k-Means algorithm, the optimum number of centroids is identified by the elbow point in the graph shown in Figure 20. [17]

6.3.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering

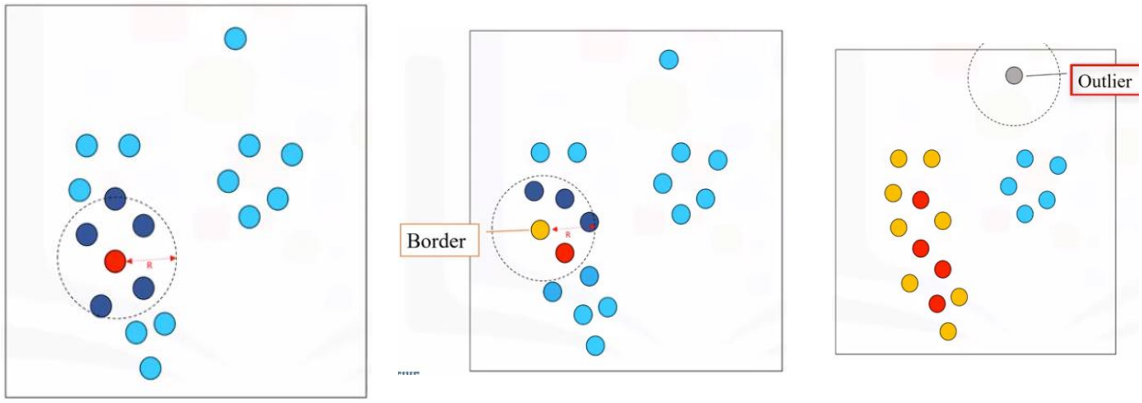
Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm typically used when analyzing spatial data, meaning data represented by numerical values based on a geographic coordinate system. DBSCAN identifies regions of high density that are separated from regions of low density; density is defined as the number of points within a specified radius. DBSCAN clusters the data points according to object density. In creating clusters, DBSCAN evaluates two parameters: radius of a neighborhood, notated R, and the minimum number of neighbors, notated M. A cluster is created when the density of the neighborhood is maximized for a given radius R, for a minimum number of neighbors M, specified by the programmer. The general algorithm outline is as follows:

1. Label each point in the algorithm as a core point, border point or outlier point.
2. Group all core points that are neighbors, and their neighbors, are grouped into a cluster.

Types of DBSCAN groupings

Core point	Border point	Outlier point
Points that, for a sphere centered at the point, with specified radius, R, and minimum number of neighbors, M, contains M neighbors.	A data point that, when drawing a sphere of radius R centered at the point, the sphere contains less than M data points or the sphere contains a core point.	An outlier point is a data point that is not a core point and cannot be reachable by a core point.

For R=2 and M=6



DBSCAN has a number of algorithmic advantages in comparison to other clustering algorithms, including an ability to exclude outliers when synthesizing clusters, an ability to generate clusters within clusters, an ability to generate arbitrary shaped clusters and it can automatically

optimize the number of clusters based on a specified radius and a specified minimum number of neighbors. [17]

6.4 Other machine learning algorithms

Other machine learning algorithms include the ensemble method that combines multiple weak machine learning models to generate a robust machine learning algorithm. A model is called weak when it could perform only slightly better than random guessing. Many applications of the ensemble method exist; some essential examples are boosting, bagging, and stumping.

Ensemble learning arises from the idea that it is possible to improve the performance of a given prediction task by combining predictions of different, usually simple, models. One way to quantify the improvement in predictions using the ensemble method is to look at the bias-variance trade-off: by decomposing the prediction error of a model into the bias and variance components. The bias represents the deviation between the model and the real function to be predicted, while the variance represents the sensitivity of the model to the individual data points. Averaging the predictions of several low-bias, high-variance models can improve the model error, lowering the variance component without affecting the bias.

To obtain different predictive models, different datasets related to the same phenomenon are needed. For cases where different datasets are not available, variability is introduced artificially by applying techniques such as bootstrapping. Bootstrapping allows the original data to be used for building several, slightly different, datasets. It is possible to then train several different models on these different datasets. Using bootstrapping for generating multiple datasets for machine learning purposes is called bagging.

7 COMPARISON OF DATA SOURCES AND TRAINING STRATEGIES

The content of this chapter first appeared as a paper [22] *in the 37th European Photovoltaic Solar Energy Conference and Exhibition.*

7.1 Introduction

The previous chapters presented and explained several methods of fault detection using statistical methods including Machine Learning (ML). Those methods are being applied by developers of failure monitoring algorithms for photovoltaic systems. In this chapter an attempt will be made to analyze the qualitative effect of different types of input data and the comparative veracity of a number of ML algorithms.

To this end we consider one specific fault, typical to a number of algorithmic approaches to fault detection, that is estimating a PV system's output energy, with no attempt to understand the underlying fault or reason for the lower-than-expected energy production.

In several fault detection algorithms, the system's measured output power is compared to its estimate. Then the difference between the two quantities is computed. If this difference is above a given threshold, we consider the PV system to be faulty.

There are several ways in which to estimate the output power: using physical and empirical models is one option. Another approach is to use ML algorithms that can learn the model of a PV system.

Most ML algorithms, because of their flexibility in the data sources used as input, are very good for analyzing PV systems. In addition to environmental data (such as temperature and irradiance) used in physical modeling, ML algorithms can exploit the statistical dependencies in output power data collected from nearby PV systems, which are supposed to be affected by the same external conditions. Therefore, we expect a high correlation among them.

This possibility is particularly relevant since PV systems' output power data are cheaper for a PV fleet owner to obtain than environmental data, which require dedicated sensors or collection from third-party systems. Though for individual system owners, the use of neighboring system data sets may not be practical.

This chapter presents a case study where the performance of a number of ML algorithms and both types of data sets are compared side by side on the same PV site. The exercise will shed light on the question of how much the performance of a ML algorithm is affected by choosing a specific input data source, in this case, environmental data, or power data from nearby systems.

A variety of ML algorithms have been trained to estimate the output power of a PV system using the two types of data sources. Then, they have been tested, and their objective performance is compared.

Besides comparing the different data sources, the ML algorithms' training strategies are also compared. Given the data's temporal nature, the training strategy's choice might influence the evaluation of the algorithm's performance.

7.2 Details of the comparison

This section presents the data and the training strategies that were compared. We briefly describe the dataset used and how it fits the comparison.

7.2.1 Data

The input data used by ML algorithms to estimate the output power of a system falls into two main categories: environmental data and output power data from nearby systems. The use of temperature and irradiance to model the output power is well known from the physical modeling; ML algorithms' use makes it possible to model this dependency efficiently without additional information about the system under investigation, only historical data.

In addition to this possibility, the flexibility of ML algorithms enables the use of performance data from nearby systems as well. PV systems that are close to each other are assumed to be affected by the same environmental conditions. Therefore, a high correlation among the signals coming from different systems is expected. These high correlations make it possible to estimate a given system's power output, knowing how nearby systems are performing.

To perform a meaningful comparison, a complete data set containing both types of data was required. For this purpose, the data set used was acquired from the American National Institute for Standards and Technology (NIST) [23].

The data set contains data from three PV installations installed at the NIST campus in Gaithersburg Maryland, USA, and a weather station. The three PV installations measure both power data and environmental data. Therefore, the environmental data is available both from the campus weather station and dedicated onsite sensors.

The distances between the weather station and the PV installations and between the different PV installations vary between 300 and 700 meters; Figure 21 shows the PV installations' specific location and the campus weather station (the vertical edge of the figure corresponds to 1 kilometer).

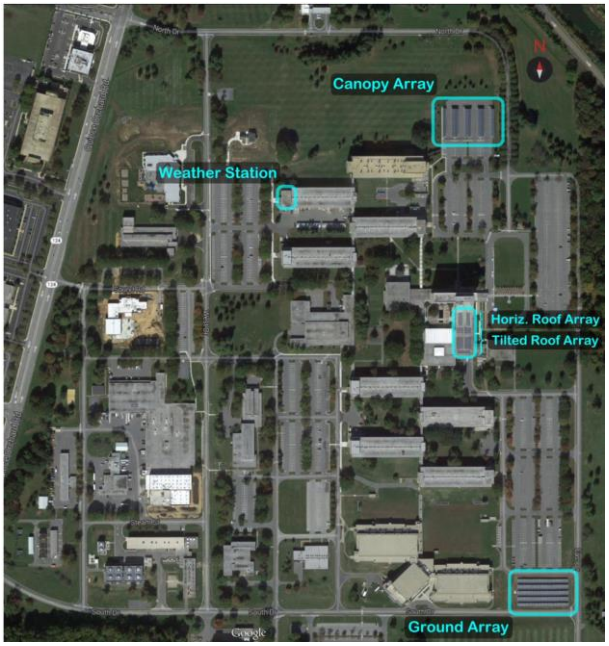


Figure 21: satellite map of the 4 PV system locations

The availability of environmental data from both the campus and the onsite weather stations allows a more detailed comparison, taking into account the measuring site's distance from the installation of interest.

7.2.2 Training strategies

As well as the effectiveness of the datasets, different training strategies for the ML algorithms are also tested; the data's temporal nature implies dependencies that can be exploited to make the algorithms perform better

Three different training strategies were tested:

- Random split: the available data is randomly split between a train and a test set, not considering the data's temporal structure. This training strategy mimics what can be achieved with simulations when several combinations of input data can be generated and used for training.
- One-time training: the available data is split temporally: the first part of the data is used as training, while the rest is used for testing. For example, on a data set corresponding to one year, the first three months might be used as training data and the remaining nine months for testing.
- Periodic training: not all the data is used at once; for each week of data used for testing, the previous four weeks are used as training data. This strategy is based on the observation that today's environmental conditions are more likely to be similar to that of the last month than three months ago. Therefore, the noise in the data should be reduced by this choice of training data.

Figure 22 gives a visual representation of the different training strategies explained above.

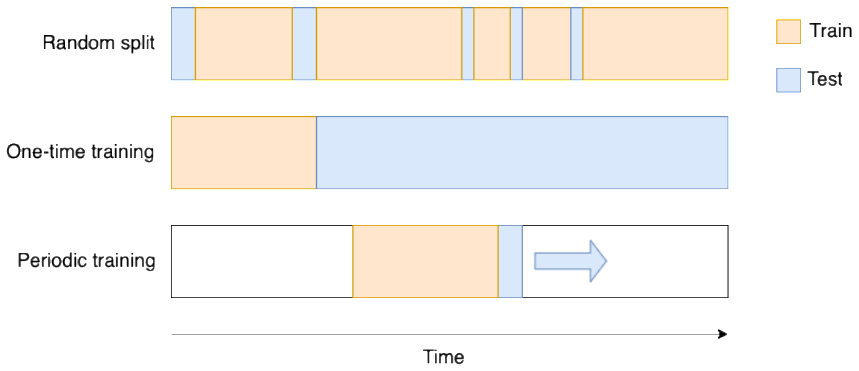


Figure 22: Visual representation of different training strategies.

7.2.3 Comparison

To provide general results that did not depend on a specific ML algorithm, five algorithms were tested, specifically linear regression, decision trees, random forest, k nearest neighbors, and neural networks.

The actual comparison consists of training each of the ML algorithms with all possible combinations of training strategies and data sources, computing a performance metric, and then comparing the results through visualization of the error distributions.

The chosen performance metric is the root mean squared error normalized to the given installation's nominal power (*nRMSE*): the normalization ensures that it is possible to compare data from installations with different sizes. The equation of the *nRMSE* is:

$$nRMSE = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{true} - y_{est})^2}}{y_{nom}} \quad (22)$$

where y_{true} is the true output power, y_{est} is the estimated output power, and y_{nom} is the nominal power of the PV installation; n spans the data points throughout the period of interest, one day in the analysis.

7.3 Results

Each point in the visualizations of this section refers to the daily nRMSE. The results are aggregated at different levels to give a complete overview.

7.3.1 Effect of the training strategy

Figure 23 shows the global effect of the training strategy on the performance of estimating the PV output power. The data is aggregated from all PV installations and all ML algorithms. For all three data sources, it can be seen that the periodic training strategy returns the error distribution with lower values; the hypothesis that more recent data contain less noise seems to be confirmed in this case.

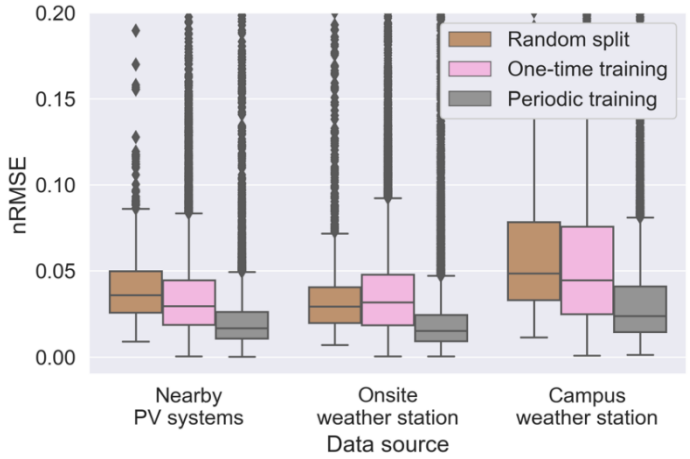


Figure 23: Effect of training strategy on the performance. Data aggregated from all ML algorithms and all PV installations.

For the following visualizations, only results obtained with the periodic training strategy are included.

7.3.2 Effect of the input data source

Figure 24 shows the performance of each ML algorithm, when trained using the periodic training strategy and each of the available data sources. For every data source, the trend for each algorithm is the same: the data from the campus weather station returns an error distribution slightly skewed towards higher errors; data from onsite weather stations or from nearby power systems result essentially in the same performance.

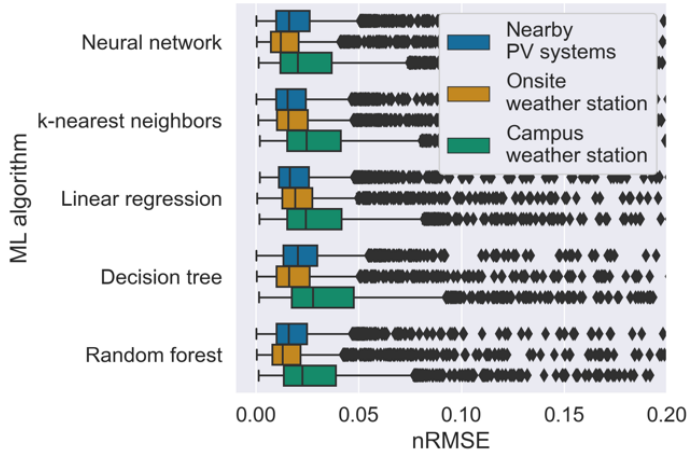


Figure 24: Effect of ML algorithm's choice on the performance. Data aggregated from all sites; algorithms trained with the periodic training strategy.

Figure 25 contains the same data as Figure 24, but rearranged to show that there is no significant difference in the way each algorithm extracts information from the input data, resulting in very similar performance.

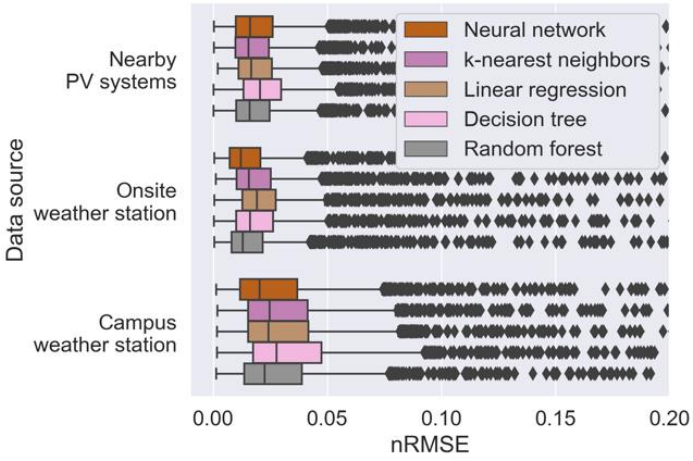


Figure 25: Same data as Figure 24, but rearranged to compare the performance of different ML algorithms.

In all previous visualizations, the performance data from all sites were aggregated. Here they are divided, considering only a single ML algorithm, linear regression. Figure 26 shows the performance of training linear regression with the periodic training strategy, for each PV installation and each input data source. It is apparent that there is no clear trend: for the *Canopy* installation, all input data sources perform similarly, while for the *Ground* installation the data from the campus weather station performs slightly worse than the other two.

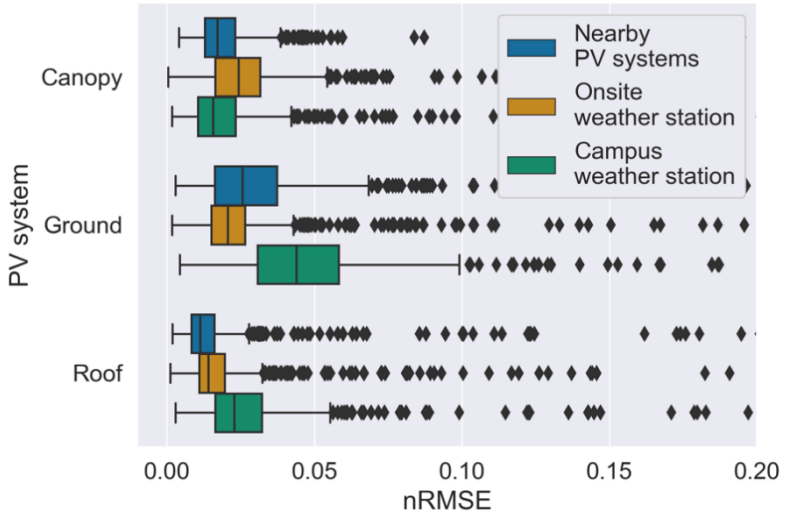


Figure 26: Performance of linear regression algorithm, using periodic training, for each PV installation.

7.4 Conclusions

The visualizations presented in the last section does not allow for a given data source to be assumed as always better than the others in estimating the output power of the PV system. Although the effect of the training strategy seems more evident, the impact of the choices of the ML algorithm and of the data source are less clear. What can be said with more confidence, at least in this case, is that power data from nearby systems is, in general, at least as informative as environmental data, collected both onsite and from the campus weather station. The lower cost in collecting power data, compared to environmental one, makes such result interesting for PV monitoring, especially when considering residential installations, where weather stations are almost always absent.

8 OVERVIEW OF CURRENT PUBLICATIONS ON PHOTOVOLTAICS FAULT DETECTION SYSTEMS

This chapter summarizes 22 of the research papers studied for this report on the use of advanced algorithms in PV failure monitoring.

8.1 Real-time fault detection in massive multi-array PV plants based on machine learning techniques

Title of Paper: Real-time fault detection in massive multi-array PV plants based on machine learning techniques [24]

Year of publication: 2019

Authors: Hsu Chung-Chian, Li Jia-Long, Chen Yu-Sheng

Institutions: National Yunlin University of Science and Technology

Countries: Taiwan

Parameters: irradiance, power, nominal power, current ratio, voltage ratio, other…

Data: previous three months of historical data

Data resolution: power and irradiance data sampled every 5 minutes.

Data filters: filtering for irradiance if lower than 250 W/m^2 and abrupt changes in data

Algorithms: Linear regression, K Nearest Neighbors, rule-based system

Statistical tests/models: array ratio (estimated by using a nonlinear regression algorithm from its own data collected in the past 3 months)

Hardware: pyrheliometer for recording solar irradiance

Types of faults: partial shading, inverter fault, inverter late boot, fuse blown, inverter overheated, string open circuit and other faults

Description of system: Monitoring 150 systems with a cumulative power rating of 7kWp.

Advantages: only irradiance and power are required for detection, ability to provide real time detection.

Authors Summary: "We propose an approach based on machine learning techniques which analyze historical power and irradiance data of PV arrays and do not require additional sensors. The developed system currently monitors 150 plants including 7028 arrays is deployed on a Spark-cluster distributed platform such that the detection and diagnosis process can be finished within 5 minutes. Three months of historical power and irradiance data are used to obtain a range of valid Array Ratio (AR) via linear regression. If the AR falls outside the normal range for more than n consecutive times, the system detects a fault.

If the plant is new and there is no historical data, a KNN approach is used to estimate the power generated from the measured irradiance and power data from the last few days. A rule-based system diagnoses the fault detected."

8.2 Automatic fault detection of photovoltaic array by convolutional neural networks during aerial infrared thermography

Title of Paper: Automatic fault detection of photovoltaic array by convolutional neural networks during aerial infrared thermography [3]

Year of publication: 2019

Authors: Vidal de Oliveira Aline Kirsten, Aghaei Mohammedreza, Rüther Ricardo

Institutions: Universidade Federal de Santa Catarina, Fraunhofer Institute for Solar Energy Systems

Countries: Brazil, Germany

Parameters: Video frames from aIRT camera

Data: aIRT video frames

Data resolution: The IRT Camera has a spectral response range between 7 and 17 µm, and its resolution is 640 pixels. The captured videos have a frame rate of 60 FPS. The dataset consists of frames of the videos that are recorded in grayscale and the intensity differences represent the temperature distribution on the modules.

Data filters: A Gaussian filter is chosen to decrease noise, highlighting the boundaries of PV modules and facilitating the segmentation part. The contrast is adjusted in order to highlight the PV modules borders for the next steps. An additional fisheye reduction filter is applied in order to minimize the distortion of the camera.

Algorithms: Convolutional Neural Networks, Digital Image Processing

Hardware: Aerial Infrared Thermography (aIRT) camera (and drone for large systems)

Types of faults: hot spots, disconnected strings, disconnected substrings

Description of system: Images were taken during the inspection of a 37 MWp PV power plant by UAV-based Aerial IRT measurement system, covering a 97 hectares area in the Northeast of Brazil. The PV plant consists of more than one hundred N-S single-axis trackers.

Stage of development: academic research study

Authors Summary: "This paper proposes a method for detecting and classifying faults on PV modules, through aerial IRT images, combining Digital Image Processing (DIP) and Convolutional Neural Networks algorithms (CNNs). The IR images acquired are processed with DIP techniques to detect the faults of PV modules in the power plant that are used as samples for training the CNNs. The developed neural network algorithm can detect faults on the aIRT images and classify them in three categories: disconnected substrings, hot spots, and disconnected strings."

8.3 PV O&M optimization by AI practice

Title of Paper: PV O&M Optimization by AI Practice [25]

Year of publication: 2019

Authors: Chang Maoyi, Hsu Chung-Chian, Chen K.H., Hsu T. P., Wei Kyle, Chuang Ken, Chen Yu-Sheng

Institutions: Sinogreenergy, National Yunlin University of Science and Technology

Countries: China

Parameters: solar irradiation, installed capacity, power, voltage, current, Vmpp, Impp

Data: 7 days of data used for training. To assess the AI system capability, they implemented the system at 152 project sites with eleven inverter and nine module suppliers up to 54MW located from central to southern Taiwan. 7,583 MPPT data processing completed every 5 minutes, and there are at least 6,369,720 batch data input for model training for all project sites.

Data resolution: The system receives data for each MPPT in inverters every five minutes with in-situ analysis.

Algorithms: convolutional Neural Networks, K Nearest Neighbors, non-linear regression, Long-short-term-memory

Hardware: pyranometer, data logger to access inverter data

Types of faults: Inverter faults, clogged inverter fans, shading, burnt fuses, string problems, communication error, other.

Description of system: 150 project sites up to 54 MW with 11 inverter brands and 9 module suppliers

Stage of development: Monitoring more than 150 project sites up to 54 MW

Authors Summary: Without specific module, inverter and location parameters as inputs, the power prediction model for each inverter-MPPT is trained based on its own historical production data and established as its own fingerprint. Each inverter-MPPT behavior from all the projects (over 7,583 MPPTs) is monitored and analyzed by machine learning every five minutes. The output power predicted is compared to that measured, if the production is below a certain percentage threshold, an alarm is issued, after diagnosing the type of fault with unspecified algorithm. Fault detection alert with failure mode is automatically judged, and prompt notification is sent to user by mobile device or email.

Advantages: Except for MPPT data (power, current and voltage) the only parameter needed to input is the solar irradiation

8.4 Real time fault detection in photovoltaic systems

Title of Paper: Real Time Fault Detection in Photovoltaic Systems [26]

Year of publication: 2016

Authors: Mohamed Hassan Ali, Abdelhamid Rabhi, Ahmed El Hajjaji, Giuseppe M. Tina

Institutions: University of Catania DIEEI laboratory

Countries: Italy

Parameters: IV curve point of Isc, IV curve point of Voc, slope between the short-circuit point and the maximum point, slope between the maximum and the open circuit point, variation of series resistance, Solar module nominal power Pmax, Solar module Vmpp, solar module Impp, Solar module Voc, solar module Isc, Pmin, solar module efficiency, solar module temperature factors, number of solar module cells.

Data: Real data from an experimental photovoltaic generator used by three Polycrystalline PV modules.

Programming languages: Matlab Simulink, other

Algorithms: Fractional Order Darwinian Particle Swarm Optimization

Hardware: IV curves sensor (e.g., Pordis 140a tracer)

Types of faults: Partial shading, interconnection resistance. Specific faults analyzed: Condition where a third part of a module shaded only, Condition where 3 cells of each module are shaded, Condition where half part of each module are shaded, a line resistance connected having value RC=3Ω which represent resistive losses on connections, a line resistance connected having value RC=5Ω which represent resistive losses on connections

Description of system: The experimental photovoltaic generator used in this study is a string formed by three Polycrystalline PV modules, CLS-220P by CHINALIGHT Solar Co, connected in series. Each module contains 60 series connected PV cells gathered into three sub-strings, each one is constituted by 20 PV cells and connected in parallel with a bypass diode. The experimental setup (PV string; electronic load and Model China Light Solar CLS 220P Electrical Data Nominal power PMAX [W] 220 Maximum voltage VMPP [V] 28.9 Maximum current IMPP [A] 7.61 Open circuit voltage Voc [V] 36.8 Open circuit current Isc [A] 8.24 Minimum power guarantied PMIN [W] 220 Output efficiency [%] 13.5 Maximum voltage of system [VDC] 1000 Temperature factor of PN [%/°C] -0.0044 Temperature factor of VDC [V/°C] -0.0032 Temperature factor of Isc [mA/°C] 0.0004 TNOCT [°C] 47 Number of cells 60. System is installed in the power system laboratory at the DIEEI Department of University of Catania (Italy)

Simulation model: two diode solar module model

Stage of development: Research experiments in a laboratory setting on three solar modules connected in series.

Authors Summary: The passive part of diagnosis involves comparing in real time the measured power and simulated power generated by the fault detection system model. The diagnosis strategy is to measure voltage and current in real time and calculate the produced power by PV system. The captured data is compared with the simulation results. The fault detection will be determined by fixing a normal threshold and a failure threshold based on the comparison of the simulated and real data. Each value of residue is generated using the diagnosis method based on the model.

Advantages: Sophisticated and innovative method of identifying, analyzing and categorizing faults. May be a very useful tool for researchers studying PV faults in a laboratory setting to identify signature faults and rules for identifying faults in the field

8.5 A statistical tool to detect and locate abnormal operating conditions in photovoltaic systems

Title of Paper: A Statistical Tool to Detect and Locate Abnormal Operating Conditions in Photovoltaic Systems [14]

Year of publication: 2018

Authors: Silvano Vergura

Institutions: Polytechnic University of Bari, Italy

Countries: Italy

Parameters: DC current, DC voltage, AC current, AC voltage, AC power, AC energy

Data: Three applications of the fault detection methodology are discussed: the first one, based on the energy dataset of one month; the second one, based on the energy dataset of six months; the last one, based on the energy dataset of one year. The energy performance of a

real operating 90 kWp grid-connected PV plant, installed in a private area of a company located in Bari, a city in the south of Italy, has been studied.

Data resolution: The datalogger has a sample time of 2 seconds. Internal software calculates the mean of all the measures after 10 min and only then is stored in a database and used by the fault detection system.

Programming languages: Matlab Simulink (for simulation), other

Statistical tests/models: Hartigan's dip test used to determine uni or multimodal distributions to determine if parametric test is possible. ANOVA, Kruskal Wallis test, Mood's median test, homo-scedacity test.

Hardware: Hardware necessary for collecting inverter data (e.g., data logger).

Types of faults: low-intensity anomalies (degradation)

Description of system: 90 kWp grid-connected PV plant, installed in a private area of a company located in Bari, a city in the south of Italy.

Stage of development: academic study

Authors Summary: The main idea is to compare the statistical distributions of the energy of the arrays. For small-medium-size photovoltaic plants, it is assumed that the environmental conditions affect equally all the arrays, so the comparative procedure is independent of the solar irradiation and the cell temperature; therefore, it can also be applied to a photovoltaic plant not equipped with a weather station. If the procedure is iterated and new energy data are added at each new run, the analysis becomes cumulative and allows following the trend of some benchmarks. The methodology is based on an algorithm, which suggests to the user, step by step, the suitable statistical tool to use. The proposed methodology is devoted to the small-medium-size PV plants, constituted of several arrays, and does not require environmental data such as solar irradiance or cell temperature. The fault detection system analyses the dataset of the energy produced by each array and extracts the features of their statistical distributions, in order to choose the best performing statistical tool to use. Depending on the modality (unique or multiple) of the distributions and on other statistical parameters, a parametric or a non-parametric test is used, to evaluate whether identical arrays, in the same unknown environmental conditions, produce the same energy. The monitoring of the statistical parameters and of their mismatches with respect to the benchmarks allows detecting and locating possible anomalies, before they become failures.

Advantages: Weather sensors unnecessary. As new data is acquired the system becomes more accurate, allowing for the estimation and location of low-intensity anomalies in modules before they affect neighboring modules as may occur with hotspots.

8.6 General, robust and scalable methods for string level monitoring in utility scale PV systems

Title of Paper: General, robust and scalable methods for string level monitoring in utility scale PV systems [11]

Year of publication: 2016

Authors: Skomedal Asmund, Ogaard Mari B., Selj Josefin H., Haug Halvard, Marstein Erik S.

Institutions: University of Oslo

Countries: Norway

Parameters: Energy yield of PV system

Data: Daily data from three MW-scale PV plants located in Sub-Saharan Africa, the Middle East and Northern Europe.

Data resolution: Can be any resolution. Authors use daily energy yield.

Data filters:

1. A minimum plane-of-array irradiance (Gi) – ensuring minimal differences in low irradiance losses
2. A minimum solar elevation angle – ensuring minimal shading and self-shading, and minimal differences in air mass (AM), ensuring minimal differences in spectral mismatch
3. A maximal angle of incidence (AOI) – ensuring minimal performance differences due to module orientation
4. A filter for clear sky conditions – ensuring uniform irradiance over the plant, and minimal spectral differences

Statistical tests/models: ANOVA,

Hardware: Data logger

Types of faults: Health of system at any PV system resolution including module, string, or string-combiner of any size depending on available energy for data for a given PV array.

Description of system: The test plants are located in three locations: one in Sub-Saharan Africa (L1), one in the Middle East (L2), and one in Northern Europe (L3). L1 and L2 are ground mounted, while L3 is roof-mounted. L1 and L3 are fixed tilt while L2 is a single-axis tracker system.

Stage of development: academic study

Authors Summary: The algorithm uses production data as input, filters out unwanted datapoints and calculates a performance metric with a pre-defined frequency, and uses this metric to evaluate the performance of different sub-arrays. A main contribution of this work is the proposal of a procedure for selecting filtering thresholds to reduce noise in the performance metric. The paper shows that by applying suitable filters the sensitivity of the fault detection algorithm is increased 2 – 5 times; thereby, greatly improving the robustness of the algorithm. The paper differentiates between hard and soft faults. Hard faults are those that lead to a sudden loss in the performance, usually because a whole section of the system is down due to a critical component (blown fuses, broken cables, etc.). Soft faults are faults that lead to slow changes in the performance (cell cracks, corrosion, discoloration, Potential Induced Degradation, soiling, partial shading, etc.), and often cause smaller reductions in the performance of the PV system. For this reason, soft faults are usually detected through visual inspection or thermal imaging. One of the main goals of this work is to enable more robust data-driven detection of soft faults.

In this paper, a procedure for making a fault detection algorithm based on the calculation of a statistics-based performance metric is proposed. The method simply considers the relative differences in all the units' energy output. In this way the system is relying on statistics to give us a reference for comparison, rather than a model. In other words, the fault detection system is looking for relative differences between the units, and the prevailing environmental factors are implicitly accounted for.

Advantages: This form of performance assessment has the advantage of being insensitive to sensor drift and missing sensor data. Numerically efficient than more advanced approaches involving modelling.

8.7 SolarClique: detecting anomalies in residential solar arrays

Title of Paper: SolarClique: Detecting Anomalies in Residential Solar Arrays [10]

Year of publication: 2018

Authors: Srinivasan Iyengar, Stephen Lee, Daniel Sheldon, Prashant Shenoy

Institutions: University of Massachusetts Amherst

Countries: USA

Parameters: solar power of PV site being monitored and at least five neighboring sites.

Data: public datasets available through the Dataport Research Program at an hourly granularity. 88 homes for our evaluation in the year 2014 and 2015. The first three months of data to train the model, and the remaining 21 months of data for testing the model.

Data resolution: Hourly energy production values

Programming languages: Python SciPy stack, Python scikit-learn

Algorithms: half-sibling regression, ensemble method

Statistical tests/models: Bootstrapping, Seasonal and Trend decomposition using Loess (STL) technique. For bootstrapping, training data is sampled by randomly selecting 80% of the training samples with replacement. These samples are then used to build an estimator, and this process is repeated 100 times to learn the properties of the estimator.

Hardware: not used

Types of faults: Identifies energy loss due to long term malfunctioning not related to weather or shading. Anomaly category types: Single system no production, multiple system no production, Single system under production, multiple system under production, severe degradation

Mean Absolute Percentage Error (MAPE)

Description of system: data from 88 solar installations between 0.5 to 9.3 kW. Residential size (sq. ft.) 1142 to 3959

Stage of development: research study

Authors Summary: Inspired by a study in astronomy for removing noise from measuring instruments, SolarClique identifies faults in PV systems by comparing a monitored sites power and energy data with at least five neighboring PV sites. SolarClique's fault detection innovation is in its approach in determining faults: utilizing neighboring PV sites to reveal expected power behavior for monitored PV sites. Implementing this realization, SolarClique avoids the need of obtaining and analyzing large and complex weather data to predict expected energy production to identify discrepancies between expected and produced energy. Instead SolarClique applies a variety of machine learning algorithms correlating neighboring PV sites performance with the monitored PV sites performance.

Advantages: Robust enough to distinguish between reduction in power output due to anomalies and other factors such as cloudy conditions. Simple and inexpensive to implement on many rooftop systems.

8.8 Statistics to detect low-intensity anomalies in PV systems

Title of Paper: Statistics to Detect Low-Intensity Anomalies in PV Systems [27]

Year of publication: 2018

Authors: Silvano Vergura, Mario Carpentieri

Institutions: Polytechnic University of Bari

Countries: Italy

Parameters: Energy production

Data: Energy data of several arrays belonging to the same plant. Data covers a full year with monthly analysis (January), quarterly analysis (January–March) and yearly analysis (January–December).

Data resolution: The PV plant has a data acquisition system, constituted by a datalogger that acquires the data from the six inverters at a 2 second frequency. An internal software calculates the average value of the sampled data each 10 min and stores only this value into the database.

Data filters: Unknown

Programming languages: Matlab

Algorithms: Statistical based

Statistical tests/models: ANOVA, Kruskal-Wallis test, Mood's median test, homoscedasticity's test, normal probability test

Hardware: Data logger

Types of faults: low-intensity anomalies (no diagnosis of the type of fault)

Description of system: 9.8 kWp grid-connected PV plant, located in the South of Italy. The 132 modules of the plant are partitioned in 6 equal arrays. Each PV module has a nominal power of 150 Wp, so the peak power of each array is 3300 Wp.

Stage of development: research phase. tested on real plants using real data.

Authors Summary: using statistical tests to detect low-intensity anomalies before they become actual failures. Use of ANOVA and non-parametric tests. Cumulative analysis on 12 months of data. Examination of p-values, skewness and other statistical values eventually point to possible low intensity anomalies. The least performing array is checked when the statistical indicators say that there's a problem.

Advantages: no hardware need besides a datalogger for sending data to a private server for analysis.

8.9 Automatic fault detection in grid connected PV systems

Title of Paper: Automatic fault detection in grid connected PV systems [27]

Year of publication: 2013

Authors: Silvestre Santiago, Chouder Aissa, Karatepe Engin

Institutions: Universitat Politecnica de Catalunya (UPC) BarcelonaTech

Countries: Algeria

Parameters: Energy, voltage (AC and DC), irradiance, current (AC and DC), temperature

Data: unknown

Data resolution: unknown

Data filters: unknown

Programming languages: Lab View

Algorithms: unknown

Statistical tests/models: unknown

Hardware: pyranometers, reference cell, thermocouple, transformer, measurement of DC voltage and AC voltage performed by a resistive voltage divider and AC transformer in order to adapt voltage levels to the input of data acquisition respectively. While the output PV plant, DC current and the output inverter AC current are measured and amplified using hall effect transducers. All the dynamic variables are gathered in the Agilent 34970A data acquisition system. The communication with a personal computer is achieved by a GPIB bus.

Types of faults: the system does not allow to clearly attribute an anomaly to one specific fault. The threshold system proposed aims at giving a set of possible faults.

Description of system: 9.6 kWp system installed on a roof top, 90 PV modules divided into three arrays linked to the main grid via three single phase inverters each one with a nominal power of 2.5 kW.

Stage of development: academic research paper

Authors Summary: the fault detection algorithm is based on the comparison of simulated and measured yields by analyzing the losses present in the system. The identification of the kind of fault is carried out by comparing the amount of error deviations of both DC current and voltage with respect to a set of error thresholds evaluated on the basis of free fault system. The proposed method has been validated with experimental data in a grid connected PV system in the Centre de Developement des Energies Renouvelables (CDER) in Algeria.

Advantages: simple and easy to interpret methodology

8.10 Fault detection for PV enhanced adimensional approach

Title of Paper: Advanced fault detection for PV plants: an enhanced adimensional approach [28]

Year of publication: 2019

Authors: Barone V., Guastella S., Maugeri G., Bertani D.

Institutions: Ricerca Sistema Energetico (RSE SpA)

Countries: Italy

Parameters: DC voltage per module, DC current per module, AC voltage, AC current, irradiation, temperature, complete SCADA system

Data: This developed approach for fault detection operates on a dimensionless dataset obtained by sets of data that are acquired directly from the SCADA system. The approach classifies working points into cluster boxes, each representing different operational or failure conditions. Based on a set of thresholds, the method allows for a rapid classification of the working points. The resulting “density” of each cluster represents the weight used to determine the status of the plant.

Data resolution: the DC unit is used to perform current and voltage measurements with around 1% of accuracy. Each measure is acquired every 10 seconds and then mean values are calculated every 15 minutes.

Data filters: data has been filtered in order to remove night hours and communication errors. Measurements with an Irradiance level lower than 10 W/m^2 have also been removed.

Algorithms: clustering algorithms

Hardware: The PV plant used for this testing phase is monitored with a SCADA system according to the requirements set by the IEC 61724 standard [5]. The SCADA system includes a meteorological unit for the temperature measurement (NTC thermistors) and solar radiation on the array plane (IKS-ISET reference solar cell). A DC unit is used to perform current and voltage measurements.

Types of faults: open-circuited string, bypassed module, soiling, other.

Description of system: the proposed Advanced Failure Detection (AFD) method has been tested using a real PV system installed in Milan (Italy), at the RSE's Distributed Energy Resources Test Facility (DER-TF). The PV plant is made of two strings of 24 modules, with a nominal power of 155 Wp for each module and a total power of around 7.4 kWp. Two strings are connected to an inverter equipped with two MPPTs (one MPPT for each string), grid connected and mounted on a roof.

Simulation model: Typical failures have been simulated in order to collect data that allows the recognition of different types of faults (open-circuited string, bypassed module, soiling, etc..). The simulation method applies ideal IV curve models.

Stage of development: academic research

Summary: The developed approach for fault detection operates on an adimensional dataset obtained by a set of data that are acquired directly from the SCADA system. The approach classifies working points into cluster boxes, each representing different operational or failure conditions. Based on a set of thresholds, the method allows for a rapid classification of the working points. The resulting "density" of each cluster represents the weight used to determine the status of the plant. Three transformations are applied to the string current and voltage measurements: correction to STC, normalization and standardization. The resulting set of points is expected to be found around an equivalent IV curve. Deviations from such curves is classified in clusters of working points, each one representing a specific class of fault or a particular operative condition. The density of points inside each cluster gives an insight into the plant or string status. The resulting dataset, after the filtering and transformation phase, is then plotted in an adimensionless graph located around an equivalent IV curve that must be properly calculated to take into account the age of the plant, the electrical configuration and the module specifics. Results show that the density of points in each cluster does not undergo large variations if no faults occur in the monitored string. On the other hand, when a failure occurs a drastic shift towards a different cluster is observed.

8.11 Fault detection and diagnosis of photovoltaic system using fuzzy logic control

Title of Paper: Fault detection and diagnosis of photovoltaic system using fuzzy logic control [2]

Year of publication: 2019

Authors: Zaki Sayed A., Zhu Honglu, Yao Jianxi

Institutions: Cairo University, North China Electric Power University

Countries: Egypt, China

Parameters: Three ratios: theoretical Voc/actual Voc, theoretical DC voltage/measured DC voltage and theoretical DC current/measured DC current. Each parameter is a function of irradiation and temperature.

Data: data collected using simulated and real data of voltage ratio (VR), current ratio (IR) and open circuit ratio (OCR).

Data resolution: The measurement of power, voltage, and currents are obtained and collected by internal sensors at one-minute intervals.

Data filters: not specified

Programming languages: Matlab Simulink

Algorithms: The Sugeno FL classifier is exhibited and confirmed experimentally using fuzzy logic (FL) control. The architecture of the implementation is based on the Max-Min arrangement procedure with a centroid type for the defuzzification, moreover, eight FL rules were selected and implemented in order to detect accurately the occurred faults in the PV array.

Hardware: temperature sensor, solar irradiation sensor.

Types of faults: partial shading with bypass diode failure, open circuit failure, short circuit failure, snow falling, foliage, bird droppings.

(total fault predictions - actual faults)/(total fault predictions)

Description of system: The algorithm is validated using a 3.34 kWp solar PV system installed at the roof of North China Electric Power University (NCEPU). The PV array consists of 13 monocrystalline silicon JKM245 P-60-I PV 245W modules, each module has 60 cells with 3 bypass diodes (one diode per 20 cells) connected in parallel with the cells in reverse connection. The PV array has 13 modules connected in series and one parallel string.

Simulation model: single diode solar model

Stage of development: academic study

Authors Summary: This method is built based on comparing the measured electrical parameters with its theoretical parameters in both normal and faulty conditions of a PV array. For this purpose, three ratios of open circuit voltage, current, and voltage are obtained with their associated limits in order to detect eight different faults. Moreover, the fuzzy logic control FLC method is performed for studying the failure configuration and categorizing correctly the different faults occurred. Different simulated and experimental tests are conducted to demonstrate the performance of the proposed method.

Advantages: fault detection system can identify a variety of faults with similar characteristics

8.12 Local outlier factor-based fault detection and evaluation of photovoltaic system

Title of Paper: Local outlier factor-based fault detection and evaluation of photovoltaic system [29]

Year of publication: 2018

Authors: Hanxiang Ding, Kun Ding, Jingwei Zhang, Yue Wang, Lie Gao, Yuanliang Li, Fudong Chen, Zhixiong Shao, Wanbin Lai

Institutions: Hohai University, Concordia University, Changzhou Key Laboratory of Photovoltaic System Integration, Sunshore Solar Energy Co

Countries: China

Parameters: IV curve data for subarray

Data: 20 x 10 PV array simulation data is generated. The input data for the simulation model are the tilted co-plane irradiance and the temperature of the PV module from the data acquisition system of an actual outdoor PV system.

Data resolution: not specified

Data filters: data is sampled from 7:30 to 17:30.

Programming languages: Matlab Simulink

Algorithms: modified local outlier factor (LOF) Described as PVLOF

Hardware: IV curve sensor, data logger and other SCADA system hardware for sending data to a server for analysis.

Types of faults: Fault degree is divided as three categories: slight fault, fault and serious fault

Description of system: 10 kWp PV power plant built on the campus of Hohai University containing 40 PV modules

Simulation model: the modified model of PV module based on MATLAB proposed in (Ding et al., 2012) is used to simulate a 20×10 PV array. It consists of 200 TSM-240 PV modules. The specific electrical characteristic parameters of TSM-240 under the standard test condition (STC).

Stage of development: research study

Authors Summary: A PV array connected by PV modules in series and parallel with each string sharing the same voltage is simulated. The value of current can be used to identify the underperforming strings. In addition, considering the non-stationary stochastic characteristics of current of PV strings, the local outlier factor (LOF) is applied to detect the fault in the PV system by evaluating the deviation between the observed data. According to this method, the abnormal data will present some mathematical characteristics. These characteristics can reveal a degree of deviation. In PV systems, the degree of deviation of the abnormal data represents the fault degree.

8.13 Fault diagnosis model of photovoltaic array based on least squares support vector machine in bayesian framework

Title of Paper: Fault Diagnosis Model of Photovoltaic Array Based on Least Squares Support Vector Machine in Bayesian Framework [30]

Year of publication: 2017

Authors: Jiamin Sun, Fengjie Sun, Jieqing Fan, Yutu Liang

Institutions: North China Electric Power University

Countries: China

Parameters: irradiation, ambient temperature, variety of electrical parameters

Data: data from 5x3 PV array. Empirical results obtained from 10 public domain data sets.

Data resolution: not specified

Data filters: not specified

Programming languages: Matlab Simulink

Algorithms: Least Squares Support Vector Machine (LSSVM) in the Bayesian framework applying multiclassification. Gaussian RBF is used as the kernel function and the “One vs. One” classification algorithm is used to build the LSSVM multi-classifiers model.

Hardware: irradiation sensor, temperature sensor

Types of faults: short circuit, open circuit, abnormal aging

Description of system: A photovoltaic string as experimental subject, which consists of sixteen modules in series. We took fifteen of them into a 5 × 3 PV array, which consists of three photovoltaic strings in parallel, and each string has five modules in series.

Simulation model: this paper sets up a general simulation model of a 5×3 PV array using Matlab/Simulink; the PV array consists of two PV strings in parallel, and each string has three modules in series.

Stage of development: academic study

Authors Summary: First, based on the elaborate analysis of the change rules of the output electrical parameters and the equivalent circuit internal parameters of PV array in different fault states, the input variables of the photovoltaic array fault diagnosis model are determined. Second, through the LSSVM algorithm, in the Bayesian framework, the fault diagnosis model based on the output electrical parameters and the equivalent circuit internal parameters of the photovoltaic array is built. The LSSVM multi-classifiers are converted into six two-classifiers by the classification algorithm of “One vs. One”, which is “the normal vs. the short-circuits”, “the normal vs. the open-circuits”, “the normal vs the abnormal aging”, “the short-circuits vs the open-circuits”, “the short-circuits vs the abnormal aging”, and “the open-circuits vs the abnormal aging”.

Bayesian theory is used to optimize the parameters of the LSSVM classifier, regularization parameter θ, and kernel parameter σ; it then obtains the optimal classifier. A posteriori probability is derived from the two-classifiers. The proposed method has the ability to construct an optimal multiple-classifiers model and to obtain the posteriori probabilities of the samples, which can identify the states of the photovoltaic array. Four kinds of working conditions are simulated to validate the effectiveness of the approach—that is, the normal condition, the short-circuits condition, the open-circuits condition, and the abnormal aging condition. An experimental platform is built to test the experimental performance of the developed approach, while the experimental results also demonstrate the effectiveness of the fault diagnosis model in a practical system.

8.14 Statistical sensor-less short-circuit fault detection algorithm for photovoltaic arrays

Title of Paper: Statistical Sensor-less Short-Circuit Fault Detection Algorithm for Photovoltaic Arrays [31]

Year of publication: 2019

Authors: Amir Maleki, Iman Sadeghkhani, Bahador Fani

Institutions: Islamic Azad University

Countries: Iran

Parameters: voltage, current

Data: simulation data

Data resolution: 1ms samples

Data filters: low-pass filter

Programming languages: Matlab Simulink

Algorithms: not specified

Statistical tests/models: kurtosis measures

Hardware: hardware used for collecting voltage and current values

Types of faults: partial shading, open circuit faults

Description of system: the simulated PV system is a 5x5 7.6 kWp array formed using five parallel strings where each string consists of five 305.2W SunPower modules.

Simulation model: single diode model

Stage of development: academic research

Authors Summary: The paper proposes a waveshape based statistical fault detection algorithm for light fault detection. The proposed algorithm quantifies the waveshape: ”tailedness” of superimposed PV array power by kurtosis function. The proposed algorithm is able to discriminate the light fault condition from the severe partial shading and is also effective for open-circuit faults. In addition to no need for additional sensors, it does not require a training data set and the prior information about the PV array.

Advantages: no weather data necessary, high detection speed, robustness to noise in data. It does not require prior information about the PV system or a training dataset.

8.15 Complex network analysis of photovoltaic plant operations and failure modes

Title of Paper: Complex Network Analysis of Photovoltaic Plant Operations and Failure Modes [32]

Year of publication: 2019

Authors: Fabrizio Bonacina, Alessandro Corsini, Lucio Cardillo, Francesca Lucchetta

Institutions: University of Rome

Countries: Italy

Parameters: DC Voltage, solar irradiance, AC voltage-phase 1, 2, 3, active output power, string current

Data: 5 months of data

Data resolution: 5-minute samples of data

Data filters: outlier removal

Programming languages: Python

Algorithms: complex network analysis

Statistical tests/models: not specified

Hardware: solar irradiation on the plane of the modules and ambient temperature have been measured by a pyranometer and a PT-100 RTD sensor respectively, both installed on a sensor box near the modules. Both inverters are equipped with resistive potential divider voltage sensors for DC and AC parameters measurement for each conversion block. Shunt resistors have been used to measure 12 string currents as a representative sample of the plant.

Types of faults: not specified

Description of system: The solar field is connected to two inverters, each with three conversion blocks. Both inverters are grid-tied, feeding a medium voltage power distribution network. They are equipped with fully independent monitoring systems and incorporate a solar power controller to regulate the Maximum Power Point Tracker (MPPT) algorithm.

Stage of development: tested on real 1 MWp PV plant

Authors Summary: This paper presents a novel data-driven approach, based on sensor network analysis in PV power plants, to unveil hidden precursors in failure modes. The method is based on the analysis of signals from PV plant monitoring, and advocates the use of graph modeling techniques to reconstruct and investigate the connectivity among PV field sensors, as is customary for Complex Network Analysis (CNA) approaches. the proposed methodology is able to discover specific hidden dynamics, also referred to as emerging properties in a Complexity Science perspective, which are not visible in the observation of individual sensor signals but are closely linked to the relationships occurring at the system level.

8.16 Multiclass adaptive neuro-fuzzy classifier and feature selection techniques for photovoltaic array fault detection and classification

Title of Paper: Multiclass adaptive neuro-fuzzy classifier and feature selection techniques for photovoltaic array fault detection and classification [33]

Year of publication: 2018

Authors: A. Belaout, F. Krim, A. Mellit, B. Talbi, A. Arabi, E Krim,

Institutions: University of Sétif

Countries: Algeria

Parameters: voltage, current, area under the IV curve, short-circuit current, open-circuit voltage, maximum power point, voltage and current at MPP, slope of IV in the vicinity of open-circuit voltage, slope between MPP and Voc, slope at MPP, slope at Isc, slope between MPP and Isc, filling factor

Data: Simulated data using a real-time emulator. 2730 IV curves (faulty PV array), and 130 IV curves (healthy PV array) have been collected and stored into a current matrix and a voltage matrix.

Data resolution: not specified

Data filters: not specified

Programming languages: Matlab/Simulink and ControlDesk

Algorithms: Sugeno Fuzzy Inference System (FIS)

Hardware: real time emulator

Types of faults: partial shading, increased series resistance, by-pass diode short circuit, by-pass diode impedance, PV module short-circuit

Description of system: the PV system consists of six PV modules connected in series comprised each of 36 solar cells connected in series

Simulation model: Bishop model

Stage of development: tested on simulated data

Authors Summary: In this paper, a Multiclass Adaptive Neuro-Fuzzy Classifier (MC-NFC) for fault detection and classification in a PV array has been developed. Firstly, to show the generalization capability in the automatic faults classification of a PV array (PVA), Fuzzy Logic (FL) classifiers have been built based on experimental datasets. Subsequently, a novel classification system based on Adaptive Neuro-fuzzy Inference System (ANFIS) has been proposed to improve the generalization performance of the FL classifiers. The experiments have been conducted on the basis of collected data from a PVA to classify five kinds of faults. Dimensionality reduction is applied to use only the most relevant features for each type of fault. The paper compares the result of the approach to a simple ANN, showing superiority.

Advantages: the dimensionality reduction technique allows for understanding which of the features contains the most information relative to a given fault

8.17 Online fault detection in PV systems

Title of Paper: Online Fault Detection in PV Systems [9]

Year of publication: 2015

Authors: Radu Platon, Jacques Martel, Norris Woodruff, and Tak Y. Chau

Institutions: Natural Resources Canada, Canmet Energy

Countries: Canada

Parameters: AC energy, irradiance and temperature

Data: four months of data collected of solar array plane irradiance, module temperature, and ac power output were used to develop the fault detection system. The ac power value is obtained using the lifetime energy measurement generated by the inverter.

Data resolution: The irradiance value is obtained by averaging 120 measurements taken every 5 s, and the module temperature value is obtained by taking one instantaneous measurement every 10 min. AC energy collected by ten-minute intervals.

Data filters: removed samples with irradiance less than 50W/m² or zero power output, visual inspection. Removed outliers in the irradiance/power plane.

Programming languages: not specified

Algorithms: not specified

Hardware: weather station or temperature and pyranometer sensors

Types of faults: generic faulty states (not specified exactly which).

Description of system: The system is mounted on the roof of an institutional building, and it has a dc nominal capacity of 120 kWp, generated by 400 Heliene 300W 72-cell modules connected to a KACO XP100 inverter with a rated power output of 100-kW ac

Simulation model: Custom parametric approach: Pac = G (a1 + a2G+ a3 log (G)) (1 + a4 (Tm − 25)) where Pac is the ac power production (W), G is solar irradiance in the PV module plane (W/m²), Tm is the module temperature (◦C) and a1, a2, a3, and a4 are coefficients calculated, so that the model result is as close as possible to the measured data

Stage of development: research study tested on real PV system data

Authors Summary: faults in a real-world data set are identified based on a comparison between the expected power production and the measured one. Visual inspection allows for identifying faulty data points in the trained data that is used to compute thresholds for various parameter for different irradiance intervals.

Advantages: simple system that is easy to implement

8.18 Quickest fault detection in photovoltaic systems

Title of Paper: Quickest Fault Detection in Photovoltaic Systems [34]

Year of publication: 2018

Authors: Leian Chen, Shang Li, Xiaodong Wang

Institutions: Columbia University

Countries: USA

Parameters: DC mean power and DC mean voltage at the output of PV arrays, Duty cycle

Data: not specified

Data resolution: sampling frequency of 10 kHz

Data filters: not specified

Programming languages: Matlab by applying the toolbox SimPowerSystems

Algorithms: the machine learning method includes the two-class support vector machine (SVM) with a Gaussian radial basis function kernel, and the semi-supervised learning approach.

Statistical tests/models: generalized local likelihood ratio test, Sequential change detection

Hardware: data acquisition system for collecting DC voltage, DC current and the MPPT duty ratio

Types of faults: irradiance change (shading), line-line faults and ground faults

Description of system: system consists of two 100-kW PV module arrays (each comprised of 5 × 66 PV modules) as the input, a DC-DC boost converter, a three-phase three-level voltage source converter (VSC), and a 25-kV grid as the output. The Maximum Power Point Tracker (MPPT) using the "Incremental Conductance and Integral Regulator" technique is implemented.

Summary: This paper focuses on the detection of faults due to irradiance change (shading), line-line faults and ground faults in a grid-connected PV system by monitoring the mean power and mean voltage at the output of PV arrays (i.e., DC side). The statistical properties of the observed signals over time allow for detecting the change due to the occurrence. The PV fault detection problem is approached as a sequential change detection problem with unknown post-change distributions. The proposed approach does not need extensive field works or additional equipment for model validation, but only needs to monitor the commonly measured signals (power, voltage, etc.) of the PV array. Exploited are both the time correlation of fault signals and the correlation among multiple simultaneously measured signals (e.g., voltage, current, power), by applying a vector autoregressive (AR) model to describe the faulty signal. Such a model manifests the fact that the burst change due to faults can exert similar or even the same impact on various components of the system at the same time.

The fault detection system is split into an off-line phase and online-phase. During the off-line phase the machine learning model is trained by the data from the simulated normal output signals and the faulty signals which all include three attributes: the power, the voltage, and the duty cycle. In the on-line phase, the measurements with three attributes are sampled consecutively from the PV site and fed into the model. The trained model will make a decision between normal and faulty on each sample. The fault alarm will be triggered as soon as the classifier first identifies a sample as a faulty signal. The parameter c which controls the tolerance to the misclassification for SVM, and the parameter α which controls the solution rule in the kernel function for the semi supervised learning are tuned to satisfy the target false alarm periods.

Advantages: fast reaction to faults, small number of variables required that imply high adaptability to real case scenarios.

8.19 DA-DCGAN: an effective methodology for DC series arc fault diagnosis in photovoltaic systems

Title of Paper: DA-DCGAN: An Effective Methodology for DC Series Arc Fault Diagnosis in Photovoltaic Systems [6]

Year of publication: 2019

Authors: Shibo Lu, Tharmakulasingam Sirojan, B. T. Phung, Daming Zhang, Eliathamby Ambikairajah

Institutions: University of New South Wales

Countries: Australia

Parameters: current

Data: 20,000 normal samples and 20,000 arcing samples are extracted to form the target-domain dataset with a total size of 40,000 for training. Each sample consists of 400 data points corresponding to a 20 ms window size.

Data resolution: sampling frequency 20 kHz

Data filters: All the data is collected by a 200 kHz sampling rate (the collected signal is filtered by a 10 kHz low-pass filter and down sampled to 20 kHz by taking every tenth sample of the original signal for training)

Programming languages: not specified

Algorithms: Domain Adaptation and Deep Convolutional Generative Adversarial Network (DA-DCGAN)

Hardware: Data is saved to a PC via a data acquisition system (DAQ). The DAQ comprises a NI-PXIe-1073 chassis and a NI-PXIe-4300 module with a 16-bit analog to digital conversion resolution.

Types of faults: DC arcs

Description of system: PV emulator (Magna Power TSD-1000V-20A/415 programmable DC power supply) is connected in series with an arc generator and a 1.5-kW Sunny Boy single-phase inverter without any other external inductive or capacitive components (i.e., PV cables). The system is validated offline using pre-recorded PV loop current data from a real 1.5-kW grid-connected rooftop PV system. For target-domain data collection and real-time testing, the PV emulator is replaced by a rooftop PV string consisting of four JINKO JKM350M-72 mono-crystalline PV modules. The PV emulator is programmed to simulate a 1.5-kW grid-tied PV system with open-circuit voltage (Voc) of 207.2 V and short-circuit current (Isc) of 7.95 A at standard test condition (STC).

Stage of development: research study in laboratory setting with real PV system.

Authors Summary: In this paper, domain adaptation combined with deep convolutional generative adversarial network (DA-DCGAN)-based methodology is proposed, where DA-DCGAN first learns an intelligent normal-to-arcing transformation from the source-domain data. Then by generating dummy arcing data with the learned transformation using the normal data from the target domain and employing domain adaptation, a robust and reliable fault diagnosis scheme can be achieved for the target domain. The PV loop current is framed and arranged into a 2D matrix as input for cross-domain DC series arc fault diagnosis. The system is validated offline using pre-recorded PV loop current data from a real 1.5-kW grid-connected rooftop PV system. Also, the proposed method is implemented in an embedded system and tested in real-time according to UL-1699B standard.

Advantages: system only detects DC arc faults. The system requires a lot of hardware making the system potentially expensive.

8.20 Intelligent real-time photovoltaic module monitoring system using artificial neural networks

Title of Paper: Intelligent Real-Time Photovoltaic Module Monitoring System Using Artificial Neural Networks [35]

Year of publication: 2019

Authors: Sufyan Samara, Emad Natsheh

Institutions: An-Najah National University, Nablus, Palestine

Countries: Palestine

Parameters: irradiance, temperature, voltage, current

Data: obtained from cloud database used for real-time logging and monitoring of PV systems including environmental data.

Data resolution: not specified

Data filters: not specified

Programming languages: Matlab

Algorithms: feedforward neural network

Statistical tests/models: not specified

Hardware: current sensor, voltage sensor, pyranometer, temperature sensor, ATMEGA2560 microcontroller, data logger

Types of faults: the monitoring system will flag a PV module for maintenance if the predicted output power for that PV module, obtained from artificial neural network model, and the actual output power of that PV module, obtained from sensors, has a percentage difference of more than 10%.

Description of system: A current sensor; namely ACS712, and a voltage sensor; namely a voltage divider, are used for each PV module. The inputs to PV modules are collected in real-time using a Pyranometer sensor; apogee SP-212-SS, for measuring irradiance and a temperature sensor; Analog Device ADT7420. The Sharp's-NUS0E3E, and Astronergy-CHSM6610P are used.

Simulation model: single diode model

Stage of development: research study with an advanced prototype analyzing data from a real PV system.

Authors Summary: by comparing predicted and actual energy production of solar modules using current sensors, voltage sensors, a pyranometer, a temperature sensor, an ATMEGA2560 microcontroller and a data logger, the fault detection system can identify unexpected energy reductions on the solar module level. The fault detection system implements the feedforward neural network algorithm.

Advantages: the system is composed of low-level hardware and contains individual, custom designed, voltage and current sensors potentially allowing for identifying faults at the level of individual solar modules. This type of information can be very useful in identifying faults.

8.21 Improving efficiency of PV systems using statistical performance monitoring

Title of Paper: Improving Efficiency of PV Systems Using Statistical Performance Monitoring [1] -Chapter 2

Year of publication: 2017

Authors: Mike Green, Birk Jones, Eyal Brill, Jonathon Dore

Institutions: University of South Wales, SolarAnalytics

Countries: Australia

Parameters: Location, PV module type, inverter type, PV module orientation, PV module tilt, string configuration, current, voltage, frequency, active energy over 5 seconds, reactive energy over 5 seconds

Data: Meteorological data and satellite irradiance maps are supplied by the Australian National Weather Service. daily irradiance data is further manipulated using algorithms developed in collaboration with the University of New South Wales for: temporal irradiance separation, direct/diffuse irradiance separation, plane-of-array irradiance transposition

Data resolution: Temperature and wind-speed are supplied in 30-minute intervals. The irradiance data is supplied as a daily aggregate of satellite-derived global horizontal irradiance, energy parameters are collected with a resolution of 5 seconds

Data filters: not specified

Programming languages: not specified

Algorithms: not specified

Hardware: proprietary power meter mounted in the electrical distribution board

Types of faults: The PV generation is assessed every hour to determine if the site is online and producing. If the power is negligible, an alarm is sent to the system owner. At the end of each day, the daily energy generation is compared with the values calculated from the system parameters. If the performance is lower than expected, diagnostic algorithms are run and an alert is sent. Diagnostics are run on the system when the production is lower than the calculated production values. The analytics then compares the fault signal with known fault signatures to identify the likely cause of the underperformance.

Types of faults: shading, inverter clipping, power factor correction, string module faults, excessive soiling, degradation

Simulation model: energy production is simulated using system configuration and solar irradiation maps as input to their algorithms

Stage of development: Australia's largest independent solar monitoring company

Authors Summary: The PV system configuration is input to the program during configuration. The daily energy yield is simulated using the developed algorithms and the meteorological data including the irradiation maps supplied by governmental agency. If the energy yield is lower than the calculated production the system will then compare the fault signal with known fault signatures to identify the likely cause of the underperformance. Some examples of fault finding include shading, inverter clipping, power factor correction, string/module faults, soiling and degradation. The system employs different levels of resolution to ascertain the type of fault. Starting with hourly resolution down to 5 second resolution to match the fault signature.

Advantages: The system is relatively mature and is proven in the field. The system is capable of identifying when production is lower than expected and also the reason for this.

8.22 Monitoring the health of PV systems

Title of Paper: Improving Efficiency of PV Systems Using Statistical Performance Monitoring [1] Chapter 3

Year of publication: 2017

Authors: Mike Green, Birk Jones, Eyal Brill, Jonathon Dore

Institutions: MG Lightning LTD, Decision Makers LTD, PVpredict LTD

Countries: Israel

Parameters: Temperature, humidity, barometric pressure, wind speed, dew point, rain, sky maps, hourly energy or power

Data: not specified

Data resolution: varied; from 5 minutes to 30 minutes

Data filters: not specified

Programming languages: Java, Python, shell scripting, R

Algorithms: clustering regression trees

Hardware: none necessary

Types of faults: Health of system is quantified as follows: A = 100 to 97 % as expected; B = 97 to 95 % as expected; C = 95 to 90 % as expected; D = 90 to 85 % as expected; E = 85 to 80 % as expected; F = less than 80 % as expected, inverter clipping, power factor correction, string module faults,

Description of system: Monitoring internationally ten different residential, commercial and utility sized PV sites varying in design, configuration

Simulation model: machine learning used on inverter data parameters in conjunction with meteorological data supplied by commercial internet weather servers.

Stage of development: successful pilot projects ready for market as software as a service (SaaS) for existing monitoring companies

Authors Summary: Using regression trees and a variety of machine learning algorithms, the fault detection system, marketed as the SolarPet, predicts the energy production and compares PV system output to what was expected.

No hardware is required and the PV system configuration is not of interest. The machine learning algorithms require only the data field from the inverter and meteorological parameters, including skymaps, from local inexpensive weather servers.

The algorithms were developed to predict day-ahead and hour-ahead energy yield for all PV inverters, large and small, for the purpose of aiding grid managers in managing the grid using virtual PV plants to aid in easing the cost of spinning reserve and to enable accurate predictions for energy traders, both conventional and those pioneering in peer-to-peer block chain start-ups.

By repeating the prediction process at the end of the day using the historical weather parameters in place of the predicted weather parameters, the same algorithms can ascertain if the monitored inverter is performing as expected under the weather conditions that prevailed over the system that day.

Advantages: The system does not require any hardware or PV system configuration, only the data feed from the inverter and the location for matching a weather server.

9 COMPARISON OF UNSUPERVISED MACHINE LEARNING ALGORITHMS FOR FAULT DETECTION

9.1 Introduction

Chapter 8 presented several summaries of papers describing work on fault detection and identification. In this chapter, we attempt to apply and compare their performance on a common dataset, a sort of controlled environment. The ideal scenario would be to have a dataset with labeled faults, on which we could apply both supervised (requiring labels) and unsupervised (not-requiring labels) algorithms.

Given the difficulty to collect the data with explicitly labeled faults, we compared only unsupervised approaches, that do not need such labels. We implemented the algorithms from what was described and explained in each paper.

Among all the algorithms reviewed in the report, eight were suitable for the comparison (uniquely labeled in parenthesis for comparison):

1. SolarClique: detecting anomalies in residential solar arrays (SC) [10]
2. Local outlier factor-based fault detection and evaluation of photovoltaic system (LOF) [29]
3. Real-time fault detection in massive multi-array PV plants based on machine learning techniques (RTFD) [24]
4. Online fault detection in PV systems (OFD) [9]
5. Intelligent real-time photovoltaic panel monitoring system using artificial neural networks (NN) [35]
6. A statistical tool to detect and locate abnormal operating conditions in photovoltaic systems [14]
7. Statistics to detect low-intensity anomalies in PV systems [7]
8. Complex network analysis of photovoltaic plant operations and failure modes [32]

The comparison described in this chapter serves two goals: first, comparing the results obtained by the various algorithms, and then to identify trends in the approaches.

9.2 Comparison

The algorithms reviewed in this report cover a wide variety of scenarios, differing in the data they use and the nature of the results they return. This variety makes it necessary to point out that the data available for the comparison might not match exactly what was used by each algorithm in terms of features. We adapted the approaches as best we could. Note that this might have influenced the results.

9.2.1 Grouping the algorithms

Upon study, it became apparent that the algorithms examined could be naturally divided into two groups, based on their main working principle and type of output.

In the first group, including algorithms 1-5, a fault is explicitly identified as a significant deviation from the estimated normal behavior of the system. This normal behavior is usually estimated

via ML algorithms. In the second group, that includes algorithms 6-8, a fault is identified by the user, that observes the time evolution of statistical or structural indices of the data; in this second group, no machine learning algorithms are involved, only statistical measures.

Within the two groups identified above, we can further observe natural divisions. In the first group, we can classify the algorithms based on how the normal behavior of the system is estimated: in algorithms 1 and 2, this is performed according to the behavior of other systems, while in algorithms 3-5 it is done by looking at external factors, such as irradiance and temperature. In the second group, we can divide the algorithms based on the statistical indices they use: algorithms 6 and 7 use purely statistical indices, while algorithm 8 uses network measures, that describe statistical interdependencies of selected monitored sensors.

These first observations are already relevant, giving a structured overview of the unsupervised approaches to fault detection.

9.2.2 Comparison on common data

We continue the comparison in the following way: we look at the actual performance of the algorithms in the first group, comparing their sensitivity, and, for the second group, we look at the type of results we obtain when applying them.

FIRST GROUP

As previously mentioned, in the first group of algorithms, faults are explicitly identified as significant deviation from the estimated normal behavior of the system. Usually, the significance of the deviation is measured according to a threshold: if the deviation is above a given value, the system is labeled as faulty at that time instant. An example is in the figure below, where we show the application of algorithm 2 to one of the data sets available for this comparison.

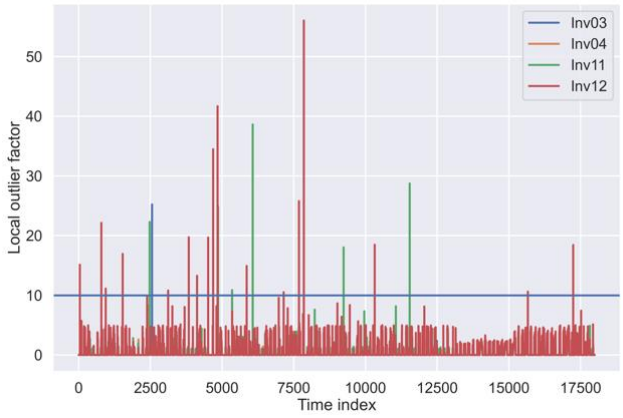


Figure 27: Results obtained applying algorithm 2 to a dataset containing four systems. Every point above the threshold 10 is considered a faulty data point.

The choice of the threshold can be made once and for all by the user, like the value 10 for the outlier factor in Figure 27, or it can be related to some statistics of the results, for example, a value of the deviation larger than three standard deviations might be considered a fault.

Since the algorithms in this first group explicitly identify faulty points, we can compare the number and the position in time of the faults they identify. Table 5 is an example of such investigation.

Table 5: Summary results from applying the algorithms of the first group. Each row and column represent an algorithm; each cell contains the number of faults identified simultaneously by the two corresponding algorithms. The diagonal elements are the total number of faults identified by the corresponding algorithm.

Algorithm	*LOF*	*SC*	*OFD*	*RTFD*	*NN*
LOF	**8**				
SC	1	**1742**			
OFD	0	11	**235**		
RTFD	1	23	39	**494**	
NN	5	1010	190	156	**2739**

In the table, each row and column refer to one algorithm, while the cells contain the number of faults detected: the diagonal contains the total number of faults identified by a given algorithm, while the other cells contain the number of faults identified by both algorithms (row and column). It is easy to see that both the total number and the number of common faults identified vary greatly, showing different sensitivities of the algorithms, and a general low agreement. Such behavior is repeated on the other available data sets.

This behavior could be explained both by the necessary adaptation of the algorithms to the data sets available and by the difference in approach of the individual algorithms, that allows them to identify more easily different types of faults.

SECOND GROUP

The second group contains algorithms that use statistical measures to suggest the user possible faults. In order to obtain these statistics, we need multiple identical systems to compare them, since we cannot use thresholds.

A sample result that can be obtained applying for example algorithm 6 is shown in Figure 28:

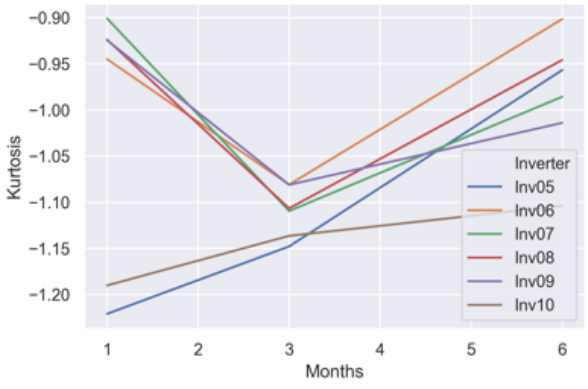


Figure 28: Evolution of the kurtosis index of the energy distribution over 6 months of analysis.

Figure 28 represents the evolution of the kurtosis index for the energy distribution of seven inverters in a PV plant. It is clear that at the beginning of the analysis, the user would have identified a fault for inverters 10 and 5, given their very different behavior with respect to the other inverters.

Algorithm 8 uses network measures to identify possible faults. This approach requires a more in-depth explanation, using concepts less familiar than normal statistical indices.

In this algorithm, we look at each sensor available, from current sensors to voltage, power, temperature, irradiance and so on, as a node in a network. We insert links, with varying strength, between the nodes of the network according to some statistical index computed between the signals recorded by each sensor-node. Since the signals change over time, the links change in strength. Network measures computed over this graph change over time as a consequence of this, and we can observe the interplay of such indices over time and try to spot faults when dramatic changes happen, as in the Figure 29 where around time index 10000 the trajectory changes.

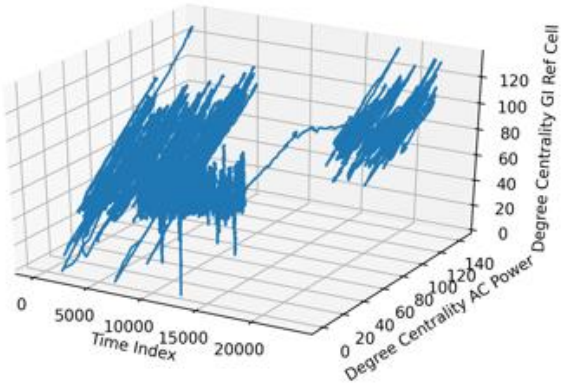


Figure 29: Evolution of network measures obtained applying algorithm 8.

In this second group of algorithms, the last word is left to the user, that must observe the evolution of the indices and decide which events might represent faults in the PV system.

9.3 Conclusions

The brief comparison carried out in this chapter has identified different natural groupings among the unsupervised approaches to fault detection.

The analysis of algorithms that explicitly identify faults (first group) has highlighted that different algorithms have very different sensitivities: we showed the results only for one dataset, but the situation is similar for the other datasets available. Further light on these differences might be shed using a labeled dataset, to give an objective performance metric for each algorithm and test their actual efficacy. The use of unlabeled data makes it possible only to point out how different the approaches are, noticing that there is still room for improvement for all the approaches presented.

REFERENCES

[1] M. Green, C. Birk Jones, E. Brill and J. Dore, "Improving Efficiency of PV Systems Using Statistical Performance Monitoring," International Energy Agency (IEA), 2017.

[2] S. A. Zaki, H. Zhu and J. Yao, "Fault detection and diagnosis of photovoltaic system using fuzzy logic contro," in *E3S Web of Conferences*, 2019.

[3] A. K. Vidal de Oliveira, M. Aghaei and R. Rüther, "Automatic fault detection of photovoltaic array by convolutional neural networks during aerial infrared thermography," in *European PV Solar Energy Conference and Exhibition*, Marseille, 2019.

[4] M. R. Maghami, H. Hizam, C. Gomes, M. A. Radzi, M. I. Rezadad and S. Hajighorbani, "Power loss due to soiling on solar panel: A review," *Renewable and Sustainable Energy Reviews,* vol. 59, pp. 1307-1316, 2016.

[5] S. Li, L. Chen and X. Wang, "Quickest Fault Detection in Photovoltaic Systems," *EEE Transactions on Smart Grid,* 2016.

[6] S. Lu, T. Sirojan, T. Phung and D. Zhang, "DA-DCGAN: An Effective Methodology for," *IEEE Access,* vol. 7, pp. 45831-45840, 2019.

[7] M. Carpentieri and S. Vergura, "Statistics to Detect Low-Intensity Anomalies in PV Systems," *Energies,* vol. 11, no. 30, pp. 1-12, 2018.

[8] M. H. Ali, A. Rabhi, A. El Hajjaji and G. M. Tin, "Real Time Fault Detection in Photovoltaic Systems," *Energy Procedia,* vol. 111, pp. 914-923, September 2017.

[9] R. Platon, J. Martel and N. Woodruff, "Online Fault Detection in PV Systems," *IEEE Transactions on Sustainable Energy,* vol. 6, no. 4, pp. 1200-1207, 2015.

[10] S. Iyengar, S. Lee, D. Sheldon and P. Shenoy, "SolarClique: Detecting Anomalies in Residential Solar Arrays," in *1st ACM SIGCAS Conference*, 2018.

[11] Å. Skomedal, M. B. Øgaard, J. H. Selj and H. Haug, "General, Robust and Scalable Methods for String Level Monitoring in Utility Scale PV Systems," in *36th European Photovoltaic Solar Energy Conference and Exhibition*, Marseille, 2019.

[12] M. Manjunath and J. Walters, "Detecting Corrupt Data in a PV Plant Database Using Statistical and Classical Approaches".

[13] *Statistics 101: ANOVA, A Visual Introduction.* [Film].

[14] S. Vergura, "A Statistical Tool to Detect and Locate Abnormal Operating Conditions in Photovoltaic Systems," *Sustainability,* vol. 10, no. 3, pp. 1-15, 2018.

[15] C. Shalizi, "Chapter 5: The Bootstrap".

[16] J. McCarthy, "Arthur Samuel: Pioneer in Machine Learning".

[17] S. Aghabozorgi and J. Santarcangelo, "Machine Learning with Python," [Online]. Available: https://www.coursera.org/learn/machine-learning-with-python?specialization=ai-engineer#instructors.

[18] "Linear Regression," Massachusetts Institute of Technology.

[19] F. Jia, L. Luo, S. Gao and J. Ye, "Logistic Regression Based Arc Fault Detection in Photovoltaic Systems Under Different Conditions," *Journal of Shanghai Jiaotong University,* 2019.

[20] S. Y. Sheehan S, "Deep Learning for Population Genetic Inference.," *PLoS Comput Biol,* vol. 12, no. 3, 2016.

[21] IBM, "IBM Machine Learning Professional Certificate," IBM, [Online]. Available: https://www.coursera.org/professional-certificates/ibm-machine-learning?page=4. [Accessed 25 11 2020].

[22] P. Graniero, A. Louwen, R. Schlatmann and C. and Ulbrich, "Comparison of Different Data Sources for Machine Learning Algorithms in Photovoltaic Output Power Estimation," in *37th European Photovoltaic Solar Energy Conference and Exhibition*, Lisbon, Portugal, 2020.

[23] M. Boyd, "High-Speed Monitoring of Multiple Grid-Connected Photovoltaic Array Configurations," U.S. Department of Commerce, NIST, 2015.

[24] C.-C. Hsu, J.-L. Li and Y.-S. Chen, "Real-time Fault Detection in Massive Multi-array PV Plants Based on Machine Learning Techniques," in *36th European Photovoltaic Solar Energy Conference and Exhibition*, Marseille, 2019.

[25] M. Y. Chang and C.-C. Hsu, "PV O&M optimization by AI practice," in *36th European Photovoltaics Solar Energy Conference and Exhibition*, Marseille, 2019.

[26] M. Hassan Ali, A. Rabhi, A. El Hajjaji and G. M. Tina, "Real Time Fault Detection in Photovoltaic Systems," *Energy Procedia,* vol. 111, pp. 914-923, 2017.

[27] S. Silvestre, A. Chouder and E. Karatepe, "Automatic fault detection in grid connected PV systems," *Solar Energy,* vol. 94, pp. 119-127, 2013.

[28] K. Kara, F. Harrou, E. Garoudja and Y. Sun, "Statistical fault detection in photovoltaic systems," *Solar Energy,* vol. 150, pp. 485-499, 2017.

[29] J. Zhang, Y. Wang, K. Ding and H. Ding, "Local outlier factor-based fault detection and evaluation of photovoltaic system," *Solar Energy,* vol. 164, pp. 139-148, 2018.

[30] J. Sun, F. Sun, J. Fan and Y. Liang, "Fault Diagnosis Model of Photovoltaic Array Based on Least Squares Support Vector Machine in Bayesian Framework," *Applied Sciences,* vol. 7, no. 11, 2017.

[31] B. Fani, I. Sadeghkhani and A. Maleki, "Statistical sensorless short-circuit fault detection algorithm for photovoltaic arrays," *Journal of Renewable and Sustainable Energy,* vol. 11, no. 5, 2019.

[32] A. Corsini, F. Bonacina, F. Lucchetta and L. Cardillo, "Complex Network Analysis of Photovoltaic Plant Operations and Failure Modes," *Energies,* vol. 12, no. 10, 2019.

[33] A. Belaout, A. Mellit, A. Belaout and F. Krim, "Multiclass adaptive neuro-fuzzy classifier and feature selection techniques for photovoltaic array fault detection and classification," *Renewable Energy,* vol. 127, pp. 548-558, 2018.

[34] L. Chen, S. Li and X. Wang, "Quickest Fault Detection in Photovoltaic Systems," *IEEE Transactions on Smart Grid,* vol. 9, no. 3, pp. 1835-1847, 2016.

[35] S. Samara and E. Natsheh, "Intelligent Real-Time Photovoltaic Panel Monitoring System Using Artificial Neural Networks," *IEEE Access,* vol. 7, pp. 50287 - 50299, 2019.

[36] J. Starmer, "StatQuest with John Starmer," 19 August 2019. [Online]. Available: https://www.youtube.com/watch?v=g9c66TUylZ4.

ISBN 978-3-907281-07-9