# Anomaly Detection and Classification for Photovoltaic Systems

## Enhancing PV monitoring through a universal method, leveraging neighbouring systems data

### Master's Thesis


| | |
|---|---|
| Course of study | **MSc in Circular Innovation and Sustainability** |
| Author | **Maxime Charrière**<br>*ID 17-295-551*<br>*contact@maximecharriere.ch*<br>*https://linkedin.com/in/maximecharriere/* |
| Advisor | **Prof. Dr. Christof Bucher** |
| Co-advisor | **Prof. Dr. Horst Heck** |
| Date & Location | **August 28th, 2024 — Bern, Switzerland** |

Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

**Bern University of Applied Science** — BFH

- School of Engineering and Computer Science
- School of Architecture, Wood and Civil Engineering
- School of Agricultural, Forest and Food Sciences HAFL
- Business School

**MSc Circular Innovation and Sustainability**

# Abstract

The primary objective of this thesis is to enhance the monitoring of residential photovoltaic (PV) systems by developing a novel methodology for anomaly detection and classification. This approach is designed specifically for companies managing small-scale PV systems, typically ranging from 3 to 100 kWp, which often lack advanced monitoring sensors and professional management, such as utility-scale farms. Our method requires only the energy production measurements from the monitored system and nearby PV systems, making it a cost-effective and universal solution that can be implemented across a wide range of PV installations without significant additional investment.

As Switzerland aims to increase its solar energy production from 4 TWh today to 34 TWh by 2050 to meet its energy transition goals, there is a pressing need for effective monitoring solutions for PV systems. The planned expansion will involve a substantial increase in the number of small-scale, decentralized PV systems, which will require consistent monitoring and maintenance. Given the diversity of equipment, installer companies, and collected data, a universal anomaly detection system is critical. Such a system will help minimize Mean Down Time (MDT) and Mean Time To Repair (MTTR), ensuring rapid and efficient identification of potential issues to maximize solar production, reduce economic losses, and helping Switzerland to achieve its objectives.

The methodology developed in this thesis utilizes a combination of data analysis techniques, machine learning, and statistical methods. By employing the Half Sibling Regressor principle alongside a physics-based normalizer, the method accurately estimates the expected daily production of a PV system based on the energy production data of its neighbouring systems. The system then monitors for deviations between expected and actual production, effectively detecting anomalies when significant underperformance is identified. This capability is crucial for minimizing the Mean Down Time and Mean Time To Repair of PV systems, thereby enhancing operational efficiency and reliability.

The performance of this novel approach was evaluated using real-world data from 326 PV systems, each with an average of 400 days of historical data. The results demonstrated an average Mean Absolute Percentage Error (MAPE) of 4.38% in estimating the expected production, with a standard deviation of 4.60% between the performance of each tested system. The anomaly detection algorithm successfully identified 97.4% of simulated anomalies. Moreover, the model's ability to handle missing data ensures continuous monitoring and anomaly detection, even when some neighbouring systems fail to provide data due to technical issues. Our study's MAPE of 4.38% is in line with similar studies: SolarClique reported a MAPE of 7.81% in a study involving 88 systems in Austin, Texas, while SunDown achieved a MAPE of 2.98% by comparing each module performance within a single PV system.

Despite these promising results, several areas for further improvement have been identified. Future work should focus on incorporating higher-frequency data, such as hourly or minute-level measurements, to enhance the precision of anomaly detection and allow for more frequent monitoring updates. Additionally, developing an advanced anomaly classification system would provide more detailed insights to the company into the nature of the detected issues, enabling quicker and more targeted maintenance actions. Real-world field testing and iterative refinement based on industry feedback will also be essential to optimize the model further and develop more sophisticated detection and classification rules tailored to the specific needs of PV system operators.

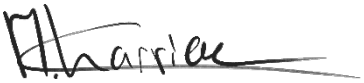
**Keywords**: Anomaly Detection, Photovoltaic System, Machine Learning, Data Analysis, Solar Energy, Monitoring, Sustainability, Renewable Energy

# Declaration of Independent Authorship and Granting of Usage Rights

I hereby declare that:

- I have read and understood the "Regulations on Scientific Integrity at Bern University of Applied Sciences" (WissIR) and am aware of the consequences of not complying with them;
- I have written this piece of assessed coursework in compliance with these principles;
- I have created this work personally and independently, and have identified and marked all content not authored by me with a precise reference to its origin;
- I have worked in a thorough manner throughout the creation of this work and have not incorporated any content generated by an artificial intelligence without careful consideration;
- I accept that my work will be checked by plagiarism-recognition software and subsequently stored in the BFH database;
- I grant BFH a free, perpetual, non-exclusive licence to use my work.

Place, Date: Bern, 23.08.2024

Signature: 

**Information regarding the use of assessed coursework submitted by students at the Bern University of Applied Sciences BFH.**
Theses are components of the degree programme and are written by students independently. BFH accepts no responsibility for possible mistakes in these theses and is not liable for any possible resulting damage.

In accordance with intellectual property rights, theses written by students not employed by BFH belong to their authors. However, students grant BFH free, perpetual and non-exclusive rights to their theses by signing the Declaration of Independent Authorship and Granting of Usage Rights.

The Core Team and the Steering Committee of the
Master of Science in Circular Innovation and Sustainability

April 2024

# Table of Contents

# Glossary and Nomenclature

This chapter provides definitions and explanations of key terms, concepts, and acronyms used throughout the thesis. It includes a glossary of abbreviations and technical terms. Additionally, it clarifies the distinctions between closely related concepts and terminologies. This section aims to ensure a clear understanding of the technical language and concepts used, facilitating better comprehension of the subsequent chapters. It is possible to read this chapter now, or later when the terms are discussed in the thesis.

## Glossary

| | |
|---|---|
| **ADC** | Anomaly Detection and Classification. |
| **FDD** | Fault Detection and Diagnostics. |
| **GHI** | Global Horizontal Irradiance: Solar irradiance received on a horizontal surface, including both direct and diffuse components of sunlight, during clear sky. |
| **PAO** | Plane of Array Irradiance: Solar irradiance incident on the surface of the PV modules |
| **ML** | Machine Learning |
| **ANN** | Artificial Neural Network |
| **Seasonality** | Seasonality in time series data is a trend that occurs at regular intervals. It's not necessarily linked to annual seasons. |
| **MAPE** | Mean Absolute Percentage Error |
| **PV** | Photovoltaic |
| **PV System / PVS System** | /All represent a PV system, as defined in the next section |
| **Target PV System** | The PV system that is currently analysed for anomalies detection among all PV systems |

## Definition of a PV Cell / Module / String / Array / System

The definition of the different parts constituting a PV System differ from one source to another. In this thesis, the following definitions will be used to name the different sub-parts of a PV system:

- **PV Cell**: A solar cell is the basic building block of a photovoltaic system. It is a semiconductor device that converts sunlight directly into electricity through the photovoltaic effect.
- **PV Module**: Also known as a solar panel, a module is a collection of interconnected solar cells, in parallel or series, encapsulated within a protective casing.
- **PV String**: A string is formed by connecting multiple PV modules together in series, typically to achieve higher voltage outputs. Stringing modules in series increases the system voltage while maintaining the current level.
- **PV Array**: A PV array consists of multiple strings of PV modules connected to a single inverter. Arrays can vary in size and configuration depending on the specific application and energy requirements.
- **PV System**: The PV system refers to the entire setup of interconnected components, including modules, inverters, mounting structures, electrical wiring, and batteries components. It

encompasses all the hardware and components required to generate, convert, and distribute solar electricity at a given site.

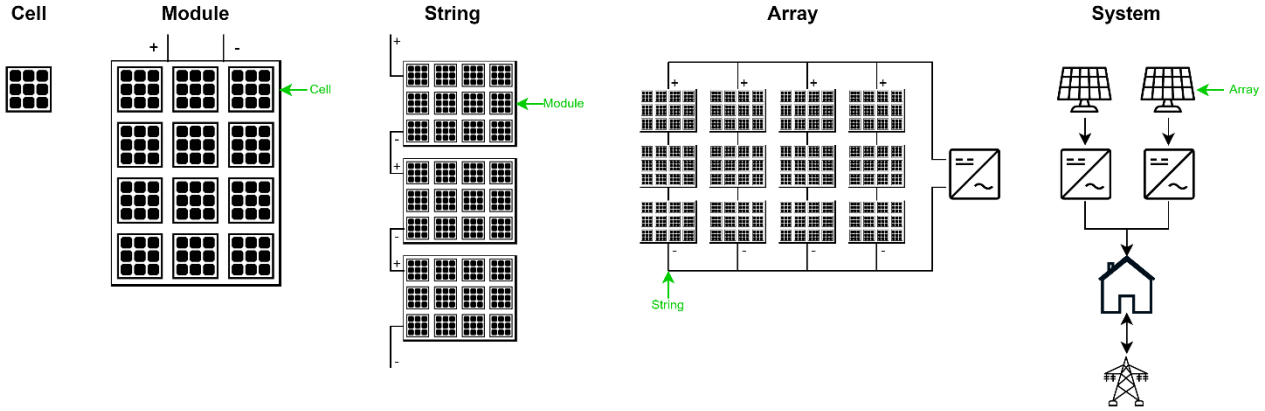



Figure 1 Representation of the different subdivision of a PV system use in this thesis, i.e. a cell, a module, a string, and an array.

# Definition of FDD vs. ADC

In the various papers concerning the handling of PV panel anomalies and faults, two different terms are used to define this practice. It is important to understand the differences between them to understand the different solutions presented in the literature.

**Fault Detection and Diagnostics (FDD),** which is a well know subject in the literature, focuses on identifying specific faults or malfunctions within a PV system that may impact its performance or operation, and aim to diagnose the root cause and location of the fault in the PV system. FDD techniques typically involve the use of specific sensors, such as I-V curve tracker, cameras, or irradiance meter, as well as predefined fault models and algorithms.

On the other hand, **Anomaly Detection and Classification (ADC)** aims to identify deviations or anomalies in the behaviour of a PV system that may indicate unusual or unexpected conditions, without being able to attribute them to specific known faults, but rather to a class of possible anomalies. ADC techniques need fewer specific sensors and try to detect anomalies using historical or real-time data, and leverage statistical analysis and machine learning algorithms to identify patterns of abnormal behaviour.

In this study, we will mainly talk about Anomaly Detection and Classification.

In fact, the **detection of an anomaly is a type of classification**. The detection of an anomaly is simply a classification of this one into two categories named “normal” or “anomalous”. Further classification means that more categories where added, such as “normal", “disconnected string”, “broken PV module”, …, “other”.

# Definition of Machine Learning

Firstly, one confusion that arises in the study of machine learning is in understanding the difference between Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), and Artificial Neural Network (ANN). As ML is used in this thesis, it is important to understand the differences between these terms.

- **Artificial Intelligence** encompasses the field of creating machines capable of performing tasks that typically require human intelligence. It has been used since the year 1950s and spans various techniques, including machine learning (ML), as well as traditional rule-based approaches.

- **Machine Learning** is a subset of AI focused on developing algorithms and statistical models that enable computers to perform tasks without explicit programming. ML algorithms learn from data and improve their performance through experience. These algorithms can be categorized into supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning, depending on the nature and labialization of the training data and learning process.
- **Artificial Neural Networks** are a subset of ML and are the backbone of DL algorithms. They are computational models inspired by biological neural networks in the human brain. ANNs consist of interconnected nodes (neurons) organized in layers, including an input layer, one or more hidden layers, and an output layer. Each connection between neurons is associated with a weight that determines its strength. ANNs learn by adjusting these weights based on input data and desired output.
- **Deep Learning**, also called Deep Neural Network, is one kind of ANNs with more than one hidden layer ("deep") to learn from complex relations from large datasets.

Then, when Machine Learning tools are used to make predictions, two categories of models can be distinguished :

- **Regression** is used to predict continuous values (rather than discrete values). In our case, estimating the daily expected PV energy production is considered a regressive task, since energy can be any continuous decimal number.
- **Classification** is used to classify observations into categories. In our case, classification can be the PV anomalies detection algorithm that classifies an observation as “normal”, or “anomalous”.

Finally, ML models can be categorized into three other categories:

- In **Supervised Learning**, models are trained using labelled data, where the historical input data is paired with a known output. The algorithm learns to map inputs to outputs by minimizing the error between its predictions of the model and the actual output. In our case, estimating the daily expected PV energy production is a supervised task as we know to production history.
- **Unsupervised Learning** deals with unlabelled data, where the model must identify patterns and structures in the input data without predefined output data. Unsupervised learning could be employed for tasks like identifying distinct patterns in PV system performance data to detect anomalies.
- **Semi-supervised Learning** is a hybrid approach that utilizes data partially labelled. This technique is useful when acquiring labelled data is costly or time-consuming, but there is an abundance of unlabelled data. The model initially learns from the labelled data and then improves its performance by leveraging the unlabelled data. This method could be used in our case to improve anomaly classification by labelling some of them, without labelling all previous anomalies.

# Definition of White-Box vs. Black-Box Tools

A distinction is made between "White-Box" and "Black-Box" data analysis models:

- **White-Box Models**: In a White-Box model, the structure, calculations, and parameters are consciously developed and chosen by the model designer. The advantage of a White-Box model is that each aspect of it is known, controllable, and can be improved upon. However, this requires a complete understanding of the internal workings of the analysed system, including its parameters and the effects of one variable on another.
- **Black-Box Models**: In contrast, a Black-Box model is trained using historical data, with its structure and parameters automatically determined to minimize prediction error. The Black-Box model does not require prior knowledge of the internal workings of the system or the relationships between variables, which allows it to uncover complex relationships that might not be found manually. However, it often requires a large amount of data, making it less immediately applicable when setting up a new PV system. Additionally, the model's logic is opaque, making it difficult to interpret and understand its decision-making processes.

# A. Introduction and Project Definition

The goal of this thesis is to assist companies that manage residential photovoltaic (PV) systems in detecting anomalies in their managed PV systems using a novel methodology. Our approach requires only the measurements of the energy production from the monitored system and nearby PV systems for comparison. This allows us to estimate the expected production of the monitored system and to automatically trigger an alarm when an anomaly is detected. This method involves the use of data analysis tools such as computer science, machine learning and statistical methods.

Previous fault detection methods in the literature are numerous and varied, but they often require additional measurement devices, access to inverter data, the purchase of weather data, satellite or drone imagery, or on-site operator visits. These resources are typically only available to utility-scale solar farms or expensive PV systems. In contrast, our method requires only the measurement of produced energy, an information that is generally available for the majority of PV system. Therefore, our method is universal and can be implemented on any existing system without significant additional investment. Additionally, previous research often uses laboratory or simulated data, which contains little to no measurement noise, errors, or missing values. In our study, however, the detection method was tested on 326 real PV systems with approximately 400 days of data per system. Our research is closely related to the method proposed by Iyengar et al., called SolarClique (2018), which inspired us, particularly in its use of the Half Sibling Regressor.

This thesis is conducted in partnership with a company that provides IoT solutions for automated building monitoring, including photovoltaic production monitoring. This company provided the data necessary for this study.

## 1 Problem Statement

To achieve its energy transition goals, Switzerland plans to increase its solar energy production from 4 TWh today to 34 TWh by 2050 (Federal Office for the Environment, 2021). This substantial increase will require a significant expansion of small-scale photovoltaic (PV) systems across the country in a decentralized manner. This expansion means there will be a growing number of PV systems to monitor and maintain, each installed by different providers, using varying equipment and collecting different types of data. Unlike utility-scale PV systems, these residential systems typically lack professional personnel dedicated to their management, and advanced monitoring sensors. As a result, a universal, automated anomaly detection system becomes essential to minimize their Mean Down Time (MDT) and Mean Time To Repair (MTTR). Rapid and effective anomaly detection is crucial for increasing solar production capacity, reducing economic losses, and enhancing site safety by identifying potential issues before they worsen.

## 2 Objectives

To address this problem, our goal is to answer the following research question:

> "*Which data analytics methodology can improve the monitoring of photovoltaic systems by accurately estimating expected production and automatically detecting and classifying anomalies, using only historical energy production data from the target and neighbouring systems?*"

Our partner is interested in a solution where they provide daily energy production measurements from the PV systems they manage, and in return, receive useful monitoring information for each system. This information may include:

- Expected production for the day, month, and year.
- Underproduction for the day, month, and year.
- Alerts indicating a potential anomaly has been detected.
- Suggestions on the type of detected anomaly.
- Alerts for PV systems performing abnormally poorly from its start.
- Current performance relative to the system's total capacity.
- Aging and degradation status of the PV system.

In this thesis, we will therefore design and implement a universal methodology that can provide these useful monitoring insights for PV systems, using only energy production measurements as input data. This will be achieved through computational tools, machine learning, and statistical methods.

# 3 Challenge

The challenge in achieving these objectives lies in the fact that energy production measurements are highly noisy due to the numerous factors influencing this variable. These factors include temperature, cloud cover, shading from nearby buildings, and specific characteristics of each PV system, such as losses, components characteristics, tilt, and azimuth. Weather conditions, for their part, are the most significant factors and are inherently stochastic, making the measurements almost random. This unpredictability complicates the detection of potential anomalies that may disrupt production, especially when no weather data is available.

# 4 Intended Audience of the Thesis

This thesis is primarily intended for academic advisors and experts evaluating this work, as well as the partner company commissioning the study to explore potential improvements in their PV monitoring and anomaly detection capabilities. Additionally, the thesis is aimed at readers with a foundational understanding of Photovoltaic Systems, Statistics and Machine Learning. While detailed explanations of basic principles in these fields are beyond the scope of this thesis, efforts have been made to present the content in a manner that is accessible to those with general technical knowledge.

# 5 Target Group for our Anomaly Detection Method

The anomaly detection method developed in this thesis specifically targets companies involved in the monitoring of residential, small-scale PV systems, typically ranging from 3 to 100 kWp. These systems, often installed on residential rooftops, lack the advanced sensing and monitoring tools available to larger utility-scale solar farms, which benefit from continuous professional management and extensive sensor instrumentation. Due to cost constraints, small-scale PV systems require a different approach to monitoring and anomaly detection.

# 6 Paper Structure

This thesis is organized into several key chapters to guide the reader through the research, methodology, and results.

- **Chapter A: Introduction and Project Definition**
  This chapter introduces the main goals and scope of the thesis. It outlines the problem statement, objectives, challenges, intended audience, and target group for the anomaly detection method developed in this work.
- **Chapter B: Literature Review & Knowledge Acquisition**
  This chapter provides a comprehensive review of existing research and methodologies related to anomaly detection and classification (ADC) and fault detection and diagnostics (FDD) in PV systems. It also covers relevant background information and factors affecting PV production, serving as the foundation for developing our ADC method.
- **Chapter C: Methodology – Algorithm Design and Implementation**
  This chapter details the development of the proposed algorithm for anomaly detection. It includes descriptions of the input data, software requirements, overall strategy, and each component of the algorithm, such as normalization, filtering, the Half-Sibling Regression model, and anomaly detection. It also explains the user interface and data storage methods used in the implementation.
- **Chapter D: Results & Discussion**
  This chapter presents the results obtained from applying the proposed algorithm to real PV system data. It discusses the accuracy of the expected production estimations, the effectiveness of anomaly detection, and the impact of various factors, such as the amount of historical data and the number of neighbouring systems. The chapter also evaluates the performance of different components of the algorithm.
- **Chapter E: Limitations and Future Directions**
  This chapter identifies the limitations of the current methodology and proposes future research directions to enhance the anomaly detection framework. It addresses potential challenges such as regional anomalies, hourly data usage, and data quality issues, and provides recommendations for further improvement.
- **Chapter F: Conclusions**
  This chapter summarizes the key findings and contributions of the thesis. It discusses the potential benefits of the proposed methodology for PV system monitoring and provides specific recommendations for the partner company to implement the findings.

# 7 Thesis Management

This section briefly outlines the methods used for managing this thesis project. The structure was organized around distinct work packages (WPs), each with specific objectives to achieve. A basic Gantt chart was created to plan these objectives over time. Also, the work was carried out as independently as possible, without frequently seeking external assistance from advisors or the partner company, as we are expected to demonstrate independence in carrying out the thesis.

## 7.1 Work packages

The project was divided into work packages, each addressing a specific research question (RQ):

**WP1: Literature Review on Anomaly Detection and Classification for PV Systems**
*Objective*: Conduct a comprehensive literature review on the state-of-the-art methodologies for detecting and classifying anomalies in photovoltaic (PV) systems.

*RQ*: "What are the current methodologies and findings in the application of data analytics to estimate the expected yield and automate the detection and classification of anomalies in PV systems?"

**WP2: Algorithm Development for Yield Estimation**
*Objective*: Develop an algorithm to estimate the expected yield of a PV system based on its own historical data and that of similar systems.

*RQ*: "What data analytics methodologies and techniques are most effective in estimating the expected yield of PV systems using historical data from the system itself and from similar systems?"

**WP3: Algorithm Development for Anomaly Detection**
*Objective*: Create an algorithm that detects abnormal deviations between the expected and actual yield of a PV system.

*RQ*: "How can abnormal deviations between expected and measured yield in PV systems be detected?"

**WP4: Algorithm Development for Anomaly Classification**
*Objective*: Develop an algorithm to determine the cause of detected anomalies.

*RQ*: "What types of anomalies can be identified in PV systems, and how can they be automatically classified?"

## 7.2 Planification and Objectives

To effectively manage and schedule the tasks associated with each work package, a basic Gantt chart was utilized. This chart provided a visual timeline for the planning of each objective, ensuring a structured approach to completing the thesis.

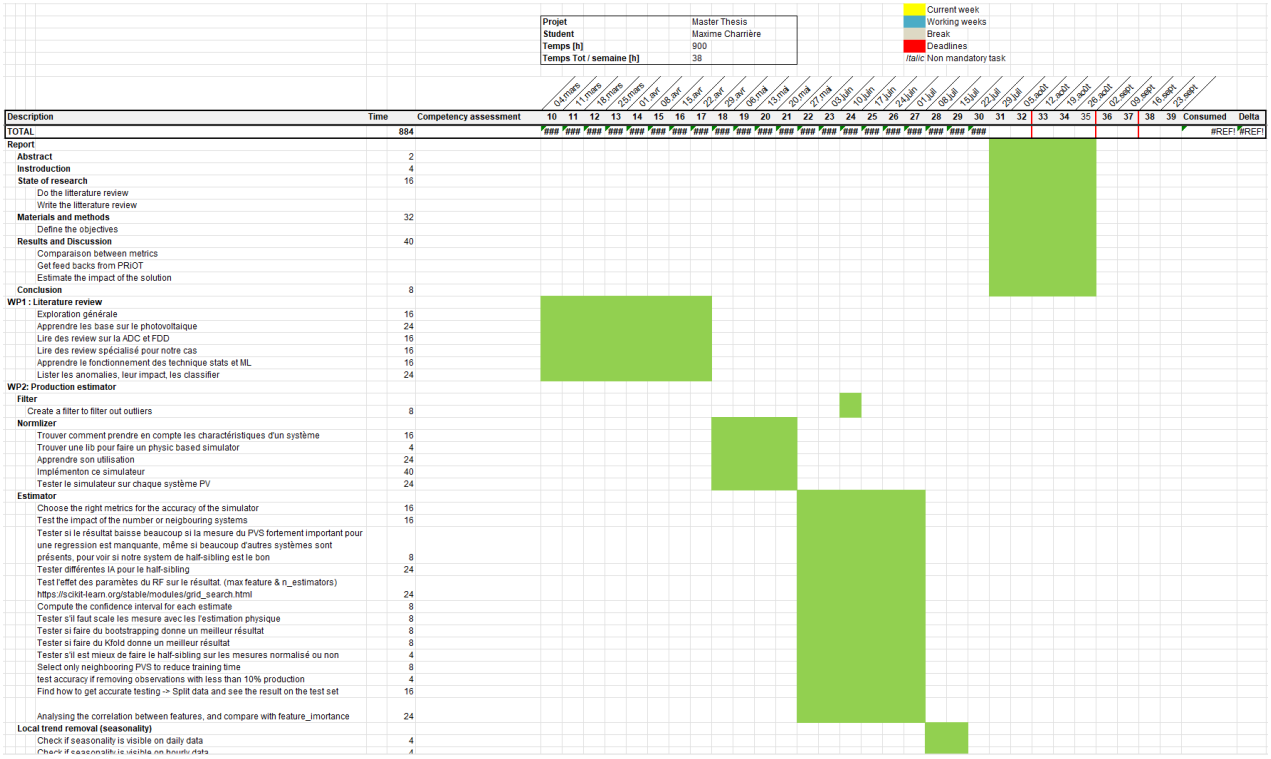



Figure 2 Screenshot of the planning of objectives

# B.Literature Review & Knowledge Acquisition

## 1 Methodology

In this chapter, we review the existing literature on Anomaly Detection and Classification (ADC) and Fault Detection and Diagnostics (FDD) methods to establish a knowledge base for designing our own detection method. This literature review follows a traditional, exploratory approach rather than a systematic or meta-analytical method. The stages of this review are as follows:

1. **Understanding PV Systems**: Initially having no knowledge of photovoltaics, the first step was to gain a comprehensive understanding of PV systems, including their components, operational principles, and various influencing factors. This phase involved studying the physics behind PV systems and examining the impact of parameters such as tilt, azimuth, system design (e.g., equipment type, module configuration in parallel or series, grid-connected or off-grid setups), and different scales (utility plants, residential rooftops). The role of various stakeholders in the PV ecosystem was also identified. This foundational knowledge was essential to understand the relationships between PV system elements, identify different use cases, potential target users for our ADC solution, and recognize potential anomalies. However, this background information will not be elaborated upon in the thesis, as it is beyond the primary scope.
2. **Review of Factors Affecting PV Production**: The second step focused on identifying the various factors that influence PV system production. The goal was to categorize the origins of each disturbance affecting power output, which is crucial for designing an effective anomaly detection solution. Insights for this step were derived from the same sources analysed in next step (3), along with inputs from the collaborating company and thesis advisor, who are experts in PV systems.
3. **Review of ADC and FDD Methods**: The third step involved a comprehensive review of the literature on ADC and FDD methods. An explanation of these terms can be found in the "Glossary and Nomenclature" section. Our review provides an overview of the methods used to detect anomalies and faults in PV systems, the input data required, and the results achieved. Since our solution relies solely on the measured AC energy of a PV system to detect anomalies, the review focuses more on methods relevant to this type of detection. Initially, existing reviews were identified using Google Scholar with the query :

   "*solar OR photovoltaic OR pv AND anomaly OR fault OR failure AND detection OR diagnosis OR classification AND review*"

   We prioritized recent reviews (published within the last 10 years) and sorted them by citation count for analysis. Further exploratory literature review was conducted by examining references within these reviews. The majority of the references used in this thesis come from four key reviews: El-Banby et al. (2023), Hong & Pula (2022), Mellit et al. (2018), and Rapaport and Green (2021).
4. **Designing Our ADC Method**: After acquiring sufficient knowledge, we were able to design our anomaly detection and classification methodology, which is detailed in the "Methodology – Algorithm Design and Implementation" chapter. This step involved outlining the architecture of the proposed solution based on insights gained from the literature review.
5. **Review of Related Topics for Implementing Our ADC Method**: The final step involved acquiring the technical knowledge necessary to implement our ADC methodology, primarily in Machine Learning and Statistics. This knowledge was obtained through spontaneous research using Google Scholar, online courses, and specialized websites.

# 2 Factors Affecting PV Production

In this section, we review the factors identified in the literature that impact the AC power output of a PV system. We begin with an initial classification of these factors, followed by a detailed presentation of each in Figure 3.

## 2.1 Factors Classification

The classification of factors affecting PV production varies across different studies. For this thesis, we primarily adopt the classification proposed by Iyengar et al. (2018), which divides factors into two main groups: "transient" (normal factors) and "anomalies" (faults and anomalies). Transient factors are further divided into "regional factors," which are common to all systems within a region, and "system-specific factors," which are unique to each PV system.

Building on this principle, we categorize the factors into three distinct groups. This categorization is essential for the design of our detection algorithm, which must distinguish the impact of these three types of factors on energy production.

### 2.1.1 Regional Factors

Regional factors are common to all PV systems in a given area and include environmental and climatic conditions such as cloud coverage, temperature, Global Horizontal Irradiance (GHI), and others (see Figure 3). These factors are relatively stable across a region and influence the overall energy production of different systems in a similar manner.

### 2.1.2 System-Specific Factors

System-specific factors are unique to each PV system and include characteristics such as geographical location, tilt, azimuth angle, module efficiency, and shading from nearby objects (see Figure 3). These factors determine the normal energy output for each system and are generally difficult to modify without significant changes, such as adjusting the system's orientation, removing nearby obstacles, or upgrading components. Therefore, these factors contribute to the baseline performance of a PV system.

### 2.1.3 Anomalous Factors

Anomalous factors are conditions that cause deviations from a PV system's normal output. Examples include line-to-line faults, short circuits, open circuits, mechanical damage, growing vegetation, dirt accumulation, or inverter issues (see Figure 3). These anomalies typically require human intervention to restore the system to its normal state.

Anomalies can be further classified into various sub-categories, based on the intended purpose of the classification. Rajasekar and Pillai (2018) categorize anomalies into physical, environmental, and electrical faults, allowing to choose which detection strategies to use based on the anomaly type. For instance, physical faults may be identified through visual inspection, while electrical faults require specific measurement devices on site. Mellit et al. (2018) categorize anomalies by their temporal impact as "temporal" or "permanent." El Banby et al. (2023) classify them based on when they occur in the lifecycle of a solar system: "infant failures," "mid-life failures," and "wear and tear failures."

Given that our study relies solely on total energy production data, without the use of advanced measured data described in Section B.3 (e.g., specialized measuring devices or drone imagery), we need to find a classification that provides the maximum information possible about potential anomalies to the user, while still being possible to achieve using only the data available to us. Therefore, we have chosen to classify anomalies based on their impact on energy production, which allows us to classify possible anomalies based solely on observed production drops. We have classified the anomalies into two categories:

**By Temporal Evolution :**
Inspired by Iyengar et al. (2018), who categorize anomalies as "no production," "underproduction," and "gradual degradation," we created two categories: "immediate underproduction" (1) and "gradual degradation over time" (2). This classification helps determine the appropriate detection technique to use: immediate anomalies can be detected at a single point in time, while gradual anomalies require an analysis of production decline over time.

**By Impact Level :**
This classification identifies the level at which an anomaly impacts the system: the entire PV system (1), an array (2), a string (3), a module (4), or a cell (5) (see Figure 3 to better understand it). This enables us, by analysing the drop in production compared to the configuration of the PV system, to know at what level the system is affected by the anomaly.

## 2.2 Factors Identification

Based on the findings from the literature and feedback from PV system experts at the partner company and the thesis advisor, several factors have been identified that can reduce the energy production of a photovoltaic (PV) system (El-Banby et al., 2023; Hong & Pula, 2022; Long-Dong et al., 2021; Mellit et al., 2018; Pillai & Rajasekar, 2018; Rapaport & Green, 2021; Triki-Lahiani et al., 2018). These factors are categorized into regional factors, system-specific factors, and anomalous factors, and are presented in Figure 3.

**Regional Factors** include environmental and climatic conditions such as Clear Sky Global Horizontal Irradiance (GHI), air temperature, curtailment by the grid, cloud shading, rain, and wind. These factors affect all PV systems in a given area uniformly and typically have a similar impact on energy production across different systems in the same region.

**System-Specific Factors** are unique to each PV installation. These include the system's coordinates, orientation (tilt and azimuth angle), size (watt peak), curtailment by design, efficiency of modules and inverters, shading from nearby objects (such as trees and chimneys), and normal aging. These factors define the baseline performance of individual PV systems and are generally difficult to change without making significant modifications to the system.

**Anomalous Factors** are abnormal conditions that cause a PV system to deviate from its expected performance. These anomalies can occur at various levels, from the entire system down to individual cells. Examples of such factors include soiling, electrical faults, glass breakage, overheating, and cell degradation. Anomalies may manifest suddenly, such as with electrical faults, or they may develop gradually over time, as with plant growth or delamination. Each factor, as classified and visualized in Figure 3, plays a critical role in influencing the overall energy output of a PV system.

Additionally, the “**signature**” of each anomaly was studied to implement potential classification. For instance, an anomaly caused by plant growth is more likely to occur between April and July and will show a gradual impact over time. However, for the sake of brevity in this report and because these specific details were not directly utilized in the study, they will not be presented here.

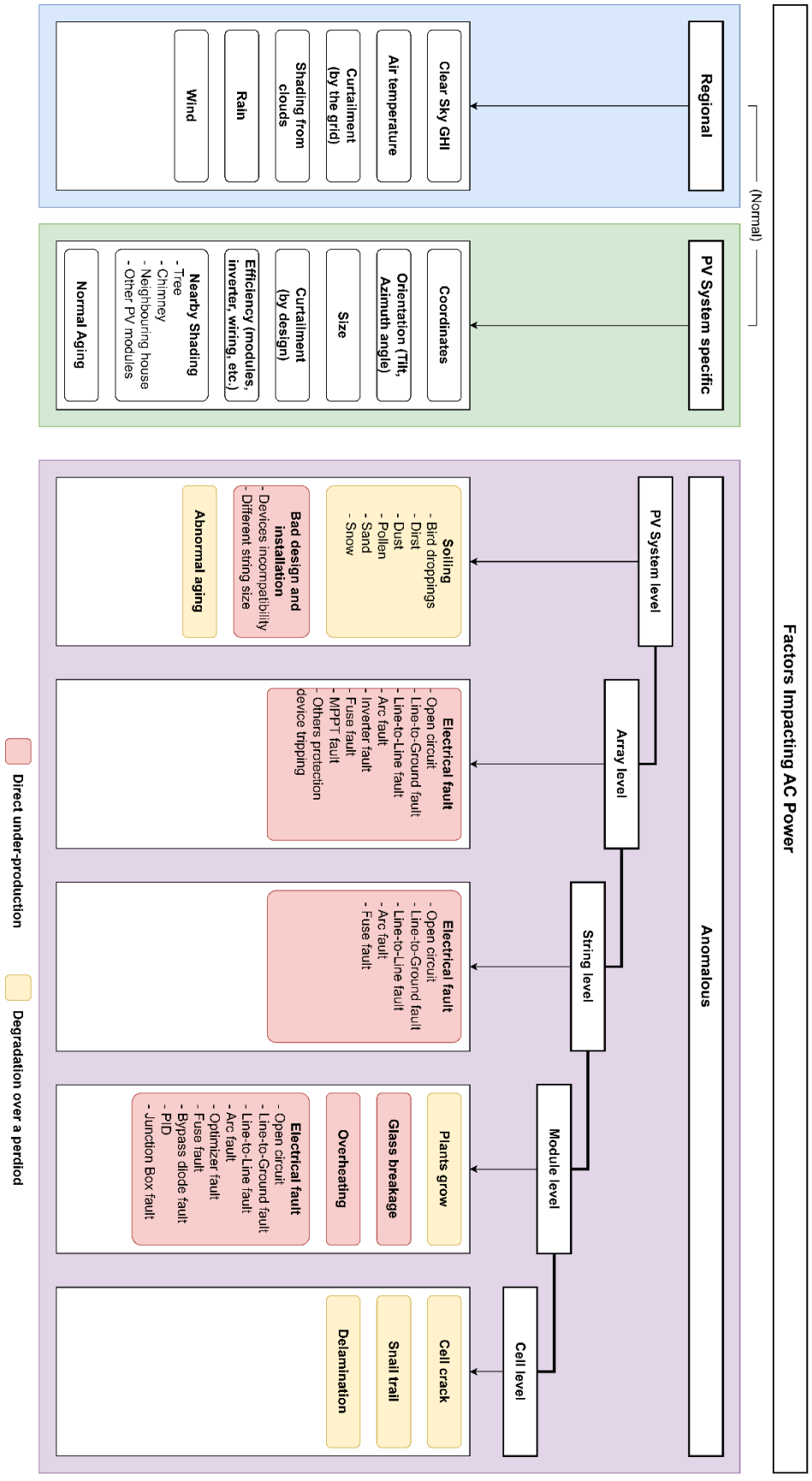



Figure 3 Identification of Regional Normal Factors, PV System Specific Normal Factors, and Anomalous Factors impacting the energy production of a PV system.

# 3 ADC and FDD Methods

Recent literature has extensively researched Anomaly Detection and Classification (ADC) and Fault Detection and Diagnostics (FDD) methods for photovoltaic (PV) systems. This is a broad field with a wide range of methods tailored to different desired outcomes. Fundamentally, the choice of method depends on two key factors: (1) the specific anomalies that need to be detected and (2) the available input data. A balance must be struck between these two considerations. The decision revolves around whether it is more important to accurately detect a specific fault, even if this requires investing resources in additional data collection, or to maximize the use of the data already available. In our case, since the data is limited, the methodology must be designed to effectively detect the maximum number of anomalies using only the available data.

## 3.1 Overview of Methods

To gain an overview of existing detection methods and design our own approach that meets our constraints, a review of the current methods was conducted. However, there are numerous detection methods, each using different types of input data (e.g., drone imagery, inverter data, weather data), different tools (e.g., Machine Learning algorithms, physical simulators), and different strategies (e.g., comparing historical behaviour with current system behaviour, or comparing actual performance to that simulated by a digital twin). Moreover, various strategies, tools, and data types can be combined to create new hybrid methods, resulting in a wide variety of approaches and terminologies in the literature. This diversity makes it challenging to provide a simplified overview of all these methods.

To address this complexity, it was necessary to standardize the structure of each method found in the literature to facilitate summarization. The proposed structure is based on the following four principles:

- **Detection Strategy**: The basic principle underlying the detection process.
- **Input Data**: The type and nature of the data used for detection.
- **Detection Tools**: The technical tools used to implement the detection strategy.
- **Detection Method**: The integration of the above three elements. A unique method is defined by a strategy implemented through specific tools and using a certain type of input data.

This structured approach, which has not been clearly distinguished in other studies, is essential for simplifying the various methods and making them more accessible for understanding.

## 3.2 Detection Strategies

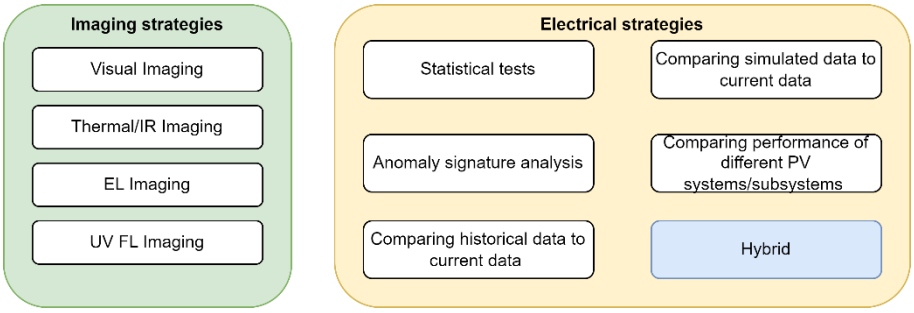



Figure 4 Categories of Strategies for Detecting and Classifying Anomalies

We begin by reviewing the different types of existing strategies for anomaly detection and classification. Broadly, these strategies can be divided into two main categories (Tina et al., 2016):

1. Strategies using imagery.
2. Strategies using electrical-based data.

The strategies presented in Figure 4 and the sections below fall into these two categories.

### 3.2.1 Imaging Strategies

Imaging strategies involve the use of various types of visual and thermal images, such as visual, infrared (IR), electroluminescent (EL), and ultraviolet fluorescence (UV FL) images, as discussed in Section B.3.3. A review by El-Banby et al. (2023) highlights how these methods can accurately detect and localize faults at the array, string, or module level in a PV system. Techniques such as traditional image processing and artificial neural networks, particularly Convolutional Neural Networks (CNNs), are commonly used to analyse these images (Hong & Pula, 2022). However, these methods require significant resources, as data must be captured from an aerial viewpoint, using tools like Unmanned Aerial Vehicles (UAVs or drones), elevated structures, or satellites. Due to their complexity, cost, and little connection with our thesis, we will not further discuss these methods here.

### 3.2.2 Statistical Test

Statistical tests are fundamental to many fault detection strategies and tools, including machine learning algorithms where statistical tests are used for model training. These tests help draw conclusions from data and include methods such as the Independent T-Test, Mann-Whitney Test, Paired T-Test, Analysis of Variance (ANOVA), Kruskal-Wallis Test, Friedman Test, Chi-Squared Test, Cohen's Kappa, and Proportion Z-Test (Rapaport & Green, 2021).

### 3.2.3 Anomaly Signature Analysis

Anomaly signature analysis uses the unique signature of an anomaly in a measurement to identify it. This method is advantageous because it can directly detect anomalies without requiring weather data. However, it requires specialized measurement devices and knowledge of the anomaly signatures, meaning labelled anomalies must be available, which we don't have.

An example is the analysis of the I-V curve, which represents all the current-voltage pairs that the Maximum Power Point Tracker (MPPT) could generate at any time. Deviations from the normal shape of the I-V curve can indicate specific anomalies (see Figure 5). Although this method is effective in laboratory settings and frequently mentioned in literature (Adhya et al., 2022), it is less feasible in real-world applications because it requires the installation of an I-V curve tracer, which is not common.

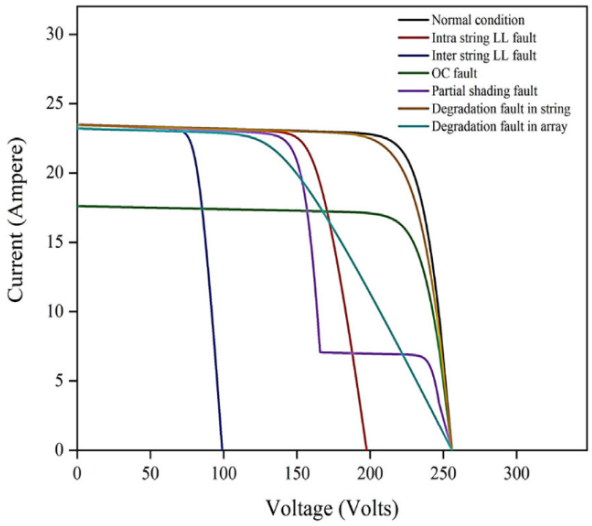



Figure 5 Typical I-V Curve Signatures of Different Anomalies (Adhya et al., 2022)

Other techniques, such as Time Domain Reflectometry (TDR) used by Hong and Pula (2022) and Earth Capacitance Measurement (ECM) discussed by Mellit et al. (2018), can detect various faults in a PV system and pinpoint their exact location, using anomaly signature analysis. However, these methods currently require on-site visits with specialized measurement equipment.

### 3.2.4 Comparing historical data to current data

Another approach to fault detection involves comparing a PV system's historical performance with its current output. Novice system owners often do this intuitively by comparing their current electricity bill or energy production with similar periods in the past. A significant and consistent deviation may indicate a potential issue with the system. Fault detection systems build on this by employing machine learning algorithms and statistical tests to analyse historical data across various time frames, from hours to months, and using multiple parameters. This method assesses the system's current health by identifying anomalies relative to past performance and quantifying any faults. For example, PV GIS use this method, using ten years of historical weather data to predict future production (European Commission, 2024). However, this strategy is relatively inaccurate as many stochastic factors, particularly weather, make it challenging to compare current behaviour with historical data.

### 3.2.5 Comparing simulated data to current data

This strategy involves comparing the PV system's actual performance with simulated data of it. When a significant deviation from the expected value occurs, the fault detection system classifies the PV system as faulty. The simulator can be based on machine learning models or physics-based models to predict expected performance. Weather data is typically included in input of the simulation, as it is one of the most critical parameters influencing PV output.

### 3.2.6 Comparing data of similar PV systems/subsystems

In this strategy, data from comparable elements, such as neighbouring PV systems, or modules from a same array, are compared to identify if one deviates from the overall trend and can thus be classified as anomalous. This approach has the advantage of not requiring weather data, as environmental factors are inferred from the neighbouring systems themselves. Given its relevance to our methodology, this strategy will be discussed in detail, as it is the primary strategy we use.

The core idea here, which aligns with our case, is that the output data of a neighbouring PV system is likely to be correlated with the data of the target system being analysed. This strategy is particularly useful when weather data is unavailable, as the performance data from neighbouring systems can serve as a proxy for sunlight and temperature conditions.

The foundational concept, known as Half Sibling Regression, is inspired by an astronomy study by Schölkopf et al. (2016), which aimed to observe the signal of an exoplanet. The method was used to remove confounding noise from measurement devices that obscured the signal of the observed exoplanet. Multiple observations of the same planet were made and compared to separate the random noise from the common planetary signal. In our case, we're doing the opposite. We want to remove the common factors, such as the weather, to keep only the noise, i.e. the anomalies. This strategy was adapted by Iyengar et al. (2018) for detecting anomalies in PV systems by using the production data from neighbouring PV systems, and later by Feng et al. (2020), who compared the production of each PV module within the same system to detect faulty modules.

La Figure 6 illustrates how Half Sibling Regression works.

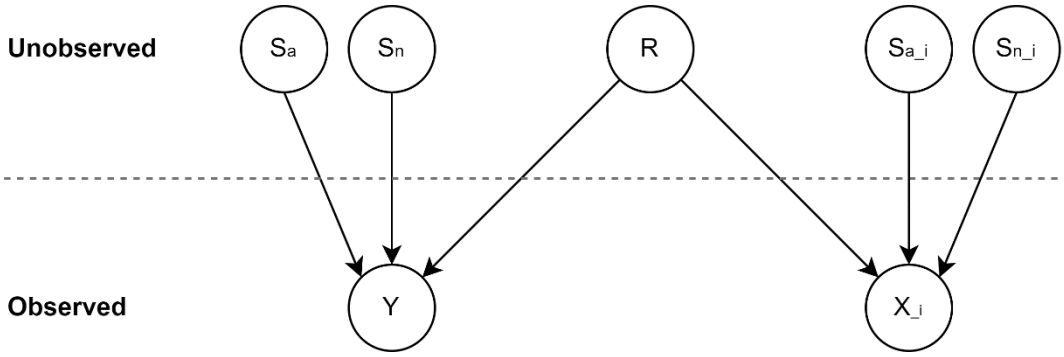



Figure 6 Graphical model representation of the Half Sibling Regression strategy

**Observed**:
$Y$ : Measurements from the Target PV System
$X_i$ : Measurements from Neighbouring PV systems
**Unobserved**:
$S_a$ : System-Specific Anomalous factors
$S_n$ : System-Specific Normal factors
$R$ : Regional factors
$S_{a_i}$ : System-Specific Anomalous factors of each Neighbouring PV system
$S_{n_i}$ : System-Specific Normal factors of each Neighbouring PV system

In this model, $Y$ represents the measured value of the PV system we want to analyse. As discussed in the Section "Factors Classification" in B.2, this value is influenced by system-specific normal factors $S_n$ (such as azimuth or shading from a tree), anomalies $S_a$ (such as a disconnected module), and regional factors $R$ (such as cloud cover). In summary, $Y$ is given by:

$$Y = f(S_a) + f(S_n) + f(R)$$

Our goal is to find $f(S_a)$, the component of the observed production $Y$ that is influenced by anomalies:

$$f(S_a) = Y - f(S_n) - f(R)$$

The observed value $Y$ is known. The system-specific normal factors $f(S_n)$) can be inferred in various ways, as explained in Chapter C : "Methodology – Algorithm Design and Implementation". However, the influence of regional factors $f(R)$, such as irradiation, cloud cover, or temperature, is unknown in our case because we do not have measurements for these variables. These factors significantly affect $Y$, making it impossible to detect anomalies $f(S_a)$ without knowing $R$. This is the essential insight of the Half Sibling Regression approach.

Since $R$ influences both $Y$ (the observed value for the target system) and $X_i$ (the data from neighbouring systems), it is possible to approximate $f(R)$ using the data from neighboring systems $X_i$:

$$f(R) \approx E[f(R)|X_i]$$

This gives us the equation:

$$f(S_a) \approx Y - f(S_n) - E[f(R)|X_i]$$

The function $E[f(R)|X_i]$ represents a supervised regression problem that can be addressed using various machine learning models, which will be discussed in the "Detection Tools" Section B.3.4.

It is important to note that each neighbouring system $X_i$ is also affected by its own system-specific normal factors $S_{n_i}$ and anomalous factors $S_{a_i}$. Therefore, it is essential to account for the normal factors $S_{n_i}$ of each neighbouring system to enhance the estimation of the regional factors $R$. However, the anomalous factors $S_{a_i}$ of neighbouring systems cannot be considered, as they are unknown. It is important to understand that these unknown anomalies can reduce the accuracy of the estimation.

Regarding the accuracy of this strategy, previous research by Shimshon and Mike (2021) used multiple PV systems on a university campus to compare the performance of three methods: using data from neighbouring PV systems (1), using weather stations integrated with PV systems (2), and using local weather stations (3). The results, shown in Figure 7, indicate that integrated weather stations outperform the analysis of neighbouring systems, while regional weather stations provide less accurate results. Additionally, Iyengar et al. (2018) achieved a Mean Absolute Percentage Error (MAPE) of 7.81% using 88 neighbouring PV systems, indicating that, on average, the regressor's estimate deviates by ±7.81% from the actual produced energy. Feng et al. (2020) achieved a MAPE of 2.98% by comparing the production of each module within the same PV system.

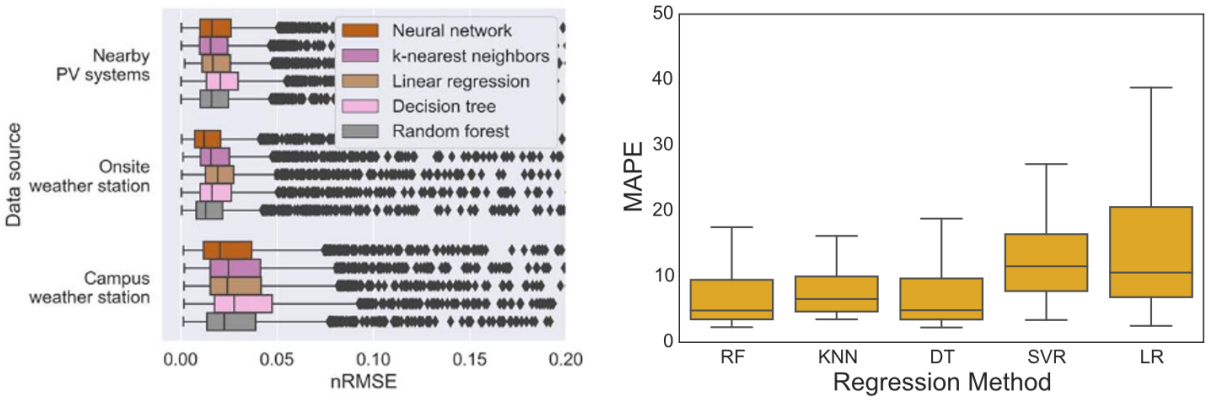



Figure 7 Performance of Other Studies in Estimating Energy Output Using Data from Neighbouring Systems. Left: (Rapaport & Green, 2021) Right : (Iyengar et al., 2018)

### 3.2.7 Hybrid

An important point to understand about anomaly detection strategies is that it is possible to combine several strategies to create a hybrid method. According to Li et al. (2021), there are two main types of architectures for detecting and classifying anomalies, as illustrated in Figure 8:

1. **Direct Method**: This is an end-to-end approach where a classifier, such as a machine learning model, takes a certain type of input data and directly outputs the classes of different anomalies. This method typically requires labelled input data, meaning that the specific fault that has occurred is known beforehand.
2. **Step-by-Step Method**: This approach involves using multiple strategies in sequence, producing intermediate results at each step. For example, in our case, where we use the produced power as input data, we might first use a regressor to estimate the expected production. The loss between the estimated and measured values is then used by a second model to detect and classify anomalies. This method offers more control through intermediate results and reduces the "black box" effect associated with direct methods.

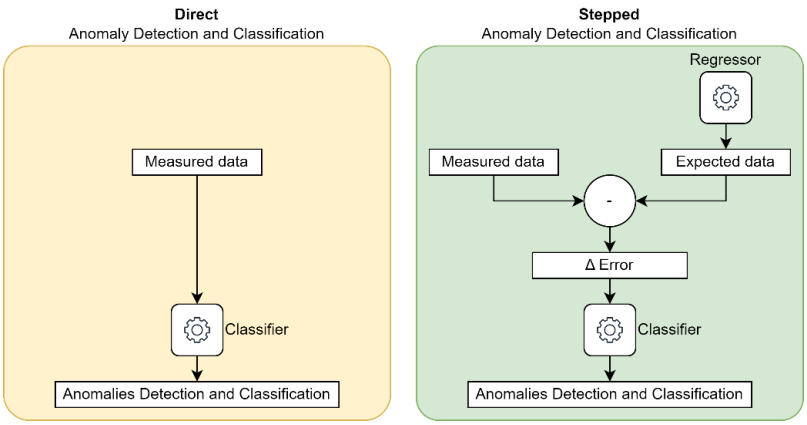



Figure 8 Difference in strategy between a Direct and a Step-by-Step architecture.

## 3.3 Input Data used in research papers

In this section, we briefly present the different types of input data used in various studies to detect and classify anomalies in PV systems. Understanding the input data types helps us identify which methods are suitable for specific data types and highlights the most effective data for Anomaly Detection and Classification (ADC). Indeed, input data is a key factor in selecting the appropriate detection method.

By understanding the types of input data and their relevance to ADC methods, we can better design our detection strategies to maximize accuracy and effectiveness in identifying anomalies in PV systems.

Following the classification proposed by Li et al. (2021), input data is categorized into three main groups: Environmental Data (1), Electrical Data (2), and Images (3). Table 1 provides an overview of the types of data used in different studies and the frequency with which each type is employed. This table shows that environmental data, such as solar radiation and temperature, are commonly used. Additionally, electrical data from the inverter, like current and voltage readings or the I-V curve of the Maximum Power Point Tracker (MPPT), are also frequently utilized. However, few studies have used the total AC output power to detect anomalies, especially without incorporating meteorological data.

Table 1 Overview of data used in ADC methods, with number of paper using them (adapted from Li et al. (2021))

| | Symbol | Name | Nbr of papers |
|---|---|---|---|
| **Environmental data** | $\boldsymbol{G}$ | **Radiation** | **22** |
| | $T_m$ | Module temperature | 14 |
| | $T_A$ | Air temperature | 7 |
| | $v_{Wind}$ | Wind speed | 2 |
| | $d_{Wind}$ | Wind direction | 1 |
| | $V_{Rainfall}$ | Rainfall volume | 1 |
| | $h_{Snowfall}$ | Snowfall height | 1 |
| **Electrical data** | $I-V\ Curve$ | I-V curve of the MPPT | 11 |
| | $V_{MPP}$ , $I_{MPP}$ , $P_{MPP}$ | Output DC Voltage, Current, and Power from the MPPT | 25 |
| | $V_{OC}$ | Voltage in Open Circuit condition | 9 |
| | $I_{SC}$ | Short Circuit max current | 8 |
| | $\boldsymbol{V_{AC}}$ , $\boldsymbol{I_{AC}}$ , $\boldsymbol{P_{AC}}$ | **Output AC Voltage, Current, and Power** | **3** |
| | $ECM$ | Earth Capacitance Measurement | 2 |
| | $TDR$ | Time Domain Reflectometry | 2 |
| | ... | Others available data from the Inverter | 3 |
| | ... | Others available data from the PV Module Optimizers | 1 |
| **Image** | $Vis$ | Visual | 10 |
| | $IR$ | Infrared/Thermal | 6 |
| | $EL$ | Electroluminescence | 12 |
| | $UV\ FL$ | Ultraviolet Fluorescence | NA |

**Importance of the environmental data**

Environmental data plays a critical role in ADC methods. According to Iyengar et al. (2017), incident radiation (G) has the most significant impact on the performance of a PV system. This variable is highly stochastic and cannot be predicted accurately, mainly due to varying cloud cover. Consequently, most detection techniques in the literature include this variable as an input, often obtained from on-site sensors, satellite imagery, or local weather stations. Without meteorological information, whether provided directly or inferred from other variables, Iyengar et al. argue that it is nearly impossible to detect anomalies using only the produced energy, as any anomaly-related variation in production is likely to be masked by the noise caused by weather fluctuations. De Benedetti (2018) supports this by showing the relationship between energy production and radiation alone, demonstrating a strong linear correlation between the two variables (see Figure 9). This data can even be used directly to detect anomalies, as evidenced by the red outliers on the graph.

Temperature is another crucial variable, as every 1°C increase above 25°C reduces the efficiency of a PV cell by 0.5% (Iyengar et al., 2017).

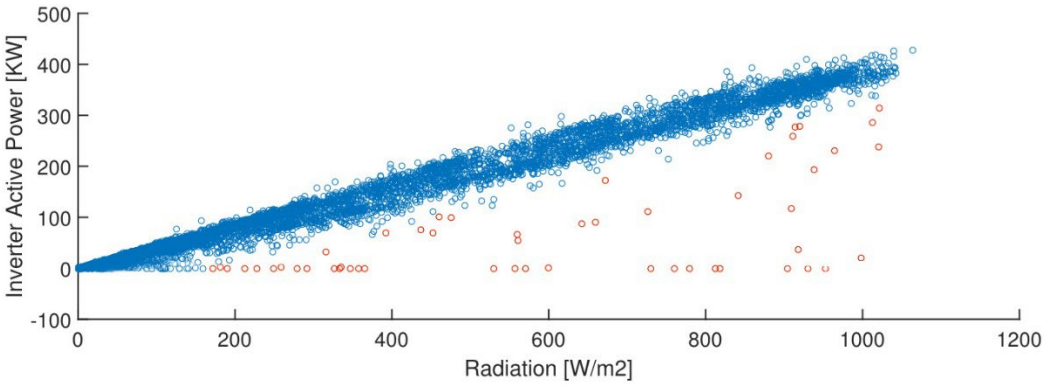



Figure 9 Linear relation between Radiation and Power. Each dot is an observation. Outliers classified as anomalous are highlighted in red (De Benedetti et al., 2018)

## 3.4 Detection & Classification Tools

To understand the various tools used in the literature for anomaly detection and classification, we will review them in this section. As previously discussed, these tools can be applied with different types of input data and to implement different strategies. The ideal method should be designed based on the available data and the desired outcome. However, due to the conciseness required in this report, we will not delve into the specifics of how each tool works or how they have been applied in various studies (including the data used, strategies employed, and results obtained). An in-depth review has been conducted internally to gather all the necessary knowledge for designing our own methodology, which is detailed in Chapter C, “Methodology – Algorithm Design and Implementation”.

The tools for detection and classification can be categorized as follows:

- Physics-Based Model :
  - White-Box
  - Black-Box
- Rule-Based Models
  - White-Box
- Machine Learning-Based Models
  - Supervised Learning
  - Unsupervised Learning
  - Semi-Supervised Learning

For a better understanding of the distinctions between these tools, it is recommended to refer to the sections "Definition of White-Box vs. Black-Box Tools" and "Definition of Machine Learning" in the "Glossary and Nomenclature" chapter. This will provide foundational knowledge about the principles underlying these tools and the differences between the types of learning.

### 3.4.1 Physics-Based White Box Models

A physics-based white box model, also known as a Digital Twin, is a digital replica of a PV system based on physical principles designed to replicate the data that the actual system should produce. By comparing the Digital Twin’s data with real-world data, anomalies can be detected (Bashir et al., 2019). However, creating a Digital Twin requires extensive information about the PV system, such as tilt, azimuth, losses, surrounding shading, module characteristics, inverter specifications, and weather data. Iyengar et al. (2017) note that generally, neither the system owner nor the installer possesses all this data, and even if the data is available, developing a reliable physical model is complex and time-consuming even for an expert.

### 3.4.2 Physics-Based Black Box Models

To address these challenges, Chen et al. (2018) propose a physics-based black box model. While the core of this model is still based on physical principles, the static parameters of the model are not manually provided by the user but are instead inferred using historical production data. By providing historical weather data and past measurements, the model automatically determines optimal parameters, such as tilt, azimuth, and losses, to minimize the simulation error. Bashir et al. (2019) use this method and require only the geographic location of the system, historical production data, and historical irradiation and temperature data to estimate other static parameters of a PV system, such as tilt, azimuth, or losses. The advantage of this model over a Machine Learning Black Box (discussed in the next section) is that the results are more interpretable for a human. This is because the model's structure is still based on physical formulas, and only tangible parameters are inferred from the historical data.

### 3.4.3 Rule-Based White Box Classifier

Rule-based white box classifiers detect and classify anomalies using predefined logical rules. These methods often rely on threshold-based Fault Detection and Diagnostics (FDD). Indicators sensitive to faults are first extracted from measured data, and then rules combining one or multiples indicators are defined to detect and classify desired anomalies. According to Z. Chen et al. (2019), these algorithms have proven to be effective, they require manual extraction of fault indicators and determination of thresholds, which is time-consuming and limits generalization performance.

### 3.4.4 Supervised Machine Learning Models

Supervised Machine Learning models can be categorized into two main types: Regressors and Classifiers. The first type, Regressors, are models designed to predict a continuous value. The second type, Classifiers, are models that predict the category or group to which an observation belongs.

In the context of anomaly detection, a regressor is not typically used to directly identify an anomaly. Instead, it is employed to generate an intermediate value that serves as an indicator for further processing. On the other hand, classifiers are used to directly analyse the input data and automatically classify the type of anomaly detected.

Both types of models are supervised, meaning they require training data that includes the information the model needs to predict. For a regressor, this means providing the continuous value it needs to predict. For a classifier, a discrete class—such as the type of anomaly of interest—is provided. Additionally, the training data must include all the relevant information, known as features, that will help the regressor to make its predictions accurately.

Various supervised models are tested in the literature for anomaly detection across different scenarios. These are widely used tools, and certain types are frequently cited. The commonly used models will be presented shortly here. It is also worth noting that these tools can generally be used both as regressors and classifiers.

**Linear Regression (LR)**

Linear Regression is a statistical method that assigns weights to each feature to find a linear relationship between the features and the target value. It can handle multiple features to predict a continuous outcome based on the linear equation:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

A simple example is illustrated in Figure 10, which shows a single feature regressor; however, the model can accommodate $n$ features.

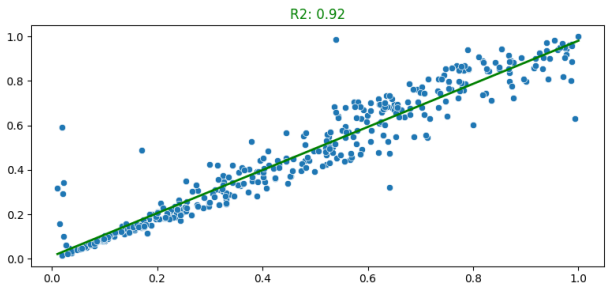



Figure 10 Illustration of a Single Linear Regression

Linear Regression often produces the worst results in anomaly detection for PV systems (Iyengar et al.,

2018). Also, this model is sensitive to missing values in the dataset, which can significantly impact its performance and universality, as it requires complete data for all features to function effectively.

**Random Forest (RF) / Decision Tree**

Random Forest is an ensemble learning model that uses multiple Decision Trees trained in parallel with bootstrapping techniques to reduce variance and improve prediction accuracy (see Section 4.4 on this chapter about Bootstrapping). A Decision Tree is composed of nodes, branches, and leaves, where each node represents a decision point that splits the observations based on the value of a specific feature. The selection of features and the optimal splitting value at each node are determined during the training phase, to reduce prediction error.

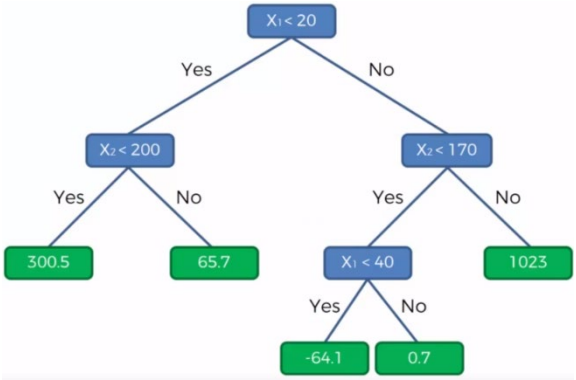



Figure 11 Illustration of a Decision Tree

Random Forest is one of the most widely used machine learning techniques for fault classification in PV systems (Hong & Pula, 2022). It is favoured for several reasons: minimal data preprocessing is required, it is less prone to overfitting due to the averaging of multiple trees, and it can handle missing values effectively. Additionally, Random Forest models are relatively easy to interpret, which makes them a preferred choice for anomaly detection in PV systems (Mansurova, 2023). Many anomaly detection methods rely on this model due to its robustness and versatility.

**Artificial Neural Network (ANN)**

ANNs are versatile models that serve as foundational building blocks for a wide range of customized models. They are largely studied in fault detection over the past few years. Thanks to their extensive customisation, there are many different types of ANN model, each suited to different tasks. However, no study has been found that is closely related to our problem. Generally speaking, other Machine Learning models are sufficient. An ANN would make the design and understanding of the model unnecessarily complex. We will, however, come back to some studies that could inspire us in the design of our own method.

1. **Feedforward Neural Networks (FNNs)**: These models perform regression or classification based on the input provided at a given time without considering the relationships between observations. They are straightforward but may lack the capacity to detect anomalies that develop over time.
2. **Recurrent Neural Networks (RNNs)**: These models have a memory of previous inputs and use the historical context to inform current predictions. RNNs are particularly effective in time-series analysis where strong temporal correlations exist between observations. For example, in anomaly detection for PV systems, RNNs can be used to identify gradual degradation over time, as depicted in Figure 3. Within RNNs, various models such as Long Short-Term Memory (LSTM) networks, optimized LSTMs, and ConvLSTMs (which combine Convolutional Neural Networks with LSTMs) have been employed for solar energy prediction tasks (Elsheikh, Katekar, et al., 2021; Elsheikh, Panchal, et al., 2021; Ibrahim et al., 2020). These models are promising for detecting temporal anomalies in PV systems.
3. **Graph Neural Networks (GNNs)**: GNNs are another category of ANNs that can be particularly useful in detecting anomalies in PV systems. They leverage relational graph representations to uncover relationships between different observations. For instance, Van Gompel et al. (2023) used a GNN with voltage and current data from neighbouring PV systems over the past 24 hours, along with their geographical distances, to classify anomalies. GNNs are well-suited for finding complex relationships in interconnected data, making them a compelling choice for advanced anomaly detection.

**Support Vector Machines (SVM)**

SVM is a supervised machine learning model that works by drawing a hyperplane or a set of hyperplanes in a multi-dimensional space to separate different classes of data. Initially designed for binary classification, SVMs have been extended to multi-class classification (multi-class SVM) and regression tasks (SVM Regressor). In binary classification, SVM determines the optimal boundary (hyperplane) that maximizes the margin between the two categories. On example is illustrated in Figure 12.

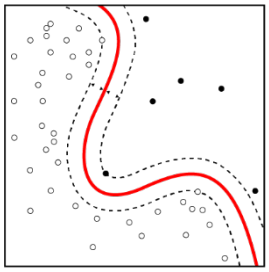


Figure 12 Illustration of a SVM

SVMs are foundational models in machine learning and have been employed in anomaly detection for PV systems. They are particularly useful for distinguishing between normal and faulty operations by identifying the decision boundary between normal and anomalous states. However, SVM models are not inherently designed to handle missing values in the input data, which limits their effectiveness. Examples of using SVM in fault detection include applications in PV systems by Benninger et al. (2019), Iyengar et al. (2018), and Jiamin Sun et al. (2019).

**k-nearest neighbours (KNN)**

k-Nearest Neighbours (kNN) is a supervised learning algorithm used for both classification and regression tasks. In kNN, the classification of a new data point is determined based on the majority class among its 'k' nearest observations, in each feature space. The 'distance' between data points is commonly calculated using the Euclidean metric. This approach allows kNN to classify a dataset into different categories, including distinguishing normal observations from anomalous one.

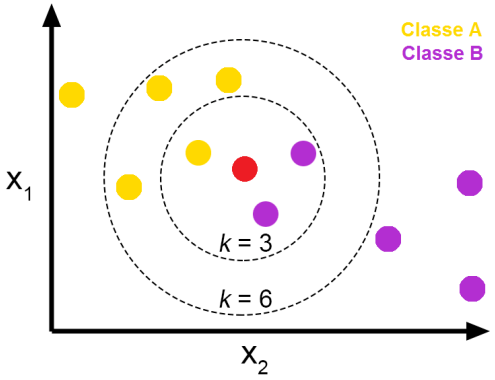



Figure 13 Illustration of K-Nearest Neighbours

In the context of anomaly detection for photovoltaic (PV) systems, kNN is used to identify deviations from expected patterns. The algorithm can effectively classify anomalies such as short circuits, open circuits, or partial shading by analysing the differences between simulated and actual data. For example, Benninger et al. (2020) utilized kNN to detect faults in PV systems by comparing real-time measurements with simulated expected values, triggering specific error messages based on the identified deviations. While kNN is relatively simple and easy to implement, it requires a complete dataset as it does not inherently manage missing values.

### 3.4.5 Un-supervised Machine Learning Classifier

Unsupervised machine learning classifiers, also known as clustering methods, are techniques used to categorize data without predefined labels. Unlike supervised learning, where models are trained on labelled data, unsupervised classifiers independently identify patterns and group similar observations into clusters. This method is particularly useful in scenarios where labelling is impractical or costly. As shown in Figure 14, clustering groups data points based on their similarities, determined by their input features. For clustering to effectively identify similar anomalies, the impact of each anomaly type must be clearly visible on the input data.

Several unsupervised clustering models are commonly used, including density peak-based clustering, k-means clustering, unsupervised clustering with probabilistic neural networks (PNN), Isolation Forest, and dilation and erosion-based clustering. These methods vary in how they measure similarity and form clusters, but they share a common goal of organizing data into groups that reveal hidden patterns or anomalies.

However, no studies have been found in the context of anomalies classification, that were using a un-supervised classifier. In every study, the type and origine of the anomalies were labellised.

Nevertheless, some unsupervised models, such as the Isolation Forest, have shown promise. The Isolation Forest is a decision tree-based model that isolates observations that looks statistically different

as all the other observations. Observations that are more easily isolated are considered anomalies. According to Hariri et al. (2021), Isolation Forest could be particularly effective in the unsupervised detection of anomalies in PV systems, as it requires no labelled data and can adapt to various types of anomalies

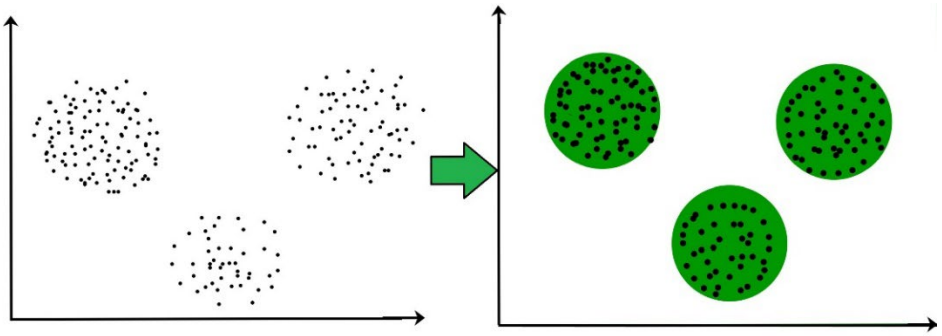


Figure 14 Illustration of clustering

### 3.4.6 Semi-supervised Machine Learning Classifier

Semi-supervised machine learning classifiers combine elements of both supervised and unsupervised learning. In this approach, only a small portion of the data is labelled, while a large amount of unlabelled data is also available. The labelled data is used to guide the learning process, helping the model to identify patterns and classify the unlabelled data more accurately. This method is particularly useful when labelling a full dataset is too costly or time-consuming.

In the context of anomaly detection and classification for photovoltaic (PV) systems, semi-supervised learning can be advantageous. For example, Zhao et al. (2013) introduced a semi-supervised technique in their paper "Graph-based Semi-supervised Learning for Fault Detection and Classification in Solar Photovoltaic Arrays". They employed a method called label propagation, which involves creating a graph representing relationships between observations (as in the Graph Neural Networks seen previously). In this graph, the known labels of a few observations are propagated to their similar, unlabelled neighbours.

# 4 Related Topics for Implementing Our ADC Method

In this section, we will review several technical principles necessary for implementing our Anomaly Detection and Classification (ADC) method. These include:

- Data Normalisation
- Data Filtration
- Metrics for Evaluating Model Accuracy
- Data Bootstrapping
- Model Validation and Testing Principles (Train/Validation Set vs. Cross-Validation vs. Out-Of-Bag Validation)

## 4.1 Input Data Normalization

One of the first crucial steps in data analysis is the normalisation of input data, which typically involves scaling data to a range between 0 and 1. Normalising input data is beneficial in both statistical calculations and as input for machine learning models. Some of the key advantages of normalisation include (Wei, 2024) :

- **Improving Model Accuracy**: Many machine learning models perform better with normalised data, as it allows them to be more accurate and efficient. Normalisation ensures that features with larger scales do not dominate the model's learning process.
- **Facilitating Model Training**: If the specific characteristics of each input feature are already known, it is preferable to normalise these features by accounting for their characteristics before inputting them into the model. This pre-processing step eases the model's training process as the model does not need to independently learn these characteristics and their impact.
- **Enhancing Compatibility between Models**: Some machine learning models, especially those that involve distance calculations, such as k-nearest neighbours (KNN) or Support Vector Machines (SVM), require normalised data to function correctly because they are sensitive to the magnitude of data. To ensure flexibility in model selection, normalising the data is advisable.

In the context of photovoltaic systems, measurements depend on system-specific factors, such as peak power, tilt, azimuth, location, or equipment (see Figure 3). This results in each system having different scales and behaviours, posing a challenge for developing universal ADC algorithms capable of operating across various PV systems without manual model adjustments. By implementing a normalisation process that considers and adjusts for the specific characteristics of each system, we facilitate the training of any machine learning model using these data. This approach eliminates the need for the model to deduce these characteristics and their impact on the measured value independently.

Regarding the techniques for implementing normalisation, different methods may be applicable depending on the type of input data discussed in Section B.3.3. The principles for normalising an I-V curve, an infrared image, or a power measurement will differ drastically. Since this study only uses the total AC energy of the system, we will focus on the normalisation techniques specific to this type of data.

There are two main techniques for normalisation in our case: (1) using known parameters and formulae, and (2) learning from historical data. For the first technique, a simple approach would be to divide the measured produced energy by the system's peak power (kWp), thereby standardising calculations for systems of different sizes. For more precise normalisation, additional parameters and formulae can be incorporated in the normalization. The second technique involves learning a system's specific characteristics, such as mean value, seasonality, and trends, from its historical data and then normalising based on these traits.

In our study, we employ a normalisation approach based on the first principle, using known parameters of each system provided by the partner company. The explanation and implementation of our normalisation process are discussed in Section C.5, “Normalization”.

## 4.2 Input Data Filtration

Once the data is normalised, a filtration process may be necessary. Fault detection methods must be able to filter out noise from the data being input. According to Rapaport & Green (2021), filtering noise from PV data can improve fault detection accuracy by 2 to 5 times. Data filtration is highly dependent on the type of input data and can be used to remove noise, extract features, handle outliers, deal with missing values, repair fragmented data, determine parameter compatibility, and map relationships (Hong & Pula, 2022).

Our filtration process is discussed in Section C.6, “Filtering”.

## 4.3 Accuracy Metrics

An essential aspect of using predictive methods, such as regression or classification, is measuring the accuracy of the model. This allows us to assess the improvement of the model during its refinement, compare different models, and benchmark our results against other studies.

### 4.3.1 Regression metrics

The choice of regression metric typically depends on the nature of the target output value $y$. The main metrics found in machine learning literature are listed in Table 2 and are described below.

- **Mean Square Error (MSE)**: MSE is commonly used internally within machine learning models during training because it allows for faster model training by squaring the errors. However, this can be problematic in the presence of outliers, as they will disproportionately affect the result due to the squared error. This is the case in our study, as we have some measurement errors with strong outliers. To mitigate this, the **Mean Squared Log Error (MSLE)** can be used. However, these metrics are not easily interpretable by humans since they do not correspond to any real-world values due to the squaring of errors.
- **Root Mean Squared Error (RMSE)** and **Mean Absolute Error (MAE)** are more interpretable as they provide errors on the same scale as the original data and do not amplify the influence of outliers. However, the scale of these metrics is relative to the scale of the measurements, making it impossible to compare results between two different PV systems with different peak powers.
- **Mean Absolute Scaled Error (MASE)** partially addresses this issue by scaling the error relative to a given value, such as the system's peak power (kWp). This metric indicates the error of the estimate relative to the system's peak power, so the error value is independent of the system. It's worth noting that if the target output value $y$ has also been normalized, as seen previously, the scaling of the metric is not necessary, and the Mean Absolute Error can be used to compare two systems-
- To understand the error in the estimate relative to the actual measured value, the **Mean Absolute Percentage Error (MAPE)** is often used. This metric is widely employed in anomaly detection and classification studies (Iyengar et al., 2018) because it provides a direct percentage difference between the estimated and measured values. However, dividing by $y_i$ means that the metric can become very large if $y_i$ is small or close to zero, or if the error magnitude is large compared to the measurement value ($|y_i - \hat{y}| \gg y_i$). This often occurs in our case, such as when a system does not produce energy on a cloudy day or during snowfall. To mitigate this issue, a minimum value can be set for $y_i$ in the denominator, to avoid division by 0.

The MAPE with a minimum $y_i$ value in the denominator is used in this thesis.

Table 2 Regression metrics. $y_i$ is the measured value, $\hat{y}$ is the expected value, n is the number of observations, $S_i$ is a scaling value

| | |
|---|---|
| **Mean Square Error** | $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2$ |
| **Mean Squared Log Error** | $MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(y_i + 1) - \log(\hat{y} + 1))^2$ |
| **Root Mean Squared Error** | $RMSE = \sqrt{MSE}$ |
| **Mean Absolute Error** | $MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}\rvert$ |
| **Mean Absolute Scaled Error** | $MASE = \frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{y_i - \hat{y}}{S_i}\right\rvert$ |
| **Mean Absolute Percentage Error** | $MAPE = \frac{1}{n}\sum_{i=1}^{n}\left\lvert\frac{y_i - \hat{y}}{y_i}\right\rvert$ |

### 4.3.2 Classification metrics

To evaluate the performance of a classifier, which determines whether the model assigns the correct classes to anomalies, a **Confusion Matrix** is typically used. This matrix provides the numbers of True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Accuracy and precision can then be calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

However, to compute these metrics, labelled data indicating when anomalies occurred is essential. In our case, such labelled data is not available, making it impossible to calculate the classifier's accuracy directly. To address this, we will simulate anomalies to generate labelled data.

## 4.4 Bootstrapping

Bootstrapping is a statistical technique used to reduce the variance of any statistical calculations or machine learning models and provide additional measures of model quality, such as bias, confidence intervals, and more.

Figure 15 illustrates the bootstrapping process. Bootstrapping involves creating multiple new datasets (bootstrap samples) of the same size as the original dataset by randomly selecting observations from the original dataset with replacement. Each bootstrap sample is then used to train a separate machine learning model. After training, model metrics—such as those discussed in the previous section—or predicted values are computed for each model. The distribution of these results across all bootstrap samples is then analysed to estimate prediction accuracy, variance, and error. The main idea behind bootstrapping is to reduce the variance of a model by introducing randomness into the dataset.

Ensemble machine learning models, such as Random Forests, inherently use bootstrapping. In a Random Forest, each decision tree is trained on a different bootstrap sample, randomly selected from the original dataset. This ensures that each tree is trained on a slightly different dataset, which reduces the overall variance of the model's predictions.

By applying bootstrapping, we can enhance the robustness and reliability of our anomaly detection and classification models, ensuring more consistent performance across different datasets.

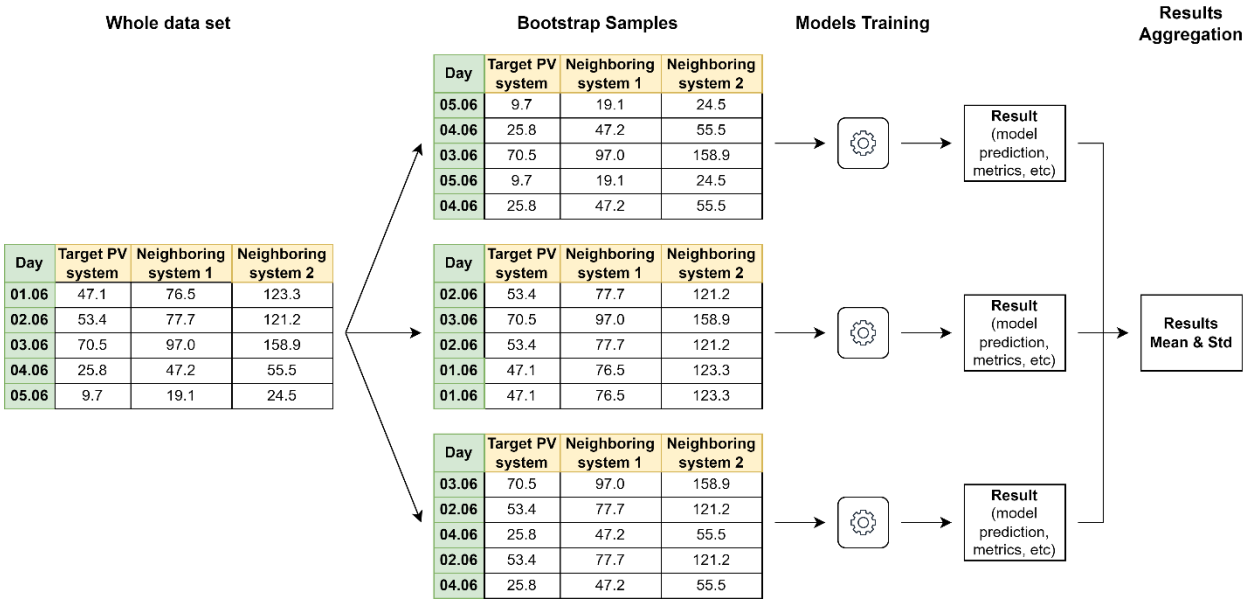



Figure 15 Illustration of the Bootstrapping Process

## 4.5 Model Validation and Testing

When using machine learning models, it is essential to assess their performance without introducing bias. To achieve this, the data is typically divided into three distinct sets: **the training set (1), the validation set (2), and the testing set (3).** The training set is used to train the model, the validation set is used to select the best model and tune its hyperparameters, and the testing set is used only at the end to evaluate the final performance of the model. This distinction is crucial because a model might overfit the training data—meaning it performs exceptionally well on the data it was trained on but poorly on new, unseen data. This is where the validation set helps. However, even the validation set can introduce bias, as the model and its parameters might be fine-tuned specifically to perform well on this set. Therefore, the final performance is evaluated on the testing set, which has never been used during the model development process.

By employing these validation and testing strategies, we ensure a robust assessment of our models, providing confidence in their ability to generalise to new, unseen data.

### 4.5.1 Testing Set Creation

For evaluating the performance of a machine learning model, the testing set typically comprises 20% of the total observations. This set can be created by either randomly selecting observations from the entire dataset (1) or by taking the last $n$ data points (2). In the field of anomaly detection, the second strategy is generally preferred (Rapaport & Green, 2021). This is due to two factors. First, it closely mirrors a real-world scenario where the model is trained up to a certain date and then used to predict future data. Second, it limits the risk of correlation between observations over time. Time series data (i.e. data linked to a specific point in time), often exhibit strong correlations between temporally proximate observations. If an observation in the testing set is temporally close to an observation in the training set, the machine learning model might overfit by taking advantage of its knowledge of the training set to obtain unusually good results on the test set (scikit-learn, 2023, para. “Cross validation of time series data”). For example, weather conditions on one day, such as snowfall, could influence energy production the next day, creating a correlation between days. Thus, a temporal separation between the testing and training sets is necessary.

We will therefore select the last 20% of observations for the test set, and the other 80% for the training and validation set.

### 4.5.2 Model Validation

For model validation, three main approaches are used in the literature: the Validation Set approach (1), the K-Fold Cross-Validation approach (2), and the Out-of-Bag (OOB) Validation approach (3). The first two are illustrated in Figure 16, and the last can be visualised using Figure 15.

- **Validation Set Approach**: This involves setting aside a proportion of the initial observations for validation and the rest for training, similar to the testing set strategy seen previously. However, this simple technique has drawbacks. First, the result depends heavily on which observations are selected for the validation set, potentially leading to model tuning specifically for this subset. Additionally, it reduces the size of the training set, which can be problematic if the number of observations is limited.
- **K-Fold Cross-Validation**: To overcome these limitations, K-Fold Cross-Validation is often used. Here, the dataset is split into $k$ folds. One fold is reserved for validation, and the remaining $k - 1$ folds are used for training. This process is repeated $k$ times, each time with a different validation set. The average metric across all $k$ models is considered the performance of the final model on the validation set. This ensures that the model's performance represents the entire dataset. Again, as explained earlier for the testing set, this cross-validation should not randomly select observations to create the groups but should keep temporally close observations in the same group. On the recommendation of our advisor, folds with two weeks of data were used to ensure no meteorological correlation between the training and validation sets. The disadvantage of the cross-validation method is that it requires training $k$ machine learning models, which is time-consuming.
- **Out-of-Bag (OOB) Validation**: This approach leverages bootstrapping, discussed in Section B.4.4. During the random selection of observations with replacement to create bootstrap samples, approximately 36.8% of the observations are statistically not selected for training (Choudhary, 2021). These unselected observations are then used for validation. This allows for using 100% of the data for model training and eliminates the need to train $k$ models as required in cross-validation. Although several sources suggest this could replace cross-validation entirely (Adkins, 2023; Leo Breiman & Adele Cutler, n.d., para. “The out-of-bag (oob) error estimate”), in our case, the problem of correlation between temporally close observations remains. Since bootstrapping is performed randomly, we cannot guarantee a temporal separation between the training and validation sets.


Figure 16 Illustration of the Validation Set and Cross-Validation Approaches for Validating a Machine Learning Model.

# C. Methodology – Algorithm Design and Implementation

This chapter details our methodology, or in other words, the design and implementation of our algorithm. We will begin by presenting the materials used, including the input data (1) and the software tools employed (0). Following this, we will explain why the detection strategy based on neighbouring systems' measurements was chosen (3), and then outline the final design of our algorithm (4). Finally, we will provide a detailed explanation of each implementation step, including data normalization and filtering, the Half Sibling Regression, anomaly detection and classification, the user interface developed, and the model storage system (0-13).

## 1 Input Data

The input data used in this study consists of real-world field data provided by a partner company, rather than simulated or laboratory data, which is often the case in the research studies reviewed. This data is divided into two parts: the **alternating (AC) energy produced in kWh by the entire PV system in a day**, and **metadata about the PV systems**. The specifics of these two data types are explained later in this section.

Data from 451 PV systems were initially provided. After validating the metadata, 95 systems were excluded due to incomplete or incorrect metadata. Additionally, 25 systems lacked sufficient measurements for either training or testing our algorithm and were therefore removed. Furthermore, 5 systems could not be tuned by the normalizer and were also excluded. As a result, our algorithm was tested on 326 PV systems, with all 326 systems being available as “neighbouring system”, to train each target model.

On average, we have 408 days of data per system, with the majority having around 470 days, as shown in Figure 17. However, this means that many systems have missing values, which need to be addressed, as we will discuss in Section C.1.1.1.

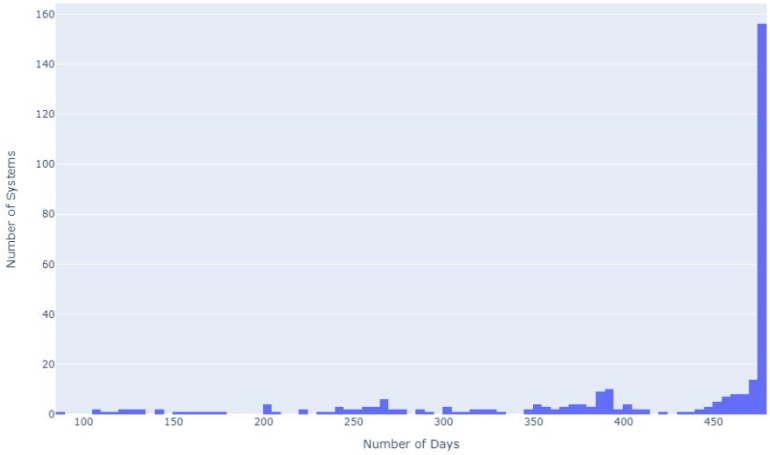



Figure 17 Histogram of number of measures per PV systems

These PV systems are primarily located in the canton of Bern, Switzerland (see Figure 18).

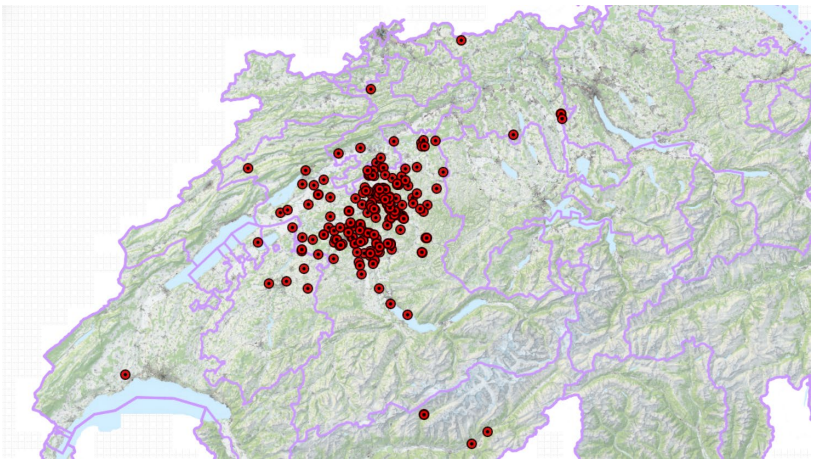


Figure 18 PV system distribution map

Our method relies solely on the AC energy production data of the PV systems and does not use any weather data or additional sensor information, which makes it broadly applicable since energy production data is commonly available for any PV system. Although additional data or sensors could improve anomaly detection, such as irradiance, we opted for a solution that avoids these to ensure the method is universally applicable and remains cost-effective for small and medium-sized installations that may not have the resources for extra equipment.

## 1.1 Energy produced per day (kWh)

The first data used is the energy produced by a system over the course of the day. It is crucial to understand the data acquisition process to get a better idea of the nature of this data, and to identify potential sources of measurement errors that could impact our results.

The partner company installs one incremental energy meter at the output of all inverters, as shown in Figure 19. This meter records the cumulative energy passing through it, which may come from any PV array, or a DC battery placed before the meter. We will see in Section D.8.3 of the results that this setup can affect the accuracy of our anomaly detection, particularly due to the presence of DC batteries.

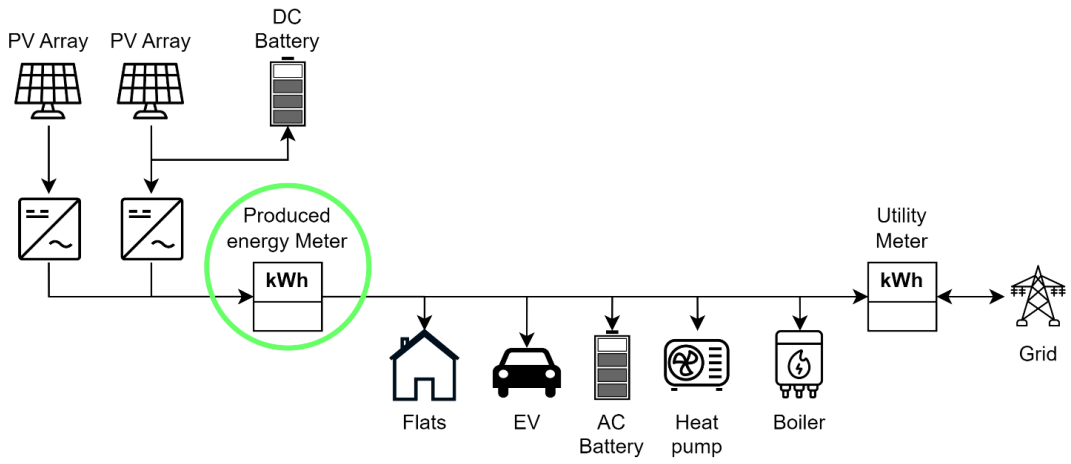



Figure 19 Diagram of the equipment in a PV system, with the location of the energy meter highlighted.

The meter takes a "snapshot" of its value at regular intervals, typically every minute, and sends this data to a server. The difference between two snapshots indicates the energy produced per minute, hour, day, month, etc. However, the data acquisition process sometimes encounters anomalies, such as when the connection between the meter and the server is lost, leading to erroneous measurements. These issues will be discussed further in Section D.8,"Impact of erroneous Input Data & Metadata".

However, the data used in this study is not the raw meter data but rather the total daily energy production. Therefore, we have one value per day for each PV system, rather than a value per minute. The reasons for this choice and its implications are discussed in Section E.1, “Limitations of daily data usage”.

An example of the input data used throughout this study is shown in Table 3. Each column represents a PV system, commonly referred to as a "feature" in machine learning, and each row represents an observation—the daily energy produced by each PV system on that day.

Table 3 Example of Input Data

| | 2026250 | 2026251 | 2026258 | 2026269 | 2026271 | a001017 | a001018 | a001020 | a001021 | a001022 | ... | a001633 | a001634 | a001637 | a001638 | a001661 | g001002 | g001003 | g001004 | g001005 | g001006 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Date | | | | | | | | | | | | | | | | | | | | | |
| 2023-03-20 | 47.1 | 76.5 | 123.3 | 51.70 | 20.5 | NaN | 43.7 | 46.4 | 25.1 | 32.40 | ... | NaN | NaN | NaN | NaN | NaN | 29.7 | 65.2 | 81.7 | 84.7 | 58.6 |
| 2023-03-21 | 53.4 | 77.7 | 121.2 | 60.20 | 24.1 | NaN | 40.5 | 54.4 | 25.4 | 40.50 | ... | NaN | NaN | NaN | NaN | NaN | 30.3 | 78.9 | 71.1 | 103.3 | 68.4 |
| 2023-03-22 | 70.5 | 97.0 | 158.9 | 75.10 | 29.8 | NaN | 57.4 | 68.5 | 32.6 | 52.30 | ... | NaN | NaN | NaN | NaN | NaN | 39.4 | 97.5 | 83.7 | 127.5 | 84.4 |
| 2023-03-23 | 25.8 | 47.2 | 55.5 | 21.80 | 9.2 | NaN | 19.6 | 24.2 | 12.1 | 18.30 | ... | NaN | NaN | NaN | NaN | NaN | 13.9 | 30.6 | 28.0 | 50.6 | 39.8 |
| 2023-03-24 | 9.7 | 19.1 | 24.5 | 9.60 | 4.6 | NaN | 6.8 | 13.3 | 5.4 | 8.60 | ... | NaN | NaN | NaN | NaN | NaN | 6.8 | 14.6 | 21.9 | 16.1 | 15.9 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2024-07-06 | 26.7 | 55.5 | 69.7 | 32.75 | 13.0 | 84.8 | 19.8 | 28.1 | 12.7 | 18.32 | ... | 35.9 | 50.95 | 38.6 | 29.6 | 43.74 | NaN | 38.1 | 42.9 | 49.2 | 44.0 |
| 2024-07-07 | 38.5 | 76.7 | 89.1 | 33.24 | 24.7 | 109.9 | 36.4 | 38.6 | 20.5 | 30.53 | ... | 60.9 | 60.53 | 52.6 | 41.3 | 47.46 | NaN | 47.4 | 61.8 | 101.0 | 62.2 |
| 2024-07-08 | 73.6 | 133.0 | 190.5 | 85.23 | 40.1 | 200.9 | 61.8 | 78.3 | 35.4 | 54.48 | ... | 102.0 | 124.97 | 105.7 | 73.5 | 113.71 | 23.0 | 102.4 | 105.9 | 188.6 | 116.3 |
| 2024-07-09 | 72.9 | 129.1 | 178.4 | 82.22 | 40.3 | 216.5 | 58.8 | 77.8 | 33.5 | 51.86 | ... | 100.2 | 121.30 | 101.8 | 71.5 | 103.27 | 42.3 | 100.2 | 102.5 | 186.1 | 110.5 |
| 2024-07-10 | 40.2 | 68.6 | 100.6 | 35.39 | 20.6 | 112.1 | 35.3 | 45.1 | 22.4 | 29.05 | ... | 62.1 | 59.38 | 57.5 | 39.7 | 56.43 | 21.7 | 50.5 | NaN | 106.3 | 61.8 |

479 rows × 356 columns

### 1.1.1 Missing Values

The input data can contain missing values, represented as "NaN" in Table 3. Figure 20 visualizes the distribution of missing values across all PV systems. Descriptive statistics analysis shows that only 11% of the PV systems have no missing values, while 14% of systems have more than half of their observations missing. This presents a unique challenge compared to the two studies similar to ours (Feng et al., 2020; Iyengar et al., 2018), which were based on laboratory measurements without missing values. Missing values pose significant problems, as most machine learning tools require complete datasets. Typically, missing data must be addressed by either filling in missing values with arbitrary numbers (*Imputation of Missing Values*, 2024) or by excluding systems and days with missing data.

However, in our case, it is not acceptable to exclude data, as our algorithm needs to work consistently, even if a neighbouring system fails to provide data on a given day due to a meter or network issue. Additionally, data imputation is also not feasible due to the substantial number of missing observations, which would drastically reduce the algorithm's accuracy. Therefore, we need to design an algorithm that is robust to missing values.

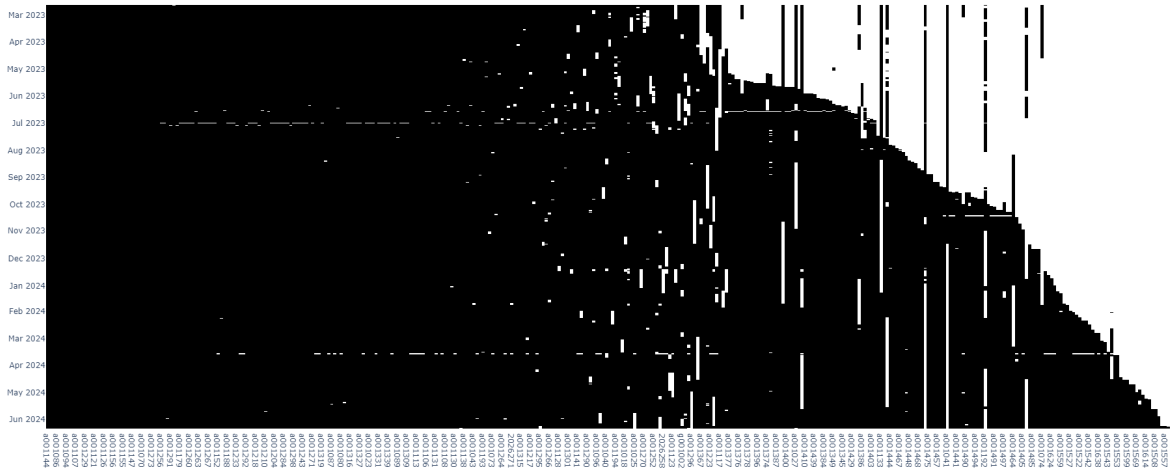



Figure 20 Missing Values Distribution. Each column represents the measures of a PV system over time. Black tiles indicate days with measured values, while white tiles indicate days with missing values.

## 1.2 Metadata

Metadata for each PV system is also provided, comprising two parts: general system information and details for each array. There can be 1 to 4 arrays per system, each with different modules, sizes, azimuths, and tilts, generally corresponding to modules mounted on different roof faces. An example of the metadata is available in Figure 21.

**General information includes :**

- Total peak power [kWp]
- Longitude & latitude in degrees (WGS84 format)
- Mounting type (on top of the roof or integrated)

**For each array, the following details are provided :**

- Array Azimuth & Tilt
- PV Module manufacturer and type
- PV Module peak power [Wp]
- Number of PV Modules

### 1.2.1 Metadata Pre-processing

Some pre-processing was performed on the metadata provided by our partner before running our algorithm. The altitude of each PV system was added using the GeoAdmin API provided by the Swiss Confederation. This query only needs to be executed once to add this information to our local database. Additionally, the structure of the metadata provided by the partner was slightly modified to facilitate data access. An example of this pre-processing is available in Figure 21.

Original:

```
"a00xxxx": {
  "metadata": {
    "pv_kwp": 8.84,
    "pv_mod2_wp": 260,
    "pv_mod1_number": 18,
    "pv_mod2_azimut": 60,
    "pv_mod1_azimut": 150,
    "pv_mod2_manufacturer": "Aleo",
    "loc_latitude": 47.123,
    "pv_pv_type": "Angebaut",
    "pv_mod1_manufacturer": "Aleo",
    "loc_longitude": 7.123,
    "pv_mod2_tilt": 26,
    "pv_mod2_type": "X81L260 HE",
    "pv_mod2_number": 16,
    "pv_mod1_tilt": 26,
    "pv_mod1_type": "X81L260 HE",
    "pv_mod1_wp": 260
  }
}
```

➔

Modified:

```
"a00xxxx": {
  "metadata": {
    "pv_kwp": 8.84,
    "loc_latitude": 47.123,
    "pv_pv_type": "Angebaut",
    "loc_longitude": 7.123,
    "loc_altitude": 562.3
  },
  "arrays": {
    "2": {
      "pv_wp": 260,
      "pv_azimut": 60,
      "pv_manufacturer": "Aleo",
      "pv_tilt": 26,
      "pv_type": "X81L260 HE",
      "pv_number": 16
    },
    "1": {
      "pv_number": 18,
      "pv_azimut": 150,
      "pv_manufacturer": "Aleo",
      "pv_tilt": 26,
      "pv_type": "X81L260 HE",
      "pv_wp": 260
    }
  }
}
```

Figure 21 Metadata Example and Pre-processing

# 2 Software Requirements

The algorithm was developed in Python using a Jupyter Notebook environment. Python was chosen due to its efficiency in prototyping, effective matrix computation management, and access to popular libraries for data analysis and machine learning.

The libraries used include:

- **Numpy & Pandas**: For data management and matrix calculations.
- **PVLib**: For developing physical models of PV systems.
- **scikit-learn**: For training and deploying machine learning models.
- **Plotly**: For visualizing graphs.
- **Dash**: For creating interactive applications.

# 3 Our Strategy

In this section, we will outline the reasons behind selecting our Anomaly Detection and Classification (ADC) strategy. This final strategy will be a hybrid approach that involves a physics-based simulator for normalizing input data, a regressor using data from neighbouring systems to estimate the expected production of the target PV system, and a rule-based anomaly detector.

## 3.1 Presentation of the problem

As a reminder, the primary objective is to detect anomalies and provide valuable monitoring information, such as the expected production or anomaly classification. To achieve this detection, we need to detect underproduction in the system, indicating a potential anomaly. However, the impact of an "anomalous" factor on production is only one component among several other factors influencing production. As discussed in Section B.2, “Factors Affecting PV Production”, the energy output is affected not only by "Anomalous Factors" but also by "System-Specific Factors" and "Regional Factors". These significantly affect production measurements. **Therefore, to detect an anomaly, we must first separate the impact of each type of factor on production.**

Figure 22 illustrates the different components affecting PV production through a simulation of a PV system's output. The breakdown is as follows:

- **Initial Solar Irradiation**: The green area represents the total solar power available at the PV system's geographic location at a given time of year. This is our starting point.
- **System-Specific Factors**: By considering system-specific factors, such as tilt, azimuth, or shading from nearby buildings, we arrive at the blue area. This represents the maximum possible production of the system. In our example, the system is oriented west, with shading present from 14:00 to 15:00.
- **Regional Factors**: Taking into account regional factors such as cloud cover and temperature, we get the yellow area, which represents the expected production when the system is functioning properly. In our example, we can see that the impact of these factors is relatively stochastic.
- **Anomalous Factors**: Finally, adding the effect of anomalies results in the red area, which represents the actual measured output, our input data. In the example, there is a 10% reduction in production in the morning, increasing to a 50% reduction at 15:30.

The goal of our algorithm is to generate the red line, which indicates the reduction in production due to anomalies, starting from the red area, which is the measured data.

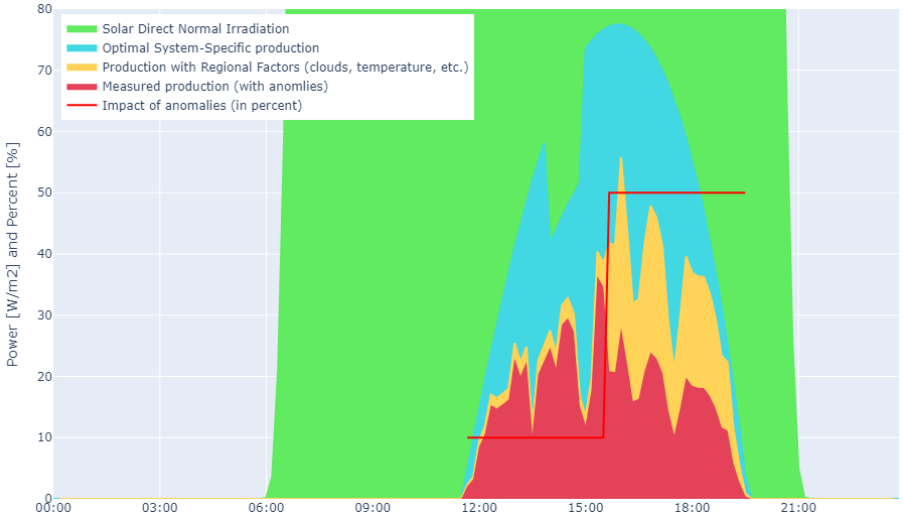



Figure 22 Simulated example of production, broken down into optimal production, production with regional factors, and real measured production (with anomalies). The red line shows the percentage of impact of the anomalies on the production.

## 3.2 Comparison of strategies

Next, we will review the detection strategies presented in the literature review in Section B.3.2, “Detection Strategies”, to select the one that best meets our needs and data constraints.

**Anomaly Signature Analysis**
This strategy involves directly analysing the behaviour of the measured power curve (the red area in Figure 22) to identify known anomaly signatures, such as a sudden drop in production due to a short circuit. However, this approach is not feasible for our case for several reasons. First, it would require studying the impact of each anomaly on production to identify its signature (1), which varies from system to system. Secondly, weather variability introduces too much noise into the signal (2), making it difficult to detect a specific anomaly signature. Third, the signature of anomalies that occur at night cannot be detected since there is no production data (3). Finally, anomalies that develop gradually over time would also be undetected (4).

**Comparing Historical Data to Current Data**
The naive approach of comparing current production to historical data, whether from the same system or a database like PV GIS, only works when comparing over long time spans, such as comparing this month's production with the same month last year. This method statistically averages out weather impacts. However, it has two major drawbacks. First, it provides low accuracy because weather conditions can differ significantly from one year to the next (1). On average, monthly production can deviate by 28% compared to the average of previous years (European Commission, 2024). Second, it limits anomaly detection only to a monthly frequency (2).

**Comparing Simulated Data to Current Data**
Using a simulator to model a PV system's behaviour, whether based on a physical model or machine learning, requires input from weather data to generate meaningful results for anomaly detection, what we don't have.

**Comparing Data from Similar PV Systems**
Using data from neighbouring PV systems is a promising approach that seems most suitable for our situation. Nearby systems are similarly affected by regional factors like cloud cover, meaning these regional effects can be partially inferred from the neighbouring systems using the Half Sibling Regressor. These systems effectively serve as indirect weather stations, capturing radiation and temperature through their energy production.

## 3.3 Final Strategy

**The primary strategy chosen is to use neighbouring PV systems as indicators of regional factors.** By applying the Half Sibling Regressor principle, we can estimate the expected production of the target system, represented by the yellow area in Figure 22.

However, we must also consider system-specific factors. Regional factors affect all neighbouring systems in the same way, but each system has its own unique baseline production. For example, a cloud may reduce production by 40% for all nearby systems at a particular time of day, but each system's maximum possible output (shown in blue in Figure 22) at that moment is different. Since the characteristics and maximum production profiles of each system vary, these system-specific factors must be accounted for if we want to compare them accurately.

To do this, **we will normalize the measured output** (shown in red on Figure 22) by the maximum possible production (in blue). This normalization transforms the measurement into a percentage of the maximum possible output, effectively removing system-specific factors from the data. This enables a more accurate comparison between neighbouring systems. This approach falls under the topic of data normalization, as explained in more detail in the literature review in Section B.4.1, “Input Data Normalization”. We achieve this normalization using a **physics-based simulator that calculates the maximum possible power production of a system** at any given time. This simulator allows us to scale each system's output between 0 (no production) and 1 (maximum production), facilitating comparisons across multiple systems.

With both the Normalizer and the Half Sibling Regressor, we can estimate the expected production of a system, as shown in yellow in Figure 22. However, we still need to detect and classify anomalies. To do this, we first require a **Comparator** that can compare the expected production with the actual measurements to calculate any underperformance of the system. Finally, we will use a **rule-based white-box classifier** to analyse the underperformance signature of the system and classify defined anomalies based on specific rules.

# 4 Algorithm Design

After selecting our strategies, the process can be summarised as follows: first, we normalise all measurements using a physics-based simulator; then, the normalised measurements from neighbouring systems are used to estimate the target system's production using a Half Sibling Regressor. The difference between the expected and measured production of the target system allows us to determine any underperformance, which can then be analysed using rules to detect anomalies.

The detailed steps of our algorithm are outlined below and illustrated in

Figure 23. The implementation of these steps is explained in the following sections:

1. **Normalization**: The measured data from the target and neighbouring PV systems are initially normalised using a physics-based model specific to each system. This ensures that all measurements are scaled between 0 and 1, representing the percentage of maximum possible production at any given time. This normalization makes systems with different system-specific factors comparable.
2. **Filtering**: The data and metadata undergoes filtering to remove any measurements and systems that deviate significantly from the norm or are incompatible.
3. **Half Sibling Regression**: The normalised measurements from neighbouring systems are then used in a regression model to estimate the expected production of the target system.
4. **Comparison**: The estimated energy production is compared to the actual measured energy to determine the under-production.
5. **Residual Seasonality Removal**: Certain seasonal factors, which reappear at regular intervals and are not taken into account by the normalizer, are removed at this step.
6. **Detection/Classification**: The under-production for each observation is analysed by a rule-based classifier to detect and classify anomalies.

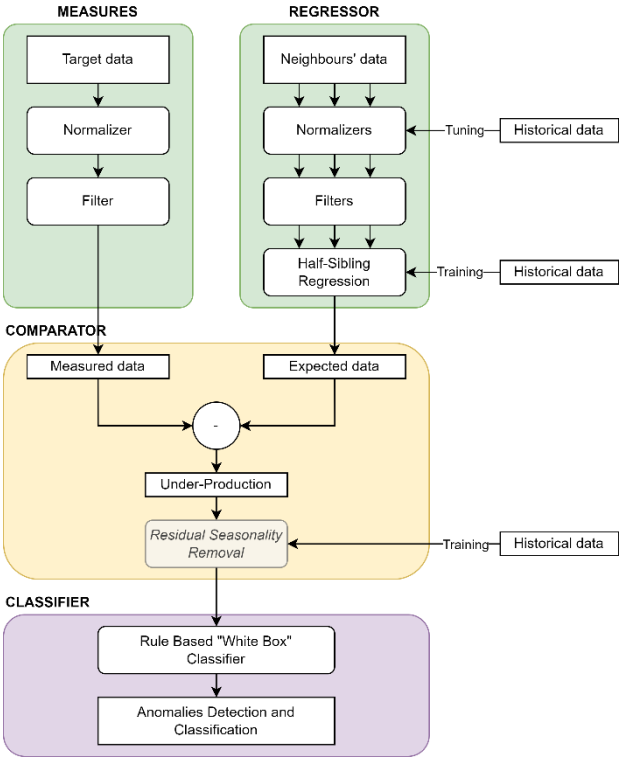



Figure 23 Illustration of our Algorythm design

# 5 Normalization

## 5.1 Goal

The goal of the normalization process is to eliminate the system-specific factors for each PV system, such as tilt, azimuth, location, and losses. After normalization, the measurements range between 0 and 1, representing the percentage of current production relative to the production under ideal weather conditions (clear sky and a temperature of 20°C). This normalized value reflects the impact of weather on production: a value close to 1 indicates ideal weather, while a value near 0.5 suggests that weather factors, such as cloud cover, reduce production by 50%. An example of the result after normalization can be seen in Figure 24. The purposes of normalization include:

- Ensuring that the values for each system are on a comparable scale, which benefits the regressor.
- Accounting for system-specific factors. For example, the system in Figure 24, with a 30° tilt, shows significant seasonal variation in production, while a system mounted vertically would have more stable output. These differences are removed after normalization, allowing systems to be directly compared.
- Reducing the number of neighbouring systems required for the subsequent regression, as systems with different characteristics can now be compared.
- Preventing overestimation of errors in the summer. Since absolute values and variations are larger in summer due to higher production, a regressor might focus more on these summer values during training, giving them more weight than winter values.

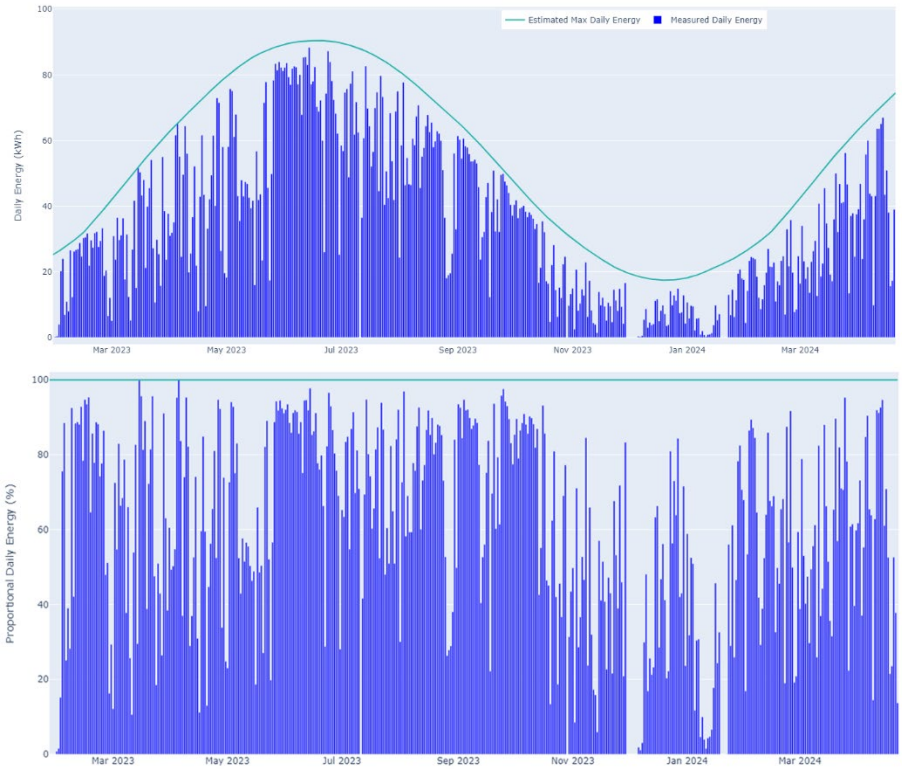



Figure 24 Normalization of data

## 5.2 Normalizer selection

Before implementing an effective normalizer, other naive techniques were tested. These included dividing the measurements by the peak power (the total installed PV module capacity) (1), dividing by

the average annual production (2), and using a moving average of the last month's production (3). We will not revisit the pros and cons of these methods in detail, but the division by peak power is basic and biased, as two systems with the same peak power can have completely different outputs depending on their configuration. The moving average method is better, as it accounts for seasonal factors in normalization. Ultimately, a physics-based simulator was chosen for implementation, as discussed in the literature review, Section B.3.4.2, “Physics-Based Black Box Models”, to more accurately normalize all the characteristics of a PV system.

## 5.3 Implementation

The physic-based model incorporates a PV simulator proposed by PV Lib (Andrews et al., 2014), implementing physical formulas to calculate power production. It’s possible to simulate the impact of various factors, such as geographic location, day of the year, time of the day, tilt, azimuth, module type, mechanical mounting, system topology (number of array and modules), losses, illumination, temperature, cloud cover, and wind. In our case, we implemented a physics-based "grey" box model. This approach utilizes known metadata provided by our partner as input for the simulator, such as system configuration and location, while estimating missing information, like system losses, by tuning the simulator using historical production data. Since we lack meteorological data, which is inferred in the next step using neighbouring systems, the simulation assumes optimal weather conditions to estimate the maximum possible production (no cloud, and 20°C).

Figure 25 and the following points illustrate and explain in detail how the simulator works.

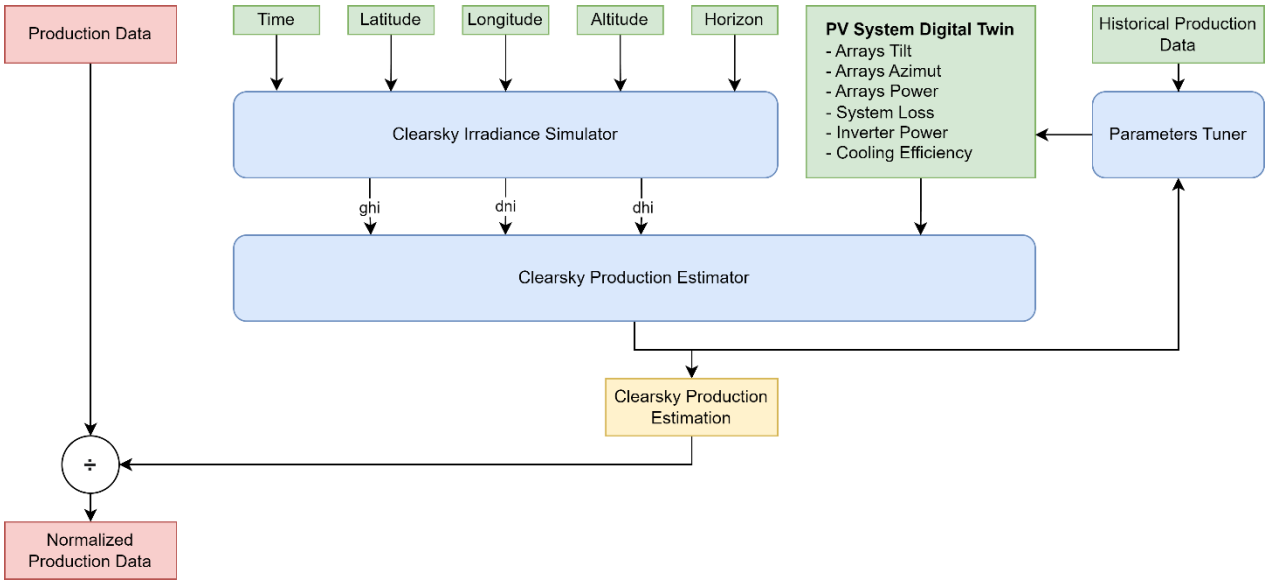



Figure 25 Physic-Based Normalizer Diagram

### 5.3.1 Clearsky Irradiance Simulator

To begin, we calculate the direct and diffuse irradiance reaching the PV system at a given moment. This can be computed using physical and geometric principles outlined in the model by Ineichen & Perez (Ineichen & Perez, 2002; Perez et al., 2002). The date, time, and location provide the sun's position, while factors such as refraction, air pressure, turbidity, and topology are used to calculate the Direct Normal Irradiance (DNI), Diffuse Horizontal Irradiance (DHI), and Global Horizontal Irradiance (GHI).

### 5.3.2 PV System Digital Twin

Next, a Digital Twin of the PV system is designed, replicating the characteristics of the actual system. The model can be improved by specifying details for the system such as:

- Total system power (kWp)
- Solar tracking system (none, single-axis, dual-axis)
- Albedo
- Inverter efficiency characteristics
- Inverter power
- Losses (wiring, connections, light-induced degradation, aging, soiling, local shading, mismatch nameplate)

And then for each array:

- Tilt
- Azimuth
- Rack type (integrated, open, back-insulated)
- Module characteristics (power, temperature coefficient)
- Module construction (glass/glass, glass/polymer)
- Number of modules per string
- Number of strings

Many of these parameters are directly provided in the metadata from our partner, allowing us to automatically create a high-quality Digital Twin for each PV system. For unknown values, defaults are used (e.g., albedo, inverter efficiency, module characteristics and construction type), while losses are determined during tuning (see next Section 0). Although it might be tempting to improve the digital twin, this is not important, as it would have minimal impact on the anomaly detection results, as shown in Section D.5.1, “Impact of the normalization”.

### 5.3.3 Computation

The simulator will compute the power of the PV system at a given time. To do it, it will execute the following steps :

1. Calculate the **direct and diffuse radiation**, using the model by Ineichen & Perez (2002; 2002).
2. Compute the **Incidence Angle Modifier** (IAM), accounting for efficiency losses when the irradiance angle is not perpendicular to the modules, using the model by De Soto et al. (2006).
3. Calculate the **DC power output** with the NREL’s PVWatts DC power model (Dobos, 2014).
4. Calculate the **AC power output** taking into account inverter efficiency with the NREL’s PVWatts inverter model (Dobos, 2014).
5. Calculate **system losses** using the NREL’s PVWatts system loss model (Dobos, 2014).

### 5.3.4 Maximum daily energy aggregation

Since the goal is to normalize daily energy measurements, we need to calculate the maximum energy produced per day. The simulator provides the power output at a specific time, so we compute the power at regular intervals throughout the day and integrate these values to obtain the total daily energy. Higher frequency of simulations increases the accuracy of the daily energy calculation, but also require more computing power. Tests showed that simulating power at intervals of 1 hour resulted in a decrease of simulation accuracy of only ±0.43% compared to 10-minute intervals, while being six times faster. Therefore, power is computed hourly, and the results are integrated to determine the daily energy output.

## 5.4 Tuning

The Digital-Twin model of the PV system was fine-tuned to align with the measured values. This tuning process is inspired by the "Physics-based Black-Box" models discussed in the literature review, which learn the parameters of a PV system based on its historical production data.

The tuning specifically aims to learn the system's static losses, focusing solely on this parameter for the following reasons:

- **Unknown Variable**: Losses are an unknown factor, and the default value of 14% is inaccurate.
- **Static Parameter**: Losses are relatively easy to tune because they remain constant throughout the year.
- **Limited Data Resolution**: Other parameters, such as tilt and azimuth, are difficult to tune without hourly production measurements to assess the system's performance throughout the day.
- **Include daily seasonality**: With daily measurements, system-specific seasonal factors like local shading are treated as a static loss in production.

However, it would be possible to tune more parameters, especially by using a higher frequency of measurements, as we will discuss in Section E.1, “Limitations of daily data usage”.

### 5.4.1 Tuning losses

The tuning process for the loss parameter is illustrated in Figure 26. The objective is to adjust the initial magenta curve to better fit the data, resulting in the green curve.

A naive approach to finding the loss would be to set a loss factor so that all measurements are equal to or lower than the maximum energy curve since theoretically, measurements cannot exceed the maximum possible production. However, this approach would fit the curve to the outliers, visible in yellow in the Figure. These outliers are erroneous measurements, and not days with maximal production.

Therefore, we need to detect days with clearsky during the entire day, representing the real maximal possible production, to fit the simulator to the measures. To detect these days, the following steps were taken:

1. **Identifying Productive Days**: We selected the most productive day of each week (shown in red in Figure 26), which should closely correspond to a clear-sky day, thus aligning closely with the estimated maximum production of the simulator.
2. **Calculating Losses**: We then calculated the losses (or relative differences) between the estimated maximum production and the measurements of these productive days:
$$Losses = \frac{measures_{max} - estimate_{max}}{estimate_{max}}$$
3. **Outlier Detection**: Losses with a Z-Score greater than 1 were considered outliers (highlighted in yellow in Figure 26), indicating they are more than one standard deviation from the mean.
$$z_{score} = \frac{|Losses - mean|}{std}$$
4. **Selecting Optimal Loss**: Finally, the remaining productive day with the minimum loss is selected to represent the static loss of the PV system.

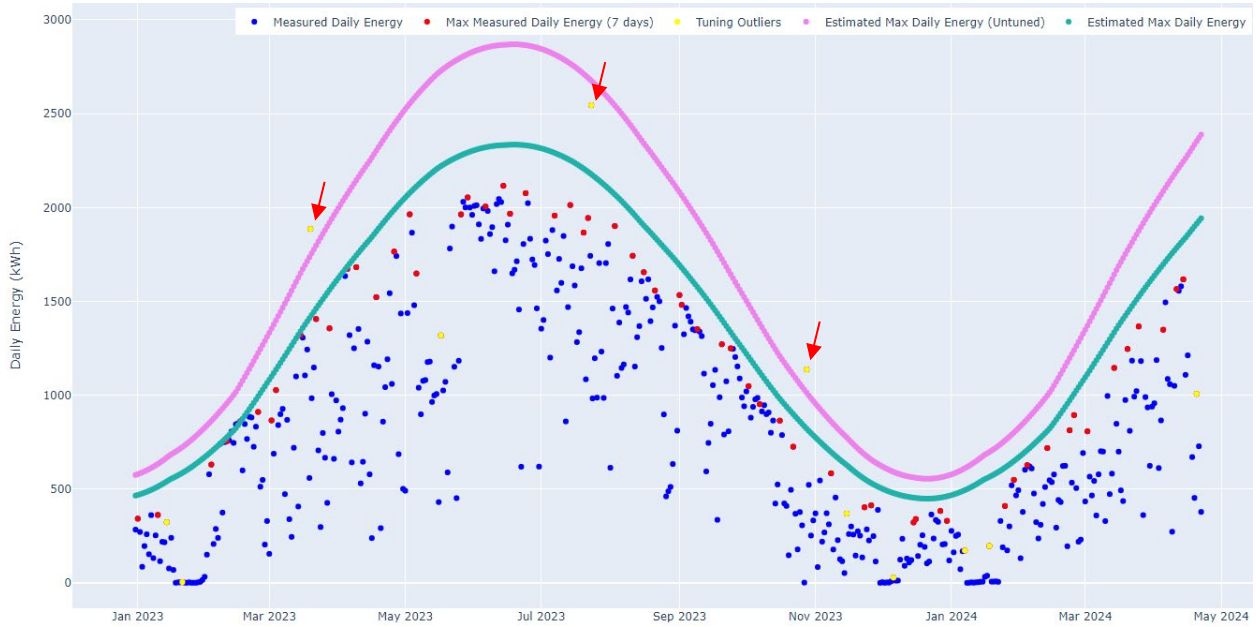



Figure 26 Illustration of the Normalizer Tuning

# 6 Filtering

The filtering process aims to remove potentially abnormal measurements before training the Half Siblings Regressor in the next step. Since the regressor's goal is to estimate the normal production of a target PV system, any anomalies in the training data of either the target or neighbouring systems will reduce the accuracy of this estimation. For instance, the outliers indicated with a red arrow in Figure 26 are examples of data points that should be removed from the training process.

Given that our data is not labelled with specific information on when anomalies occurred, a systematic filtering of all anomalies in the training data is not possible. However, we implemented two rules to help remove such anomalies:

1. **Exceeding Theoretical Maximum Production**: Any daily measurements exceeding 110% of the theoretically maximum possible daily energy production, computed by the Normalizer, were considered outliers since such values are not physically possible.
2. **Near-Zero Production**: Measurements close to or below 0 kWh produced in a day were also considered outliers. Analysis showed that these values typically result from measurement errors.

Thus, only measurements falling within the range of 1% to 110% of the maximum possible production are retained for training the regressor.

# 7 Half-Sibling Regression

The Half-Sibling Regression is the strategy discussed in the literature review in Section B.3.2.6, "Comparing data of similar PV systems/subsystems". This approach leverages neighbouring PV systems to understand the regional factors affecting the production of the target PV system. The theory and formulas for this technique are detailed in the literature review Section B.3.2.6. To summarise, we need to train a function to estimate the regional factors for our target system, $f(R)$, using $X_i$, the normalised production data of neighbouring systems:

$$E[f(R)|X_i]$$

To illustrate the regression process more concretely, let's consider Table 4 below, which shows the input data used for the training of the regressor, and during prediction. The first column, highlighted in red, represents our target system for which the regressor is being trained. The other columns contain the data from neighbouring systems, which are used to estimate the target system's production and are referred to as features in Machine Learning. Each row represents an observation. During training, a Machine Learning model is trained to estimate the normalised production of the target system with minimal error, using the normalised productions of neighbouring systems as features. In the production phase, the values of the neighbouring systems are provided daily to the trained model, which then estimates the normalised production of the target system, circled in red in the Table 4. The results and accuracy of our energy production estimation are detailed in Section D.2.

It is important to note that the time information is not used as a feature in the regression model, meaning the model does not consider when a particular measurement was taken or the temporal relationship between two measurements. This approach is taken for three reasons:

1. **Seasonal Factors**: Seasonal variations are primarily accounted for in the normalisation process and further addressed in the "Remaining Seasonality Removal" step. Thus, the Half-Sibling Regressor is not designed to consider temporal aspects in the regression.
2. **Data Requirements**: To make effective use of temporal information, at least one full year of data would be needed for all systems to identify any correlated patterns from year to year.
3. **Model Complexity**: Including temporal data would add unnecessary complexity to the model and its interpretation, which is not desired at this stage of solution development.

However, it could be interesting to observe the impact on our regressor of providing it with temporary information as a feature, and we discuss this in the Section E.9 of the " Limitations and Future Directions".

Table 4 Normalized input data of the ML model during training and production. The target system is in red, the others columns are the neighbouring systems. Each row is an observation. The circled highlight the value to predict.

| | a001036 | a001096 | a001099 | a001131 | a001138 | a001155 | a001159 | a001163 | a001190 | a001195 | a001204 | a001205 | a001209 | a001214 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Date | | | | | | | | | | | | | | |
| | | | | | | | Training | | | | | | | |
| 2022-12-31 | 0.61 | 0.57 | 0.55 | 0.43 | 0.46 | 0.43 | 0.29 | 0.73 | NaN | 0.59 | 0.61 | 0.63 | 0.27 | NaN |
| 2023-01-01 | 0.73 | 0.61 | 0.63 | 0.68 | 0.61 | 0.75 | 0.36 | 0.68 | NaN | 0.45 | 0.68 | 0.79 | 0.55 | NaN |
| 2023-01-02 | 0.57 | 0.50 | 0.40 | 0.39 | 0.42 | 0.45 | 0.43 | 0.64 | NaN | 0.94 | 0.50 | 0.50 | 0.37 | NaN |
| 2023-01-03 | 0.18 | 0.15 | 0.10 | 0.06 | 0.07 | 0.08 | 0.13 | 0.13 | NaN | 0.13 | 0.14 | 0.12 | 0.04 | NaN |
| 2023-01-04 | 0.40 | 0.39 | 0.25 | 0.18 | 0.18 | 0.22 | 0.23 | 0.49 | NaN | 0.34 | 0.49 | 0.37 | 0.18 | NaN |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2024-04-18 | 0.35 | 0.29 | NaN | 0.23 | 0.18 | 0.30 | 0.27 | 0.29 | 0.22 | 0.24 | 0.17 | 0.46 | 0.47 | 0.23 |
| 2024-04-19 | 0.23 | 0.23 | NaN | 0.18 | 0.18 | 0.20 | 0.18 | 0.20 | 0.17 | 0.25 | 0.22 | 0.22 | 0.19 | 0.22 |
| 2024-04-20 | 0.52 | 0.51 | NaN | 0.42 | 0.43 | 0.45 | 0.33 | 0.51 | 0.43 | 0.48 | 0.42 | 0.50 | 0.40 | 0.39 |
| 2024-04-21 | 0.37 | 0.37 | NaN | 0.30 | 0.31 | 0.35 | 0.29 | 0.37 | 0.34 | 0.39 | 0.20 | 0.40 | 0.23 | 0.36 |
| | | | | | | | Production | | | | | | | |
| 2024-04-22 | ??? | 0.15 | NaN | NaN | 0.06 | 0.14 | 0.31 | 0.14 | 0.06 | 0.12 | 0.13 | 0.18 | 0.14 | 0.18 |

## 7.1 Selection of Neighbouring Systems

Throughout this work, the term "neighbouring systems" is frequently used because systems that can be useful for estimating the production of a target PV system are typically geographically close. However, the concept of distance is not directly used to select the systems provided as feature to the Half Sibling Regressor. Instead, the Half Sibling Regressor itself selects the most relevant systems for its prediction. More information on this selection process is presented in Section D.12, “Selected Neighbouring PV System”.

To accelerate the training of the model, however, some candidate systems were pre-selected. A Pearson correlation was computed between the normalised measurements of each system, and those with a correlation higher than 90% were retained as features for prediction, making them potential candidates as neighbouring systems. A figure showing the result of this correlation analysis can be found in Appendix H.5.

## 7.2 Selection of the ML Model

For implementing the ML regression model that will perform the Half Sibling Regression, we first developed a custom model. This model is more intuitive than the well-known ML models and natively handles missing values. The principle of this model is explained in the following section.

We then tested various standard ML models discussed in the literature review in Section B.3.4. The models tested included Linear Regression, Support Vector Machines, Random Forest, and k-nearest neighbours. However, no Artificial Neural Network was tested. The methodology and operation of these standard ML models will not be explained further, as these details are readily available in the literature and are beyond the scope of this thesis. Only the custom Averaged Pairwise Linear Regression will be explained.

### 7.2.1 Averaged Pairwise Linear Regression

The idea behind this model follows the intuitive reasoning one might naturally use to estimate the production of a PV system using data from its neighbours. We assume that the normalised production of the target system follows a linear relationship with the normalised production of a neighbouring system. Simply put, if a neighbouring system produces at 20% of its maximum capacity, the target system should also produce at a similar level.

During training, we learn the linear regression between the target system and each neighbouring system, one pair at a time. We end up with $n$ linear regression, $n$ behing the number of neighbouring PV systems. Systems with a regression $R^2$ greater than 0.9 are retained, indicating that they are good candidates for estimating the target system's production. Figure 27 illustrates this concept.

Then, when estimating the target system's production, we compute the average of the predictions from each retained neighbouring system:

$$y \approx \frac{1}{N}\sum_{i=1}^{n} E[y|X_i]$$

where $n$ is the number of retained good neighbouring systems.

This machine learning technique is called ‘Ensemble Learning’, and consists of training several models in parallel that predict the same value, in order to improve the variance and resilience of the final model (IBM, 2024). Bootstrapping, discussed in the literature review in Section B.4.4, is part of Ensemble Learning. This technique has the advantage of functioning even if some neighbouring systems have missing data. In such cases, the prediction for those systems is not made, and they are simply excluded from the average calculation.

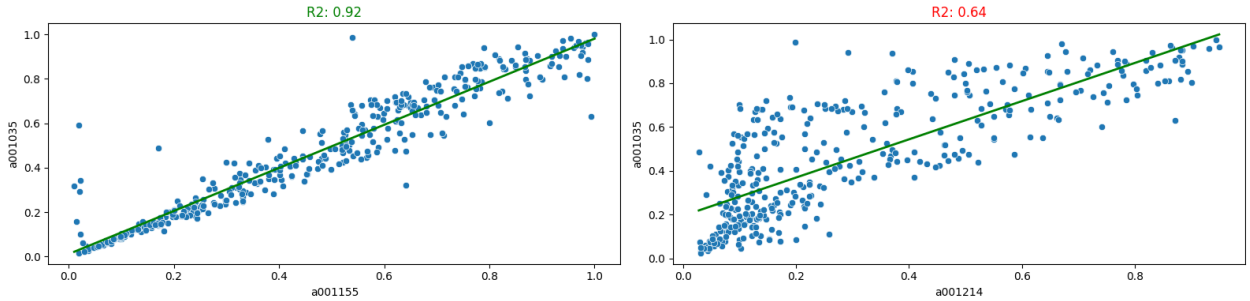



Figure 27 Pairwise Linear Regression of the target system 'a001035' with a good candidate (left) and a poor candidate (right)

### 7.2.2 Final Model Choice

After testing all the models, with results presented in Section D.6.1, we found that the Random Forest model provides the best performance. Several studies, including those cited in Figure 7 (Iyengar et al., 2018; Rapaport & Green, 2021), also suggested that Random Forest is the most effective model in our context. Additionally, the Random Forest Regressor natively supports missing values (Ding & Simonoff, 2010), whereas in other models, missing values must be removed before training and prediction, which is not feasible in our case (see Section C.1.1.1).

**Therefore, we have chosen to use a Random Forest Regressor for the Half Sibling Regression.**

## 7.3 Hyper-parameters Tuning

Every machine learning model requires hyper-parameter tuning to adjust the model's structure. The optimal hyper-parameters that yield the best results for our specific problem must be determined before the training of the regressor for each PV system. This process is typically empirical, involving testing various combinations of hyper-parameters, rather than analytically searching the minimum of a standard function (Koehrsen, 2018).

Hyper-parameter tuning was performed for the Random Forest Regressor, as well as each ML model seen previously, but we will focus on the tuning of the former, as it is the model we have selected. The results of this tuning are presented in Section D.6.2.

### 7.3.1 List of Hyper-parameters

The key hyper-parameters of a Random Forest Regressor, which is an ensemble model composed of multiple Decision Tree Regressors, are as follows:

- **Number of Trees**: The number of decision trees in the forest.
- **Max Features**: The maximum number of features considered for splitting at each node of a tree.
- **Max Depth**: The maximum number of levels in each decision tree.
- **Min Samples Split**: The minimum number of observations required in a node before it is split into two branches.
- **Min Samples Leaf**: The minimum number of data points allowed in a leaf node.

### 7.3.2 Grid Search

To find the optimal hyper-parameters, a grid search was conducted. This approach involves specifying a list of potential values for each hyper-parameter and training a new model for every possible combination of these values to identify the combination that results in the lowest error. This combination of hyper-parameter is then tested on every PV system. Although this method is computationally intensive due to the large number of models trained, it is feasible in our case because our model trains quickly (approximately 240 milliseconds), allowing us to test a wide range of combinations.

### 7.3.3 Cross-Validation

Five-fold cross-validation was used during the hyper-parameter tuning process, ensuring that each batch contained at least two weeks of continuous data to minimise the impact of temporal correlation between closely related observations. More details on these concepts can be found in the literature review, Section B.4.5.

### 7.3.4 Timing

The time required to test a single combination of hyper-parameters was approximately 6 minutes and 30 seconds. This duration results from the time it takes to train a model (~240 milliseconds), multiplied by the 326 systems being tested and the use of 5-fold cross-validation:

$$Tuning\ Time\ per\ Combination = t * n * k = 240ms * 326 * 5 = 391s = 6.5min$$

In total, 35 different combinations of hyper-parameters were tested, resulting in a total tuning time of approximately 3 hours and 50 minutes.

# 8 Comparator

The comparator calculates the under-production of a PV system by determining the difference between the estimated normalized production (based on neighbouring systems) and the actual normalized measurement. A positive value indicates underperformance compared to the neighbouring systems, while a negative value suggests that the system has outperformed its neighbours. The loss due to anomalies can be calculated as:

$$Under\ production = Expected_{norm} - Measure_{norm}$$

# 9 Remaining Seasonality Removal

In our methodology, the system-specific factors are generally taken into account by the physics-based normalizer. However, some factors that were not simulated by the physical model may still affect the production. These are typically recurring factors that impact production at regular intervals, such as shading from a nearby building that consistently reduces output at a certain time of day, depending on the time of year. This is illustrated in Figure 22, where illumination is reduced between 14:00 and 15:00 due to local shading.

If these local factors cannot be directly reflected in the initial process by the physics-based normalizer, they remain and are interpreted as under-production by the comparator. This would result in our method erroneously indicating that the PV system is underperforming at the same time every day.

To address this, we need to decompose the time series of losses into its four main components (Brownlee, 2017):

- **Level**: The average loss value.
- **Trend**: The long-term increase in losses, indicating the gradual degradation of the PV system.
- **Seasonality**: The repeating short-term cycles in loss values, representing remaining seasonal factors that need to be removed.
- **Noise**: The random variations in loss values, which are the unpredictable anomalies we aim to detect.

To implement this decomposition, we could use simple statistical tools like moving averages (Josef Perktold et al., 2024) or advanced ML models like the "Facebook Prophet" (Facebook's Core Data Science team, 2023).

However, the observed remaining seasonal factors typically have a daily seasonality, such as shading from a neighbouring building. Given that our data is collected on a daily basis, this step is not necessary. The daily underperformance due to these factors is already included in the system's daily losses, as determined during the tuning of the normalizer (Section 0). Therefore, we will not perform further

decomposition of the loss components in this work, as it would only be relevant for data collected at higher frequencies.

# 10 Detection

Once the daily losses are calculated, we can identify abnormal days by statistically determining when the losses fall outside the confidence interval of the Half-Sibling Regressor's prediction. For each day, the standard deviation of the Half-Sibling Regressor is computed, which is the standard deviation of the predictions made by each tree in the Random Forest. This standard deviation is then used to calculate the z-score for each measurement. The z-score indicates how many standard deviations a measurement is from the predicted mean:

$$z_{score} = \frac{y - mean(\hat{Y})}{std(\hat{Y})}$$

where $y$ is the measured value, and $\hat{Y}$ represents the predicted values from each tree in the Random Forest Regressor.

A measurement is considered anomalous if its z-score exceeds 1.65, meaning it is more than 1.65 standard deviations away from the value estimated by the Random Forest Regressor. This threshold provides a 90% confidence level that the observation is an anomaly (Wager et al., 2014).

Unfortunately, we cannot directly assess the accuracy of our anomaly detection in identifying true anomalies, as the input data lacks labelled anomalies. To address this, we simulated anomalies to create labelled observations, allowing us to evaluate whether the detector can accurately identify the simulated anomalies. The results of this evaluation are presented in Section D.2.1.

## 10.1 Simulating Anomalies

To obtain observations with labelled anomalies, we simulated anomalies within the test dataset. Anomalies were randomly applied to 5% of the test data, each involving a simple 30% reduction in the energy produced on the affected day. More complex anomaly simulations were not conducted, as accurately replicating real-world anomalies is challenging, and it is preferable to validate the algorithm with actual anomalies in the future.

# 11 Classification

This section outlines the planned approach for implementing the anomaly classification method. However, this classification was not implemented in this thesis for reasons discussed in Section D.4, “Anomalies Classification”.

The classification strategy follows a rule-based classifier white-box approach, as detailed in Section B.3.4.3 of the literature review. This method offers transparency by providing clear criteria for how anomalies are classified and avoids the need for extensive labelling of past anomalies, which would be essential for training a supervised black-box classifier.

Two types of rules were identified to classify the two types of anomalies discussed in Section B.2.1.3: "Direct Under-Production" and "Degradation Over a Time Period". The first set of rules is designed to classify anomalies that have an immediate impact on production, such as a disconnected module. The second set of rules analyses the trend of losses over specific periods (e.g., a year, month, or week) to detect long-term anomalies, such as plant growth causing shading or abnormal aging of system components.

The percentage of the observed loss helps determine the level at which the anomaly occurred, based on the known structure of the PV system. For instance, if an array typically contributes 30% of the total PV system’s energy production, and a loss of approximately 30% is detected, we can infer that the anomaly is likely localized to that array.

# 12 User Interface

A user interface application has been developed to visualize both the final and intermediate results of the anomaly detection process. Screenshots of the UI are available in Appendix H.7. This interface allows users to select the target PV system of interest in the upper left corner, show metrics on the system on the upper right corner, and give access to some views. The following information are accessible:

**Metric: Half Sibling Regressor - Train Set/Test Set MAPE:**
This metric shows the Mean Absolute Percentage Error (MAPE) of the regressor in estimating the production on both the training and testing sets. A higher error indicates a lower accuracy in estimating what a system should produce. It is normal for the training set error to be lower than the test set error. However, if the model significantly outperforms on the training set, it suggests overfitting, indicating that the hyperparameters need adjustment.

**Metric: Normalizer Tuning – Static System Loss:**
This displays the losses that the tuned normalizer applied to the system to fit the physical model to the actual data. Ideally, this should reflect the static losses of the system, but discrepancies can occur if the physical model does not accurately represent the real conditions.

**View: Absolute Daily Energy:**
This view allows users to see the actual daily production, the maximum possible production, and the production estimates generated by the Half Sibling Regressor.

**View: Normalized Daily Energy:**
Similar to the "Absolute Daily Energy" view, but the values are normalized between 0 and 100% based on the maximum possible production.

**View: Normalizer Tuning:**
Provides detailed information to understand and debug the tuning process of the normalizer. It shows the estimated maximum production before and after tuning the losses, the measurements used for tuning, and those considered outliers.

**View: All Normalized Energy:**
Displays the normalized production of the target system and all the neighbouring systems selected by the Half Sibling Regressor for estimation. The darker the value, the more significant it is in estimating the target system's production. This visualization helps understand the variation between systems and identify if a system is overproducing or underproducing relative to others.

**View: Dynamic Losses:**
This view shows the daily losses for a system, i.e., the difference between estimated and measured production, as output from the comparator. It also highlights the days statistically classified as abnormal.

**View: All Missing Values:**
Visualizes the amount and distribution of missing values across the data.

**View: Similar Neighbouring Systems:**
Indicates the extent to which each neighbouring system was useful in estimating the target system's production.

# 13 Data Storage

In our software, historical measurements are stored in a CSV file, while metadata is stored in a JSON file. However, it is possible to interface with other data sources depending on the partner's infrastructure. The tuning parameters of the normalizer are stored within the metadata, and the normalizer itself is not stored separately. This is because it can be reconstructed instantly from the metadata during the production phase. The Random Forest Regressors, on the other hand, are serialized and saved as `.pickle` files, a format for storing Python objects. This approach is essential to avoid retraining the regressor for each PV system during production every days, ensuring efficiency and

# D. Results & Discussion

In this chapter, we present the results obtained during this thesis and provide a discussion of these findings. We begin by examining the primary outcomes: the estimation of expected production and the detection of anomalies. We then discuss the performance of the Normalizer and the Half-Sibling Regressor. Finally, we present statistics to address the following questions:

- What is the impact of unlabelled anomalies present in the historical data?
- How do errors in metadata and measurements impact results?
- How does the length of the data history impact results?
- What is the effect of the number of neighbouring systems?
- How does the geographical distance of neighbouring systems impact results?
- Which neighbouring systems are selected?
- How many neighbouring systems are chosen?
- How long does it take to train and execute the algorithm?

## 1 Main Results

Our algorithm, when provided with the daily production data of the target PV system and its neighbouring systems, is capable of estimating:

- Expected production for the day, month, and year.
- Underproduction for the day, month, and year.
- Alerts for potential anomaly detection.
- Alerts for PV systems performing abnormally poorly from its start.
- Current performance relative to the maximal capacity of the system.
- A list of systems similar to the target PV system.

Figure 28 illustrates these results, showing the measured and expected production, the maximum possible production, and the detected anomalies. Additionally, the monthly forecast provided by PV GIS is displayed in this figure, allowing for a comparison of our results with those from PV GIS (European Commission, 2024), a predictions currently used by our partner.

The two following sections will detail the accuracy of our production estimates and the precision of our anomaly detection.

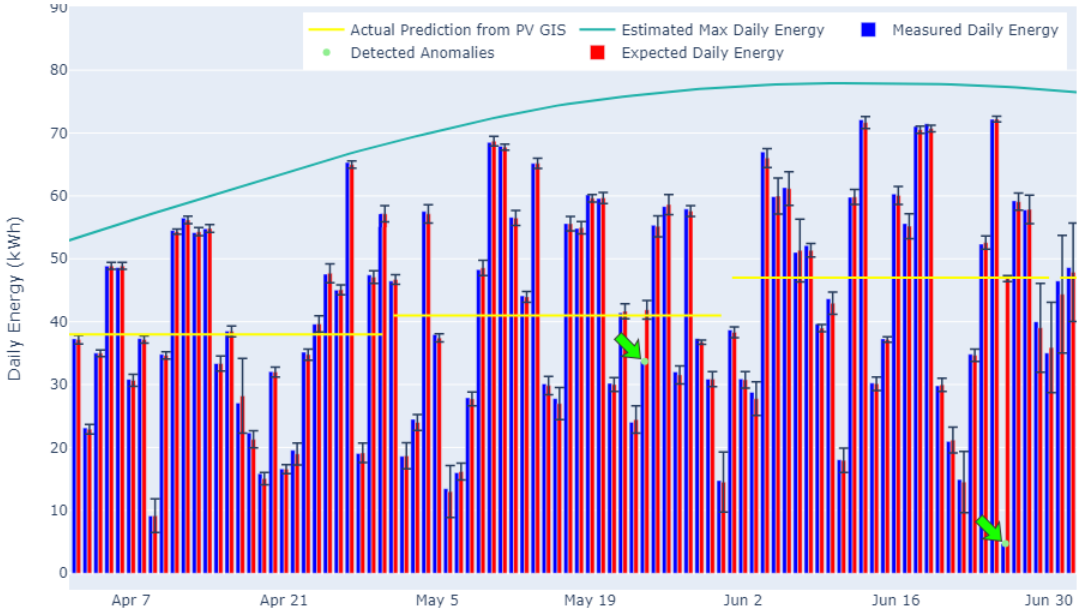



Figure 28 Expected Daily Energy and Detected Anomalies

# 2 Expected Daily Production Estimation

Estimating the expected daily production of a PV system is crucial, as it forms the foundation of our anomaly detection and classification processes, and enable the calculation of the monthly and yearly production estimations. The Mean Absolute Percentage Error (MAPE) and its standard deviation determine the accuracy of this estimation for each PV system. All performance of subsequent anomaly detection and classification steps depend on this initial accuracy; if the accuracy is low, we will not be able to detect anomalies with a lower impact on production, such as progressive degradation.

On average, the MAPE for all systems is 4.38%, which indicates that the regressor's daily estimation error is approximately ±4.38% to the real measure. The standard deviation of this error is 4.60%. The descriptive statistics and the box plot of these results are shown in Table 5 and Figure 29, respectively.

Table 5 Descriptive Statistic of the Half Sibling Regressor Accuracy

| Count | Mean | Std | Q1 | Median | Q3 |
|---|---|---|---|---|---|
| 326 | 4.38% | 4.60% | 2.24% | 3.72% | 5.32% |

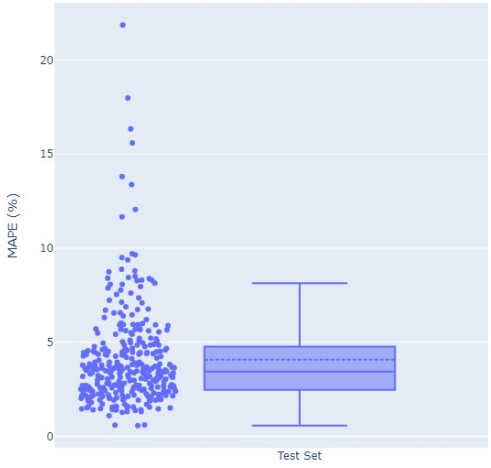



Figure 29 Boxplot of the Half Sibling Regressor Accuracy

Comparing our results with similar studies using the same principles and metrics, SolarClique achieved a MAPE of 7.81% by comparing the production of 88 PV systems in Austin, Texas (Iyengar et al., 2018). In contrast, SunDown achieved a MAPE of 2.98%, but it compared the production of each module within a single PV system (Feng et al., 2020). Our results are within a comparable range to these studies. The difference in outcomes can be explained by several factors: in our study, unlike SolarClique, we had more data, utilized a normalizer to aid the Random Forest Regressor in production estimation, and tuned the hyperparameters of the regressor. The lower error in the second study can be attributed to the use of neighbouring module instead of neighbouring PV systems, leading to less variation in characteristics such as tilt, azimuth, losses, module performance, and geographical distance.

A detailed manual analysis of our results for each systems reveals that some PV systems show excellent accuracy, with a MAPE of 1.3%, while others perform poorly, with a MAPE of 23%. The best results are observed when several systems with similar configurations are installed in close proximity, and their data contain minimal measurement errors and anomalies. Poor results typically occur for several reasons: either the target system lacks nearby neighbours for a reliable Half-Sibling Regression (1), or the normalizer inaccurately estimates the system's maximum production (2). This latter issue may arise from erroneous metadata, unique PV system configurations that the normalizer cannot simulate, or systems that have changed configuration over time. More details on these issues are provided in the normalizer results section D.5.

## 2.1 Improvement Compared to Current Monitoring performed by partner company

Currently, the partner uses estimates provided by PV GIS (European Commission, 2024). This tool estimates monthly production based on the average weather conditions over the past ten years. However, this system does not utilize current weather conditions to estimate production. As shown in Figure 28, the estimate is only provided for an entire month and does not reflect the actual expected production. In contrast, our estimation provides daily values based on the actual weather conditions of that day, significantly improving the accuracy of the production estimates.

# 3 Anomalies Detection

To evaluate the accuracy of our anomaly detection method, we simulated anomalies in our dataset, as we lacked real, labelled anomalies. As a reminder, we randomly introduced simple anomalies in 5% of the test-set days. Each anomaly corresponds to a daily underproduction of 30%.

The confusion matrix of the results is presented in Table 6. Our detector successfully identified 97.4% of the simulated anomalies; however, 1.2% of the days without simulated anomalies were incorrectly flagged as anomalous. This means that our detector has an overall accuracy of 98.7% on our simulated anomalies.

The 368 days without simulated anomalies that were detected as anomalous could partly result from pre-existing anomalies in the input data. To understand the cause of these false positives, one would need to perform a detailed hourly analysis of the production data for each day flagged as anomalous. This analysis was not performed in this thesis and is discussed further in the "Limitations and Future Directions" section.

The 43 days with simulated anomalies that were not detected are due to two primary factors. First, on days when the production was already very low or non-existent, a 30% reduction might not be significant enough to be detected. Second, systems with low regressor accuracy, as discussed in the previous section, failed to detect a 30% drop in production.

It could also be beneficial to simulate different types of anomalies, particularly those with less than a 30% impact on production, to analyse the detector's accuracy related to the severity of the anomaly. However, this was not possible within the timeframe of this study.

Table 6 Confusion Matrix of the Anomalies Detection

| | | Detected Anomalies | |
|---|---|---|---|
| | | No Anomaly | Anomaly |
| Simulated Anomalies | No Anomaly | 30’602 | 368 |
| | Anomaly | 43 | 1’587 |

# 4 Anomalies Classification

A more detailed classification, beyond simply distinguishing between "normal" and "anomalous," was not implemented in this thesis. Consequently, the output is limited to anomaly detection rather than a more nuanced classification. There are several reasons for this decision:

- The primary focus of this work was to develop the most accurate Half-Sibling Regressor possible, as previously discussed. The precision of the regressor is crucial for classification because if the daily losses due to anomalies are not accurately known, it becomes impossible to determine their cause. As a result, limited time was allocated to implementing the classification of our method.
- It was considered early in the project that the precision offered by our Half-Sibling Regressor, with daily estimates varying on average by ±4.4%, would not allow us to detect gradual degradation over several days of just a few percent, such as abnormal aging or plant growth. Feng et al. also noted in their SunDown study that they could not detect long-term degradation using neighbouring systems (2020).
- Additionally, since the available data was not labelled with information on when an anomaly occurred or the type of anomaly, any classification would have been theoretical, using simulated anomalies, and not reflective of real-world conditions. Furthermore, we would not have been able to measure the accuracy of our classification. Our partner also did not have further historical data on anomalies they faced. As we will discuss in the "Future Work and Recommendations" section, it will be necessary to test our algorithm in the field and implement classification rules iteratively, based on feedback from industry professionals.

# 5 Normalization

The normalizer generates two valuable results that can enhance the monitoring of photovoltaic (PV) systems:

1. **Maximum Possible Production Estimation**: The normalizer provides information on the maximum possible energy production of a PV system at any given time of the day or year. This data can be useful for monitoring purposes, as it allows for an assessment of the current performance of a PV system relative to its total capacity.
2. **Detection of Initial Performance Issues**: During the tuning process, the normalizer calculates the static losses of a PV system. This value can be used to automatically detect anomalies from the moment a PV system is installed if the calculated loss deviates from expected norms.

However, evaluating the quality and accuracy of the normalizer is challenging due to the lack of comparative values. The only viable approach is to manually assess the maximum possible daily production curve produced by the normalizer against the data, for each of the 326 systems. We observed five cases where the maximum production estimates did not align with the data trends. These cases are illustrated in Figure 30:

- **Case 1-2**: The summer-winter seasonality curve is either too pronounced or not pronounced enough, resulting in an overestimation or underestimation of relative production in summer or winter. This issue mainly arises from incorrect metadata about the systems, which skews the normalization. For instance, inaccuracies in the tilt angle information in the metadata can affect the estimation of seasonal production variations.
- **Case 3-4**: The physical simulator in the normalizer fails to accurately estimate the system's maximum production. This is particularly true for systems with complex configurations, such as panels installed vertically on facades (at 90°) facing different orientations.
- **Case 5**: The configuration of the system has changed over its lifetime, altering its characteristics and making our previous estimates obsolete. As shown in the graph, the production from summer 2023 does not match that of summer 2024. In such cases, the production history that no longer reflects the current configuration should be removed.

Based on these observations, the distribution of normalization issues among the 326 analysed systems is as follows:

- **Case 1**: Overemphasized seasonality of the normalizer - 16 systems
- **Case 2**: Underrepresented seasonality of the normalizer - 11 systems
- **Case 3-4**: Incorrect maximum production profile - 4 systems
- **Case 5**: Configuration change - 1 system

Thus, 294 systems were correctly normalized, providing accurate estimates of the maximum possible energy. It is important to note that even if the normalization is not highly accurate, anomaly detection remains feasible, although with reduced accuracy.

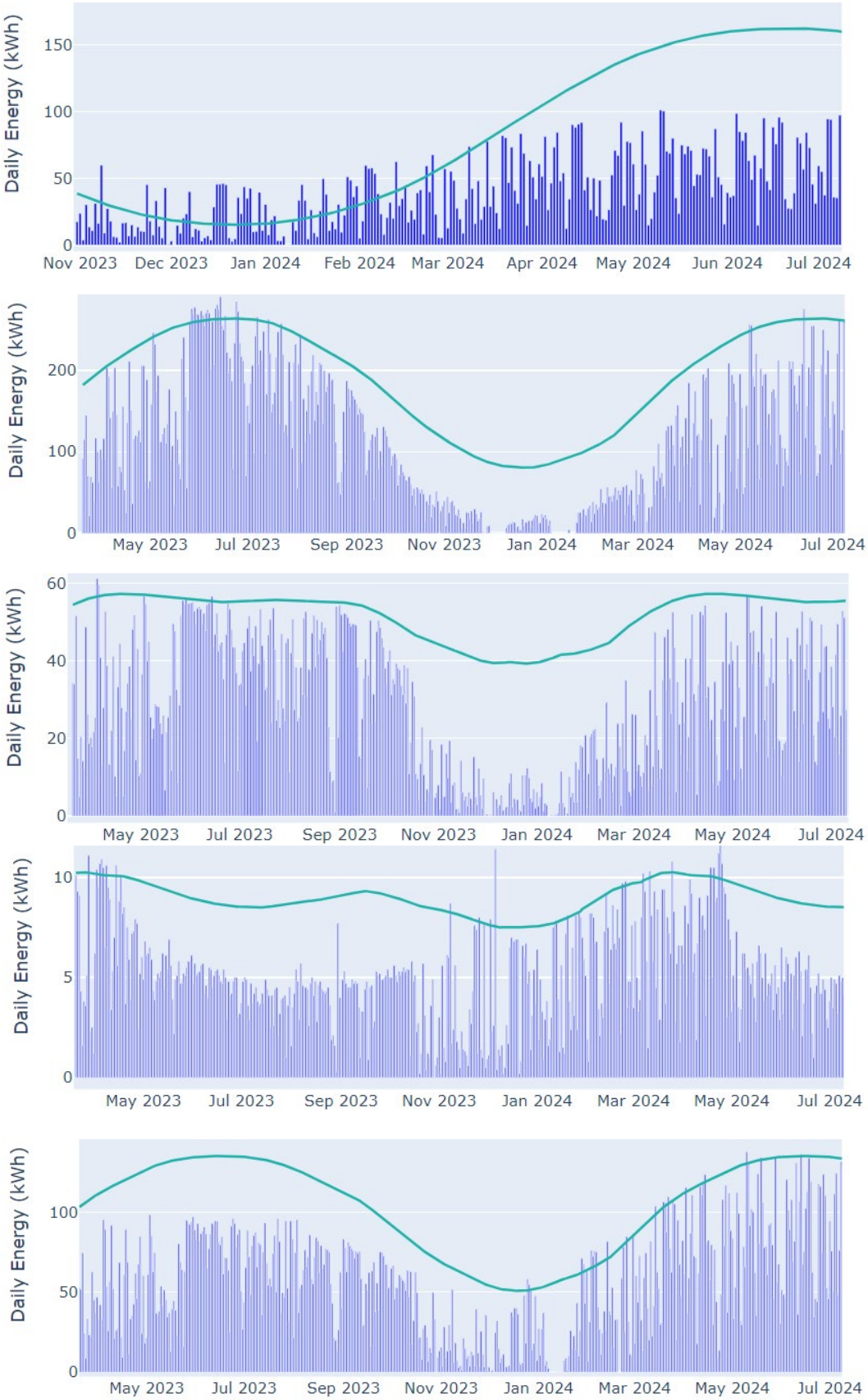



Figure 30 Examples of poor-quality estimates of maximum daily production (green line), leading to a distorted normalization.

## 5.1 Impact of the normalization

After completing the algorithm, with the Half Sibling Regression, we assessed the actual impact of normalization on the accuracy of the estimated daily production.

To do this, we trained the Half-Sibling Regressor directly on raw data, without normalizing by the maximum daily production, and compared the two sets of results.

The results indicate an average improvement in MAPE (Mean Absolute Percentage Error) of 0.56% across all systems and a reduction in standard deviation by 1.16%. This surprising modest improvement could be explained these factors:

1. When the Random Forest Regressor is trained directly on raw data, it may learn directly both the system-specific characteristics, and the relationships between systems, effectively replacing the need for a normalizer.
2. If the dataset is large enough, the regressor will likely find at least one system with a similar configuration to base its estimation on. The normalizer, therefore, becomes useful only for systems without similar neighbours.
3. Another reason could be that the normalizer enhances the accuracy for some complex systems lacking similar neighbouring systems by standardizing the data, but on the other hand, it reduces the accuracy for systems where the normalization was ineffective, as discussed earlier. So, in the end, the positive impact and the negative impact balance out, without improving the final performance.

A detailed case-by-case analysis would be needed to better understand the actual impact of normalization on the final result.

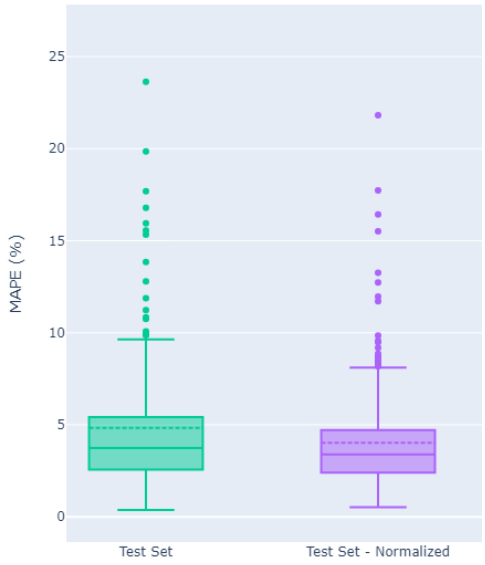



Figure 31 Box plot of the MAPEs, using the raw and normalized data of the test set

Table 7 Descriptive statistics of the MAPEs, using the raw and normalized data of the test set

| | Test Set MAPE | Test Set MAPE Normalized |
|---|---|---|
| Mean [%] | 4.94 | 4.38 |
| Std [%] | 5.76 | 4.60 |

## 5.2 Improvement Compared to Current Monitoring performed by partner company

Although the normalizer only marginally improves the estimation of the expected production, it provides valuable supplementary information for the partner's monitoring platform. Currently, the partner's monitoring tool displays the actual production relative to a system's peak power $\left(\frac{kW_{actual}}{kWp}\right)$ to gauge the system's current performance. However, peak power remains constant, regardless of the system's actual capacity, which fluctuates over time. Whether it is summer, winter, morning, or noon, this metric is always relative to peak power.

Using our normalizer, which provides the maximum currently possible power, the relative value would accurately reflect the system's current performance, regardless of the time of day or season.

# 6 Half Sibling Regression

The final results from the Half-Sibling Regressor (HSR) are the estimated expected daily production values already discussed in Section D.2. In this section, we will focus on comparing different machine learning models tested for creating the HSR and the hyperparameter tuning process.

## 6.1 Comparison between ML Models

To determine the most suitable machine learning model for the HSR, several models were tested, as outlined in Section C.7.2. Both the Random Forest and Pairwise Linear Regression models were trained and tested on the complete dataset because they natively handle missing values. For other models, it was necessary to filter out systems and days with too many missing values to work with complete data. The results of the accuracy of each model are displayed in Figure 32.

The Pairwise Linear Regression model, which we initially designed and follows an intuitive approach, performs reasonably well compared to a standard Linear Regressor, with the added advantage of handling missing values. However, it loses significant information when approximating the relationship between the production of two neighbouring systems with a linear regression. In contrast, a Random Forest model retains more information and can identify multiple relationships from several systems. Therefore, the Random Forest provides the best results, as already suggested by several studies (Iyengar et al., 2018; Rapaport & Green, 2021). The K-Nearest Neighbours (KNN) model also appears to be a good solution, but it cannot work without further adaptation to handle missing values.

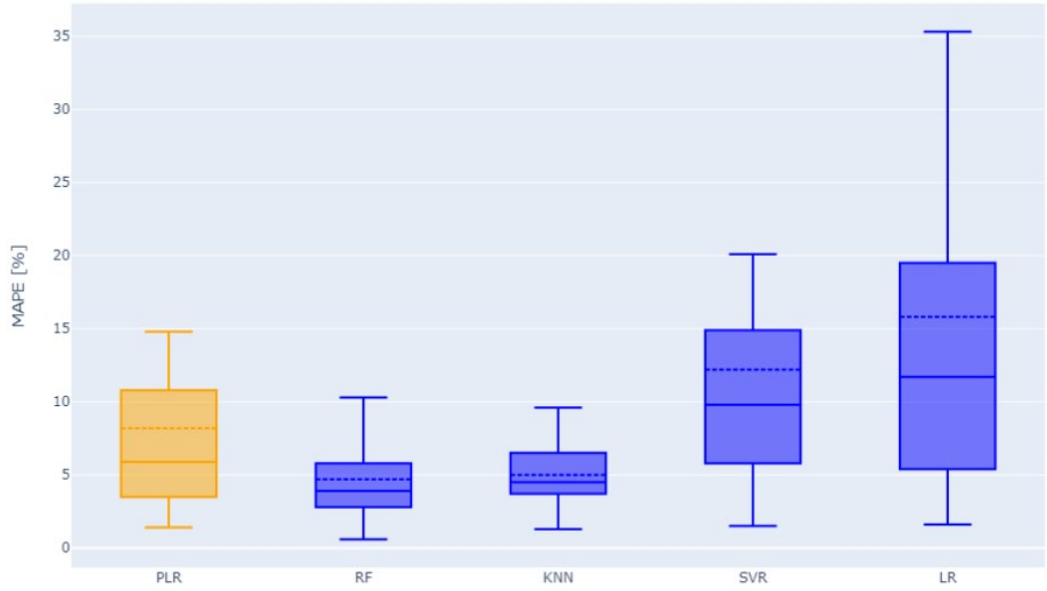


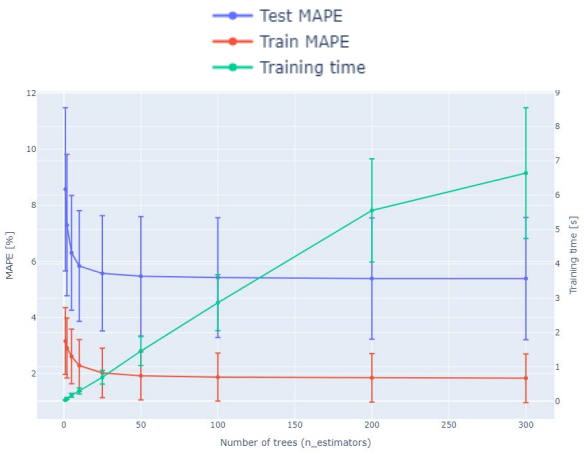

Figure 32 Boxplot of the accuracy on the Half Sibling Regressors, using different Regressors Models

## 6.2 Hyper-parameters Tuning

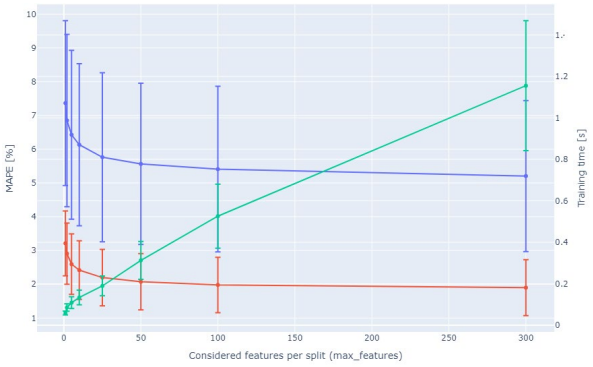
As discussed in Section C.7.3, every ML model has hyper-parameters that must be empirically tuned. We used grid search and cross-validation to tune the hyperparameters for each model. Each model was tuned using data from the 325 other systems and the entire training set, which included about 365 days.

Figure 33 shows the impact of each parameter on the Mean Absolute Percentage Error (MAPE) for each system. The dots represent the mean values, and the error bars indicate the standard deviation. The MAPE for both the training and test sets are displayed to spot over-fitting, along with the training time required per system.

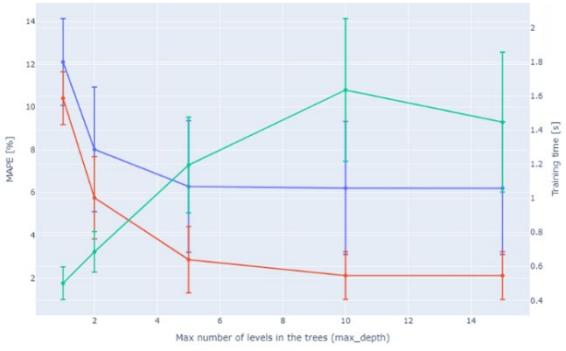
**Number of Trees**
This is the most critical parameter for a Random Forest model. Increasing the number of trees improves the model's ability to generalize statistically. We see that the error decreases as this parameter increases, but the training time increases linearly. Since the error stabilizes at around 50 trees, this number was chosen.

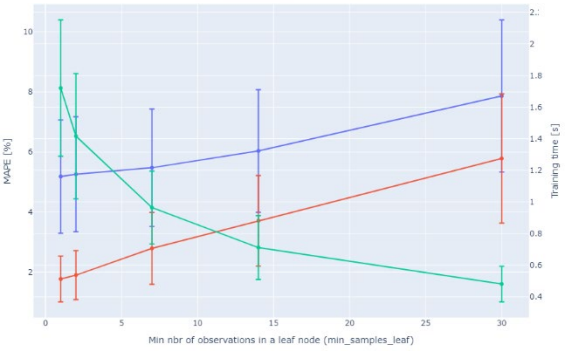



**Max Features**
This parameter represents the number of neighbouring systems (features) considered at each node of a tree when deciding how to split the observations. Similar to the number of trees, the error decreases as this parameter increases, but the training time also increases linearly. The error stabilizes at around 100 features, so this number was selected.




**Tree Depth**
**Minimum Samples to Create a Node**
**Minimum Samples in a Leaf**
These three parameters control the depth of the trees. A deeper tree will have many leaf nodes, potentially with only a single observation in each. The shallowest tree will have only two branches, splitting the observations into two leaf nodes, with the predicted result being the average of all target values in the leaf. Generally, deeper trees produce better results. The default parameters (max_depth = full depth, min_samples_split = 2, min_samples_leaf = 1), which create deeper trees, were retained. Regarding training time, deeper trees require more time to train. If time is a constraint, the tree depth could be reduced.



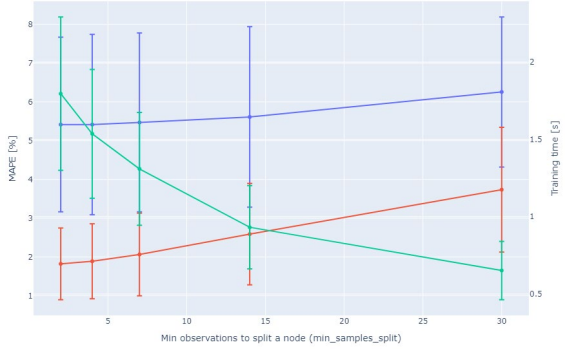




Figure 33 Illustrations of the results of the hyper-parameters tuning.

# 7 Impact of the presence of unlabelled anomalies in the historical data

## 7.1 In the training set

It is known that the input data contains anomalies because it is real-world data; however, we do not know exactly when these anomalies occurred. As a result, our Half-Sibling Regression models are trained on anomalous data, which reduces their accuracy. When a PV system's measurements include days with anomalies, the model may learn these anomalies and attempt to predict them. Consequently, in future situations similar to previous one when anomalies occur, the regressor might predict a production level influenced by the past anomaly, thereby reducing its accuracy.

To improve the regressor's accuracy, it would be beneficial to manually label days with anomalies in the historical data. Once identified, these anomalous days could be excluded from the training set. If manual labelling is not possible, another approach is to define basic rules that exclude most anomalies. For instance, in the current implementation, we filter out all measurements where the daily production is less than 1% of the system's maximum possible production, as these measurements are predominantly anomalies.

By refining our approach to exclude anomalous data more effectively, we could enhance the robustness and accuracy of the Half-Sibling Regression models, leading to better anomaly detection and classification.

## 7.2 In the test set

Anomalies in the test set will not reduce the actual accuracy of a model but will reduce our measurement of its accuracy. Indeed, our model will try to predict production without anomalies, whereas the measurement includes some, increasing the MAPE of our regressor.

# 8 Impact of erroneous Input Data & Metadata

The metadata provided by our partner, along with some of the measurements, contain errors that do not reflect the actual system configurations and production. These inaccuracies can affect the quality of the normalizer and the regression, thereby reducing the performance of our algorithm. This section will examine these errors in detail.

## 8.1 Erroneous Metadata

As discussed earlier in Section C.1 on input data and in Section D.5 on normalization, the metadata is sometimes inaccurate. This is because the metadata is manually entered by photovoltaic installers, who may make mistakes or enter information in a non-standardized way.

Our filtering step initially attempts to exclude systems with detected errors in their metadata. Out of the 451 systems provided, 95 were excluded through this filtration process. The reasons for these exclusions are listed in Appendix H.4. Additionally, our normalizer identifies systems that could not be properly tuned because the simulated curve of maximum possible power production diverged significantly from actual measurements. In this case, the normaliser considers that normalisation with the metadata provided is impossible and the system is excluded. Five additional systems were removed this way, with details available in the appendix.

Nevertheless, some errors in the metadata persist, impacting normalization accuracy, as observed in Section D.5.

## 8.2 Erroneous Measurements after a loss of connection

It has been observed that when the measuring meter loses connection, the next available measurement, recorded on the day the connection is restored, will sum up all the energy produced since the loss of connection. This results in erroneous measurements on both the day of the disconnection and the day of reconnection, which could skew the regressor's performance. However, these types of outliers can be easily filtered out during our data filtration step.

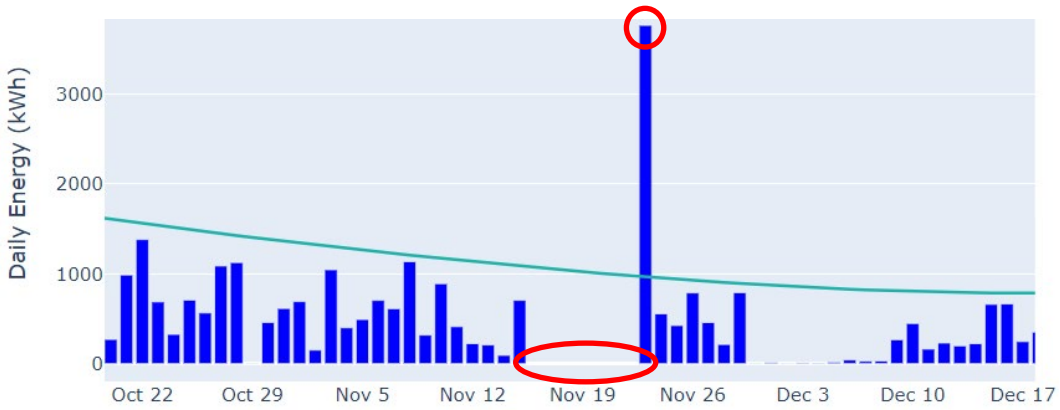



Figure 34 Illustration of the erroneous measure after the meter loss connexion (in green, the maximal possible production).

## 8.3 Erroneous Measurements with a battery behind the meter

When a DC battery is present before the meter, it becomes impossible to isolate anomalies related to the production of the PV modules because the measurements are influenced by both the production and the house's self-consumption, which we cannot predict. Figure 35 illustrates this scenario, showing that the solar production profile does not align with a standard PV system profile. These erroneous measurements degrade the anomaly detector's accuracy.

To address this, we would need information indicating the presence of a DC battery in the system to exclude it from our training set. Alternatively, with access to hourly data, we could exclude installations that produce power at night, which would indicate the presence of a DC battery.

Interestingly, our results for systems with a battery are not as poor as expected. The two systems we manually identified as having a DC battery showed MAPEs of 11.2% and 14.3%, which still allow for anomaly detection. This is because the battery generally discharges the stored energy during the same day or the next day, meaning the total daily energy is only minimally affected. Since our anomaly detection is performed on a daily basis, rather than hourly, the battery's impact on our results is minimized.

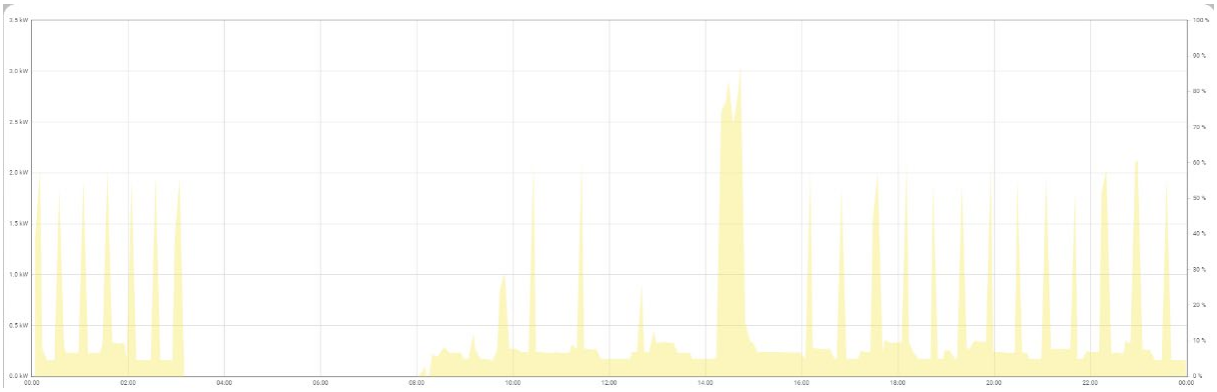


Figure 35 Illustration of the erroneous measure of a system with a battery behind the meter

## 8.4 Double Measurements per day

For some days, multiple measurements are recorded for the same day. A list of these duplicate measurements is provided in Appendix H.2. Upon investigation, it was found that these double measurements occur on days when there is a time change, twice a year. The first measurement is the standard daytime production, while the second is a counting artifact occurring at night due to the time change. The second measurement is generally zero since there is no production at night, except for systems with DC batteries, as discussed in the previous section, which may produce at night.

# 9 Impact of the amount of historical data

We investigated the impact of the amount of historical data on the model's performance. A model that requires less historical data has the advantage of being operational soon after the installation of a new PV system.

To assess this, we trained a model for each of the 326 systems, using the previously defined hyperparameters and the data from the remaining 325 systems. These models were then trained on varying amounts of historical data: 2 days, 1 week, 2 weeks, 1 month, 6 months, and 1 year.

The box plots and descriptive statistics of the resulting Mean Absolute Percentage Errors (MAPE) are shown below. As observed, the error decreases as the amount of historical data increases. While the error stabilizes, it continues to decrease slightly over time. For instance, the MAPE is 6.00% with 1 month of data, which decreases to 4.89% with a full year of data. However, we also see that a large amount of data is not necessarily required; as we achieve a MAPE of 9.32% after just 7 days.

Regarding training time, increasing the size of the historical data slightly extends the time needed for training, but the increase is minor and remains acceptable, allowing it to be considered negligible. Based on these findings, we decided to use all available historical data.

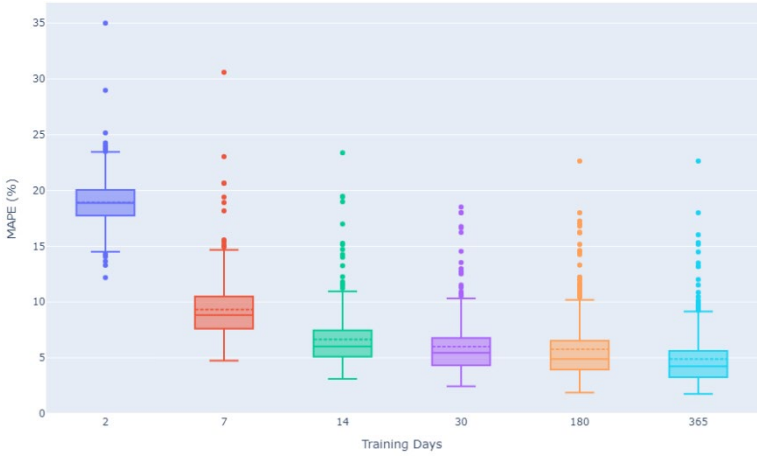



Figure 36 Boxplot of the MAPEs related to the history size

Table 8 Descriptive Statistics of the MAPEs related to the history size

| | 2 | 7 | 14 | 30 | 180 | 365 |
|---|---|---|---|---|---|---|
| **Mean [%]** | 18.93 | 9.32 | 6.64 | 6.00 | 5.77 | 4.89 |
| **Std [%]** | 4.40 | 5.68 | 5.26 | 5.24 | 6.03 | 5.4 |
| **Training time [s]** | 0.48 | 0.45 | 0.46 | 0.51 | 0.64 | 0.77 |

# 10Impact of the amount of neighbouring PV systems

We also examined how the number of neighbouring PV systems provided to the model affects its accuracy. The goal was to determine whether it is better to use all available neighbouring systems during model training or just the top 10 most correlated systems.

To assess this, we trained a model for each of the 326 systems using the previously determined hyperparameters and the full history of data. These models were then trained using an increasing number of neighbouring systems, starting with those most highly correlated with the target system.

The resulting boxplots and descriptive statistics for the MAPEs are shown below. The results indicate that as more neighbouring systems are provided, the error decreases, reaching a low of 4.36%. However, the error stabilizes after including about 25 systems and even slightly increases thereafter, rising to 4.45% when 300 neighbouring systems are used. This increase in error is likely because the additional systems are not correlated with the target system and therefore do not contribute to a more accurate production estimate; in fact, they might reduce the performance of the Random Forest Regressor by adding complexity and useless features. Moreover, the training time increases significantly with more systems, from 0.7 seconds for 25 neighbouring systems to 5.4 seconds for 300 systems.

Based on these findings, we decided to use the 50 most correlated neighbouring systems to train the model. This approach allows us to achieve close to the optimal result while keeping the training time low.

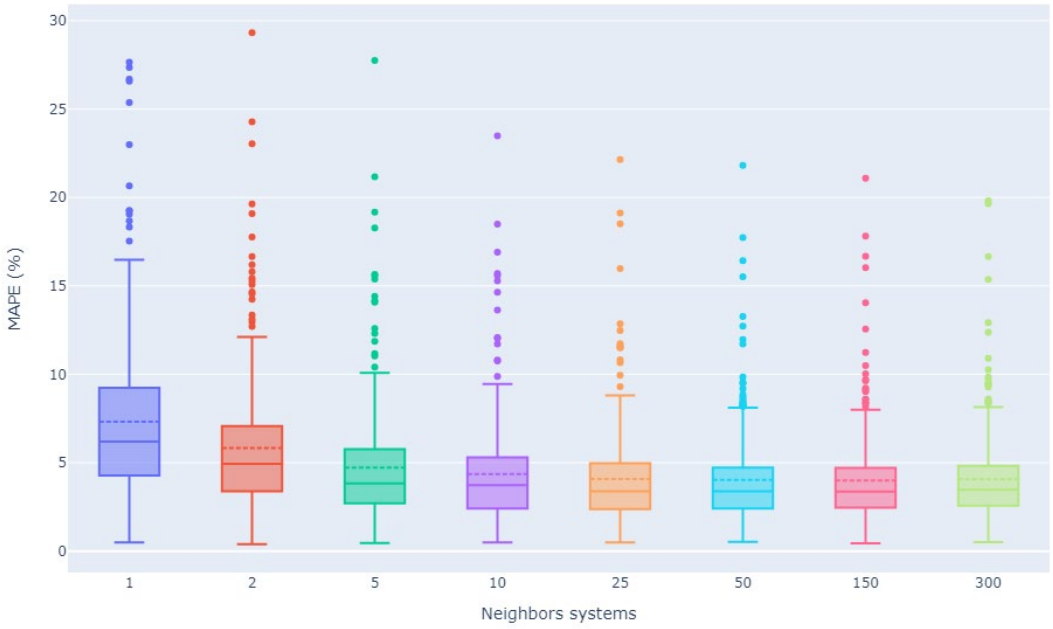



Figure 37 Boxplot of the MAPEs related to the different number of neighbouring systems available.

Table 9 Descriptive statistics of the MAPEs related to the different number of neighbouring systems available.

| | 1 | 2 | 5 | 10 | 25 | 50 | 150 | 300 |
|---|---|---|---|---|---|---|---|---|
| Mean [%] | 7.72 | 6.25 | 5.13 | 4.75 | 4.48 | 4.41 | 4.36 | 4.45 |
| Std [%] | 6.99 | 6.62 | 6.21 | 5.83 | 5.88 | 5.80 | 5.51 | 5.43 |
| **Training time [s]** | 0.19 | 0.20 | 0.26 | 0.38 | 0.70 | 1.07 | 2.87 | 5.40 |

# 11 Impact of the geographical distance between PV system

We explored how the geographical distance between a target PV system and its neighbouring systems affects prediction accuracy. The goal was to assess whether closer proximity leads to better energy production estimates.

For each target system, we calculated the weighted average distance to its neighbouring systems, with weights based on each system's contribution to the prediction, as explained in next section D.12. As a reminder, the weighted average consists of:

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} w_i * d_i$$

On average, neighbouring systems were located 6 km away, with the majority within a 3.2 km radius.

Our analysis showed no significant correlation between the distance to neighbouring systems and the MAPE. While systems in close proximity were often chosen first for predictions, the selection of more distant systems did not drastically worsen the results. Some target systems with very close neighbours still exhibited high prediction errors, suggesting that factors other than distance play a more critical role in influencing prediction accuracy.

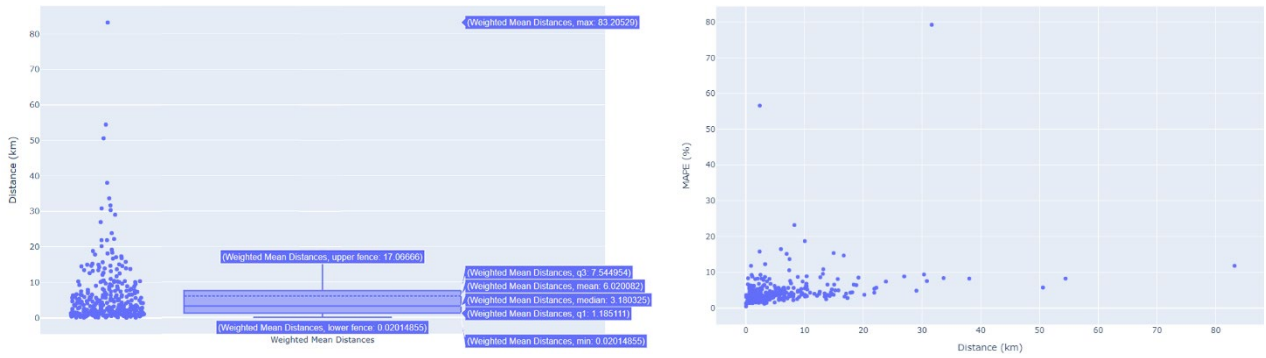



Figure 38 Boxplot of the Mean Weighted Distance to the neighbour systems, and graph of the impact of distance on the MAPE

# 12 Selected Neighbouring PV System

We examined which neighbouring PV systems were selected for each target system's production estimation. This selection is not done manually based on factors like geographical distance; instead, the Random Forest Regressor automatically determines which neighbouring systems are most useful for making accurate predictions. This information is available in the user interface under the "Selected Neighbouring Systems" tab, with an example shown in Figure 39.

Identifying the most important neighbouring systems selected by the regressor is not straightforward. Two primary techniques were used:

**Impurity-Based Importance**
This method relies on metrics directly calculated by the Random Forest Regressor during training. It provides a quick and direct measure of feature importance without additional computation. However, impurity-based importance metric can sometimes be misleading, as it may not accurately reflect real-world behaviours of the regressor. Furthermore, this approach is only possible with a Random Forest, and not with other machine learning models. More explanation is given on the page “Permutation Importance vs Random Forest Feature Importance” by scikit-learn (2022).

**Permutation Importance**
To address the limitations of impurity-based importance, permutation importance can be applied. This technique involves making predictions for the target system while randomly shuffling the data of each neighbouring system, one at a time, to break any correlations. The impact of these shuffles on the

prediction accuracy indicates the true importance of each neighbouring system. If shuffling data from a neighbouring system has an impact on the prediction result, this means that this system is important for prediction. Although this method provides a more realistic assessment of feature importance, and is not dependent on the ML model chosen, it is very computationally intensive, taking an average of 5.2 seconds per system to calculate feature importance.

Since the two methods gave relatively similar results in our study, as we can see in Figure 39, we opted for the impurity-based importance because of its lower computational cost.

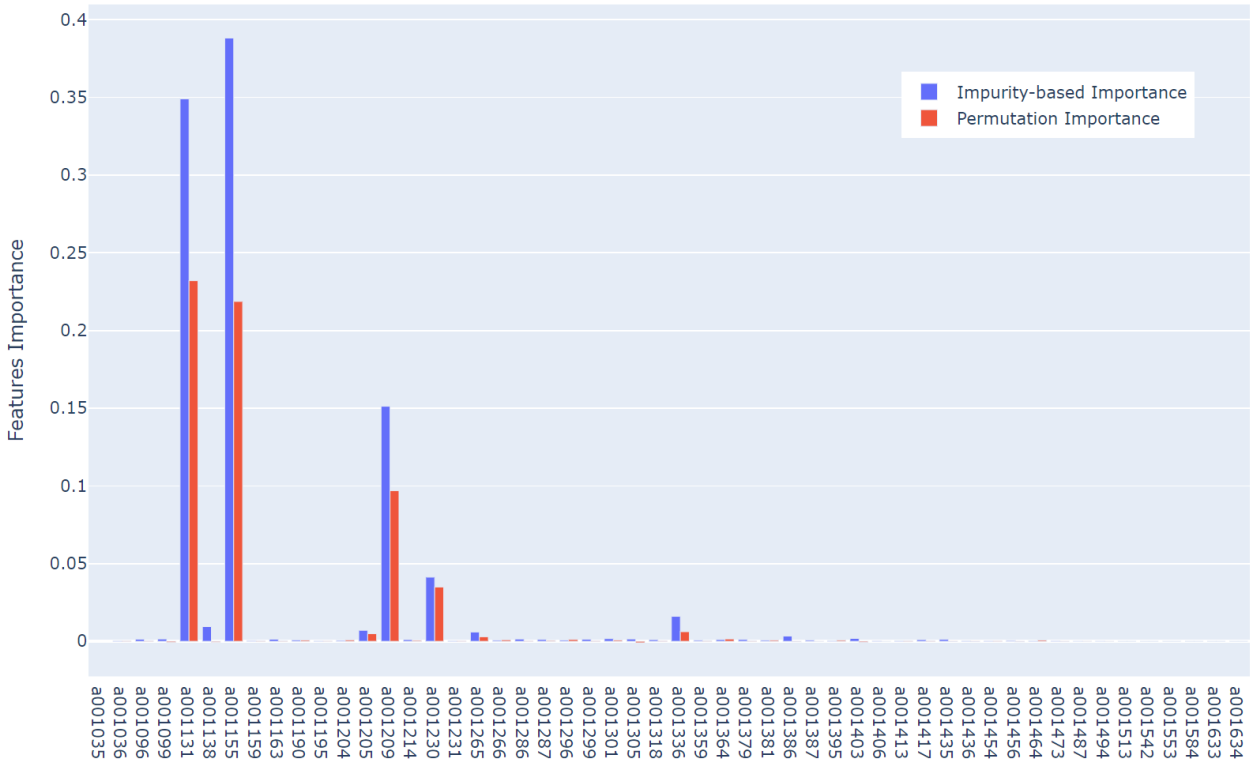



Figure 39 Illustrates the importance of neighbouring PV systems using both impurity-based and permutation importance methods.

# 13 Number of Selected Neighbours

We analysed how many significant neighbouring systems were selected by the Half Sibling Regressor (HSR) out of the 50 provided to predict the target system's production. To determine this, we counted the number of neighbouring systems that contributed more than 10% to the target system's estimation for each model.

Figure 40 show that approximately one-third of the HSR models use two neighbouring systems, and another one-third use three. It is rare for more than three systems to be used in the prediction. This dependence on a small number of neighbours could be a concern, as the model's robustness may be compromised if any of these few neighbouring systems stop providing data.

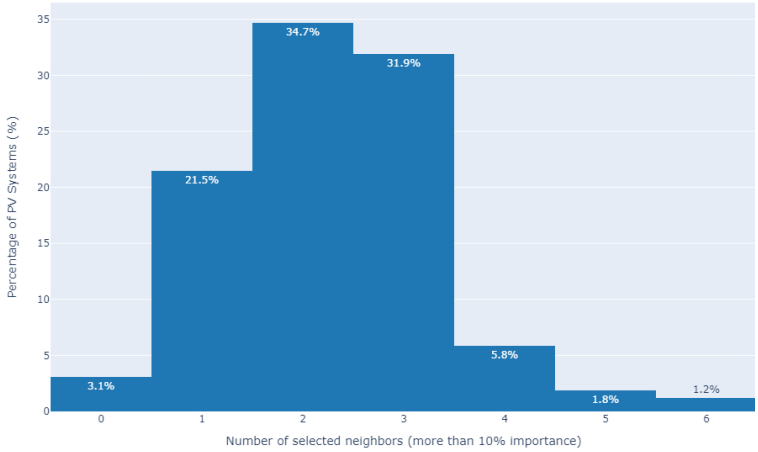



Figure 40 Histogram of the number of selected neighbouring PV systems in the HSR

# 14 Algorithm Timing

The execution time of our algorithm can be broken down into three main components, with the rest of the processes considered instantaneous. During the training phase of each system's model, two parts can consume time:

1. **Normalizer Calculation:** This component is responsible for calculating the maximum potential production for each day. For a system with 365 days of historical data, the normalization process takes an average of 319 milliseconds per system.
2. **HSR Training:** The Half Sibling Regressor involves training the Random Forest Regressor, which takes about 245 milliseconds per system.

Combined, training a model for a single PV system requires approximately 564 milliseconds.

After the initial training, therefore **during production**, the daily runtime to detect anomalies takes around 33 milliseconds per system.

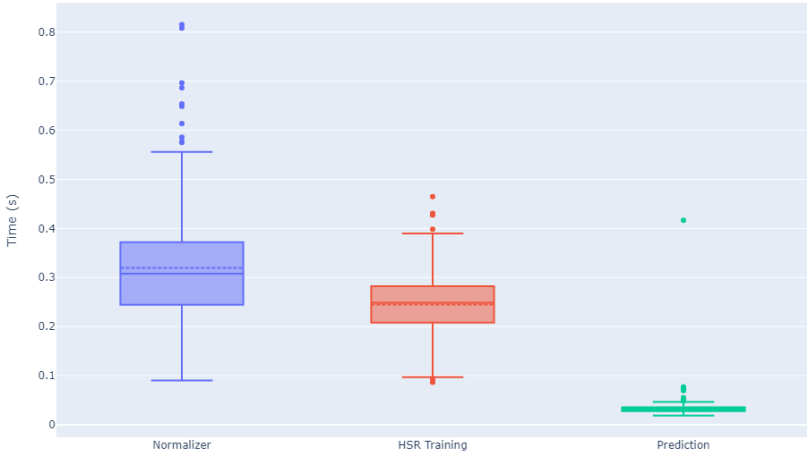



Figure 41 Boxplot of the Normalization, Training, and Prediction Time

# E. Limitations and Future Directions

In this chapter, we outline the various limitations of our method and propose potential directions for future improvement.

## 1 Limitations of daily data usage

A key limitation of our current implementation is the use of only daily measurements for anomaly detection, rather than more granular hourly or minute-by-minute data. This decision was strategic, guided by Mellit et al. (2018), who stated:

> *"Careful considerations should be given to the purpose behind the monitoring before developing a specific tool. The philosophy should be to measure only those variables that are necessary using the minimum acquisition rate required to get meaningful results" (Mellit et al., 2018).*

As daily detection of anomalies is currently sufficient for our partner's needs, implementing a more frequent detection approach was not a priority. While it might be academically interesting to estimate production and detect anomalies at a higher frequency, we chose to focus first on developing a functional daily detection system. Attempting to detect anomalies at hourly or minute intervals introduces greater complexity and risk of failure of the project. Therefore, we opted for daily detection.

For this purpose, we decided to use daily energy measurements rather than more frequent data, so as not to complicate the algorithm from the beginning of its implementation. This decision has both positive and negative consequences for our anomaly detection, which are outlined below.

### 1.1 Daily seasonality is hidden in daily data

A major advantage of using daily data is that seasonal variations occurring within the day, such as shading from a neighbouring building, do not need to be separately detected and modelled. Their impact is already included during the normalisation, reflected as daily losses, which are determined during the normaliser's tuning. This greatly simplifies the normalisation process of the system-specific factors with daily seasonality.

### 1.2 Improving normaliser tuning

However, using hourly measurements could significantly improve the tuning accuracy of the physics-based normaliser. As discussed in Section 0, the current tuning method adjusts model parameters to best fit days assumed to have a completely clear-sky day. This has two drawbacks: it requires a fully clear day, which may be difficult to find in certain seasons (1), and we cannot be entirely sure which days are truly clear (2). We have to assume that days with high production are clear, even if it’s not true.

With hourly data, even a single hour of clear sky could serve as a fitting point for tuning our physical model, providing many more data points to assist the tuning. Also, the shape of the production curve during a day help to get information on a system. This would make the tuning more accurate and enable the adjustment of additional parameters such as tilt, azimuth, or dynamic losses throughout the year. However, since the Half Sibling Regressor performs nearly the same with or without the normaliser, we did not focus on perfecting the normaliser, as this would not significantly impact overall performance.

### 1.3 Improving anomalies detection

Hourly data would also allow for better detection of anomalies that occur only during part of the day.

Consider the simulated anomaly in Figure 42, which shows the production of a PV system under clear-sky conditions with an abnormal 50% drop in production between 13:00 and 15:00.

In the current implementation, which only considers the total daily energy produced, this anomaly would result in an underproduction of only 13.6%, difficult to detect. With hourly data, the 50% drop would be clearly visible, making detection much easier.

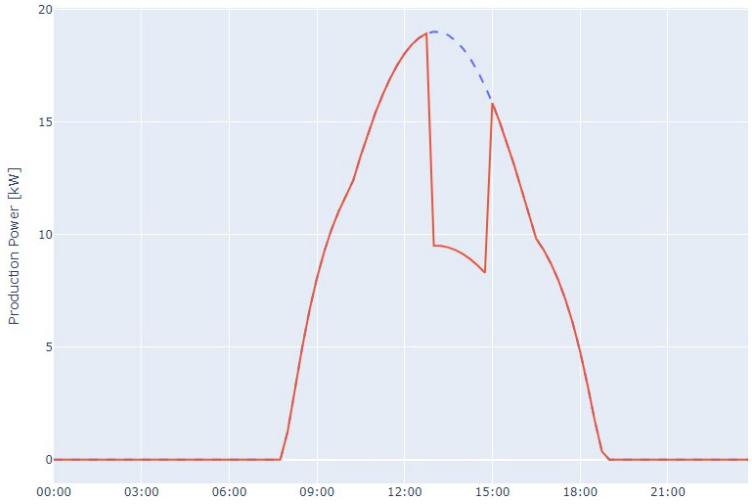



Figure 42 Simulation of daily production with clear-sky weather, with a 50% drop between 13h-15h

# 2 Lack of Anomaly Classification

As discussed in Section D.4 of the results, anomaly classification was not implemented, and only detection was performed. To implement classification, several steps need to be taken first.

One approach would be to simulate certain types of theoretical anomalies and then design the rule-based classifier to classify those simulated anomalies. However, this could be time-consuming and might not work effectively on real anomalies, being therefore not worthwhile.

The alternative approach is to deploy a prototype of our method without a classifier, then manually classify the new detected anomalies, to gather information. A primary classifier can then be implemented, and classification rules can be gradually added and refined. As the classifier becomes more effective, the need for manual anomaly classification will diminish. Eventually, once enough data has been collected, it may be possible to implement a semi-supervised black-box classifier to eliminate the need to create rules manually.

# 3 Regional anomalies will not be detected

One key limitation of our approach is that it relies on neighbouring systems to estimate the target system's production. As a result, any anomalous factor that affects both the target system and its neighbours simultaneously will not be detected as an anomaly. For example, in cases of snow or dirt accumulation that impacts an entire region, our method may fail to identify the anomaly because all systems would show similar underperformance. An anomaly will only be detected when a particular system continues to underperform while the others return to normal production levels.

This limitation results from our method's inherent design: it detects underperformance of the target system relative to neighbouring systems, rather than the real absolute impact of an anomaly on production. No leads have been found to detect regional factors as anomalies, without deviating from the use of neighbouring systems as baselines. Further research is needed to develop a strategy that can address this limitation while still leveraging neighbouring systems for anomaly detection.

# 4 Limitations with DC batteries

As discussed in Section D.8.3, our method is less effective when a DC battery is installed upstream of the meter, potentially making it incompatible with such systems. To address this limitation, three studies have explored techniques for disaggregating the components of signals from elements installed behind the meter.

Mohan et al. (2014) introduced a method for solar disaggregation using whole-house consumption signals. Their approach involves forecasting solar panel production based on net consumption and weather data, which helps accurately isolate solar generation from overall consumption, thus improving utility demand planning. Tabone et al. (2018) expanded on this by developing a real-time method to disaggregate behind-the-meter solar generation using advanced metering infrastructure (AMI) and proxy solar generation data from nearby PV systems. This method effectively identifies homes with solar installations and predicts PV generation with a MAPE between 20% and 50% of average daily PV output. Chen and Irwin (2017) proposed SunDance, a system that also uses AMI and weather data to estimate behind-the-meter solar generation. This black-box approach does not require detailed knowledge of the PV system's physical configuration.

Exploring these approaches could be beneficial for detecting anomalies in systems with batteries.

# 5 Possible biased Test Set

A consideration made towards the end of our study is that our test set consists of the last 100 available days, primarily covering the period from March to June, when production is relatively high compared to periods like November to February. This could positively bias our results, as the model may over perform during times of higher production. According to the partner company, it is common to face greater challenges in estimating production during the winter months. Therefore, testing over an entire year should be conducted for more balanced results.

# 6 Limited number of Neighbouring Systems selected

A current limitation is that our Half Sibling Regressor primarily relies on the production data from only 1 to 3 neighbouring systems for its predictions (see Figure 40). If one of these systems fails to send data or experiences an anomaly, the accuracy of the target system's estimation can be easily compromised. If this becomes problematic, we may need to switch to a regression model that incorporates more neighbouring systems in its estimation, such as the Pairwise Linear Regression described in Section C.7.2.1. However, this could decrease prediction accuracy as lower-quality neighbouring systems would be included in the prediction.

# 7 The role of the Normalizer

The normalizer is a limiting factor in our method as it requires metadata about the systems, such as their tilt, azimuth, or peak power. Although our partner provides this information, it makes our method less universally applicable to any system. It would be beneficial to rely solely on production data, without needing additional metadata. Furthermore, the positive impact of the normalizer on the accuracy of the production estimation appears negligible, as shown in the results, suggesting it might be dispensable.

A detailed analysis should be conducted to assess the actual benefit of using a data normalizer before the Half Sibling Regressor, to understand under which conditions the normalizer improves estimation accuracy. The advantages of the normalizer discussed in the methodology, in Section C.5, “Normalization”, should be verified.

If the normalizer is proven to significantly improve the accuracy of the Half Sibling Regressor, it would be worthwhile to enhance its tuning to increase its reliability and the overall precision, by using hourly data. Conversely, if the normalizer is found to be of little use, it could be removed, eliminating the dependency on metadata. However, removing the normalizer would mean losing the capability to estimate the maximum possible power output, which is a valuable information for system monitoring.

# 8 Limitation of the accuracy of the detection metrics

The accuracy of our anomaly detection has been tested on simulated anomalies, which could bias the metrics. It would be necessary to manually identify and label a substantial number of real anomalies (~100) to obtain a statistically significant and reliable metric.

# 9 Incorporating temporal data as a feature

Currently, the Half Sibling Regressor does not include temporal information, such as the month in which an observation was made, in its features. All observations are considered independently. It might be beneficial to include the month or time of day as features, allowing the regressor to learn annual and daily seasonality on its own, without relying on a normalizer or a Remaining Seasonality Removal step. However, it is not guaranteed that this will improve prediction accuracy.

# F.Conclusions

In this thesis, we developed a novel methodology to improve the monitoring of residential photovoltaic (PV) systems by accurately estimating expected daily production and automatically detecting anomalies. Our approach requires only historical energy production data from the monitored system and its neighbouring systems, making it a versatile and universal solution applicable to a broad range of PV installations. This method specifically targets companies managing residential, small-scale PV systems, typically ranging from 3 to 100 kWp.

**Key Achievements**

1. **Accurate Estimation of Expected Production**:
   Our method, utilizing the Half Sibling Regressor principle and a Physic-based Normalizer, effectively estimates the daily expected production of the monitored PV system based on the energy production data of neighbouring systems. This estimation is robust enough to differentiate the effects of anomalies from other factors, such as cloud cover or system configuration. Consequently, our approach also allows for reliable monthly and yearly production estimates.
2. **Anomaly Detection and Alerts**:
   The system continuously monitors underproduction levels to identify potential anomalies. When abnormal underproduction is detected, the monitoring system could automatically triggers an alert, notifying the managing company of a possible issue. This proactive approach can significantly reduce the Mean Down Time (MDT) and Mean Time To Repair (MTTR) of PV systems, enhancing operational efficiency.
3. **Determination of System's Maximum Capacity**:
   Our solution calculates the theoretical maximum production capacity of a PV system at any given time. This information is crucial for assessing the system's current performance against its true maximum capacity, providing valuable insights into the health and efficiency of the PV system.
4. **Detection of Initial Performance Issues**:
   By analysing the production data of newly installed systems against their theoretical maximum capacity, our method can identify PV systems that are performing poorly from the start. This early detection of initial anomalies enables quick corrective actions, ensuring optimal system performance from the outset.

**Performance Evaluation**

The proposed solution was tested on real data from 326 PV systems, each with an average of 400 days of historical data. The results demonstrated a Mean Absolute Percentage Error (MAPE) of 4.38% in the estimation of expected production, with a standard deviation of 4.60%. Our anomaly detection algorithm successfully identified 97.4% of simulated anomalies, indicating high reliability. The system shows promising results even with a minimal data history of seven days and data from two neighbouring systems within a 6 km radius. However, the results improve with more extensive historical data and closer proximity of neighbouring systems. Additionally, the model handles missing data effectively, ensuring reliable anomaly detection even when some neighbouring systems fail to provide data due to technical issues.

**Future Directions**

While the results are promising, several areas for improvement remain: Utilizing higher-frequency data (e.g., hourly or minute-level) could enhance anomaly detection precision and allow for more frequent monitoring. Also, developing an anomaly classifier would provide deeper insights into the types of issues, facilitating faster and more targeted maintenance. Additionally, field testing with industry feedback is essential to refine the model and develop more sophisticated detection and classification rules tailored to PV system operators' needs.

# 1 Next Steps and Recommendations for the Partner company

Based on the findings of this thesis, we offer the following recommendations to the partner company, for the continued development and implementation of the proposed anomaly detection and classification methodology.

**Prototype Implementation and Integration :**
Currently, the implementation of the solution presented in this thesis is a preliminary model developed for research purposes, primarily to generate initial results. To move forward, it would be beneficial to develop a prototype of the solution that is fully compatible with the company's existing data management and server architecture. This would enable the company to immediately leverage the current results of the solution, test its integration, and gather feedback for iterative improvements.

**Improving Current Monitoring :**
By implementing the method of this thesis, the maximum possible production curve for each PV system will be provided. This allows the **improvement of performance metrics**, such as the ratio of PV power to peak power $\frac{PV_{Power}}{kW_{peak}}$ (see Figure 42). The new metrics would use the actual system's maximum capacity instead of its installed capacity in kWp, leading to more accurate performance assessments. Additionally, systems that are underperforming from the start can be identified comparing their production measurements to their theorical maximum capacity, allowing for the **detection of systems with initial anomalies**.

Furthermore, the estimated expected daily production from our method could replace the current monthly PV GIS estimates, **allowing anomaly detection and providing more precise performance statistics** (see Figure 42).

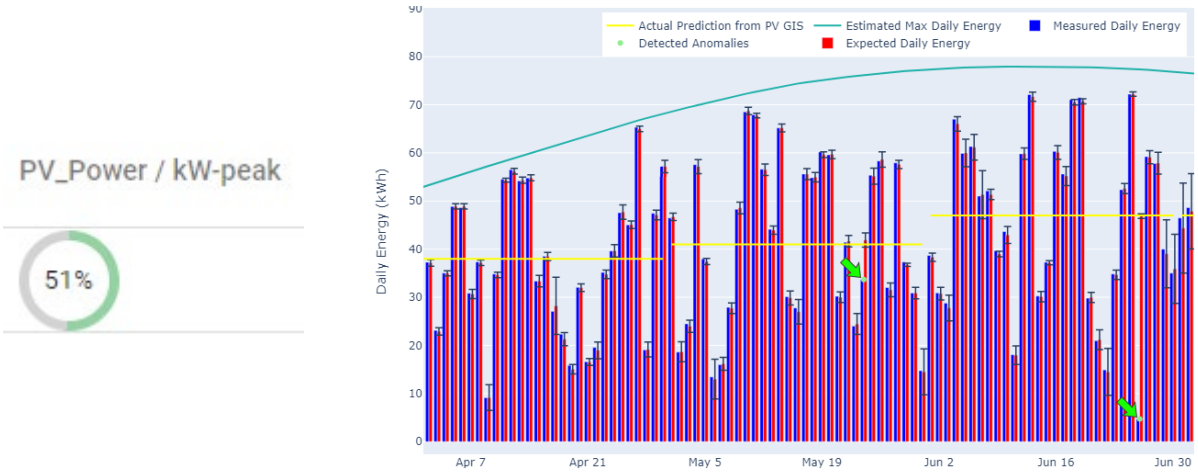



Figure 43 Left: Current performance metrics used in the monitoring tool. Right: Current production estimation from PV GIS and the potential improvements offered by our solution.

**Gathering User Feedback**
Implementing this prototype would also enable the collection of feedback from professionals using it, allowing for the refinement of the monitoring solution to better meet the company's needs. This step is essential for further development, as the solution must be reviewed, adapted, and improved based on real-world requirements and conditions. A simplified labelling system for anomalies should be established. I.e., when the system detects a false positive (an incorrect anomaly detection) or misses an anomaly (false negative), an operator should be able to easily flag these occurrences. Analysing these cases would help correct and fine-tune the algorithm to improve its accuracy.

**Developing an Anomaly Classification System**
Feedback is also crucial if an anomaly classification is to be implemented alongside detection. It is important to develop a system that allows an operator to easily classify and describe anomalies when they occur, to create a knowledge base of anomalies and their impact on energy production. This is

essential if we aim to develop a classifier capable of automatically identifying the type of anomaly. Initially, this classifier would rely on manually defined rules, and as more labelled anomalies are collected, it would be possible to implement a supervised machine learning classifier that would remove the need for manual rule definitions.

**Explanation of the operational workflow**
The workflow for setting up the monitoring of a newly installed system follows the steps below:

1. **Initial System Setup:** When a new system is installed, monitoring is not possible for the first seven days. Afterward, training can be initiated, tuning the normalizer based on the system's performance on clear sky days, and learning the relationship between the new system and its neighbours.
2. **Daily Monitoring Process:** Every day, after the daily production summary for the previous day is completed by the server, the algorithm can be run. It will use the previous day's production data from neighbouring systems to estimate what the monitored system should have produced. If the actual production significantly deviates from the estimate, an alarm is triggered for an operator to check the system's status and the cause of the possible anomaly.
3. **Regular Model Retraining:** Regular retraining of the model is not necessary unless the configuration of the target or neighbouring systems has changed, if the last training was based on limited amount of data, or if the model performs badly. If retraining is required, it is not fastidious, as the training time is about 500ms per system.

# 2 Acknowledgement

I would like to express my sincere gratitude to the company **PRiOT AG**, our partner in this research, for proposing this challenging problem and providing the necessary data for its execution. Their support and collaboration have been invaluable to the successful completion of this thesis.

I would also like to thank my **advisors and co-advisors**, whose guidance and expertise have been crucial throughout the duration of this thesis. Their insights and support have greatly contributed to shaping the direction of this research.

Additionally, I would like to acknowledge the use of artificial intelligence tools, including **DeepL and the GPT-4 language model**, which were utilized to assist in translation, grammar correction, and improving the syntax of the text.

# G. References

## 1 List of illustrations

# 2 List of tables

# 3 Bibliography

Adhya, D., Chatterjee, S., & Chakraborty, A. K. (2022). Performance assessment of selective machine learning techniques for improved PV array fault diagnosis. *Sustainable Energy, Grids and Networks*, *29*, 100582. https://doi.org/10.1016/j.segan.2021.100582

Adkins, P. C. (2023, July 7). The hidden validation set you aren't using …. *Data Science at Microsoft*. https://medium.com/data-science-at-microsoft/out-of-bag-validation-for-random-forests-378f2b292560

Bashir, N., Chen, D., Irwin, D., & Shenoy, P. (2019). Solar-TK: A Data-Driven Toolkit for Solar PV Performance Modeling and Forecasting. *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 456–466. https://doi.org/10.1109/MASS.2019.00060

Benninger, M., Hofmann, M., & Liebschner, M. (2019). Online Monitoring System for Photovoltaic Systems Using Anomaly Detection with Machine Learning. *NEIS 2019; Conference on Sustainable Energy Supply and Energy Storage Systems*, 1–6. https://ieeexplore.ieee.org/document/9000494

Brownlee, J. (2017, January 29). How to Decompose Time Series Data into Trend and Seasonality. *MachineLearningMastery.Com*. https://machinelearningmastery.com/decompose-time-series-data-trend-seasonality/

Chen, D., Breda, J., & Irwin, D. (2018). Staring at the sun: A physical black-box solar performance model. *Proceedings of the 5th Conference on Systems for Built Environments*, 53–62. https://doi.org/10.1145/3276774.3276782

Chen, D., & Irwin, D. (2017). SunDance: Black-box Behind-the-Meter Solar Disaggregation. *Proceedings of the Eighth International Conference on Future Energy Systems*, 45–55. https://doi.org/10.1145/3077839.3077848

Choudhary, D. (2021, April 18). Bootstrapping and OOB samples in Random Forests. *Analytics Vidhya*. https://medium.com/analytics-vidhya/bootstrapping-and-oob-samples-in-random-forests-6e083b6bc341

De Benedetti, M., Leonardi, F., Messina, F., Santoro, C., & Vasilakos, A. (2018). Anomaly detection and predictive maintenance for photovoltaic systems. *Neurocomputing*, *310*, 59–68. https://doi.org/10.1016/j.neucom.2018.05.017

De Soto, W., Klein, S. A., & Beckman, W. A. (2006). Improvement and validation of a model for photovoltaic array performance. *Solar Energy*, *80*(1), 78–88. https://doi.org/10.1016/j.solener.2005.06.010

Ding, Y., & Simonoff, J. S. (2010). An Investigation of Missing Data Methods for Classification Trees Applied to Binary Response Data. *Journal of Machine Learning Research*, *11*(1). https://www.jmlr.org/papers/volume11/ding10a/ding10a.pdf

Dobos, A. (2014). *PVWatts Version 5 Manual* (NREL/TP-6A20-62641, 1158421; p. NREL/TP-6A20-62641, 1158421). https://doi.org/10.2172/1158421

El-Banby, G. M., Moawad, N. M., Abouzalm, B. A., Abouzaid, W. F., & Ramadan, E. A. (2023). Photovoltaic system fault detection techniques: A review. *Neural Computing and Applications*, *35*(35), 24829–24842. https://doi.org/10.1007/s00521-023-09041-7

Elsheikh, A. H., Katekar, V. P., Muskens, O. L., Deshmukh, S. S., Elaziz, M. A., & Dabour, S. M. (2021). Utilization of LSTM neural network for water production forecasting of a stepped solar still with a corrugated absorber plate. *Process Safety and Environmental Protection*, *148*, 273–282. https://doi.org/10.1016/j.psep.2020.09.068

Elsheikh, A. H., Panchal, H., Ahmadein, M., Mosleh, A. O., Sadasivuni, K. K., & Alsaleh, N. A. (2021). Productivity forecasting of solar distiller integrated with evacuated tubes and external condenser using artificial intelligence model and moth-flame optimizer. *Case Studies in Thermal Engineering*, *28*, 101671. https://doi.org/10.1016/j.csite.2021.101671

Elsheikh, A. H., Sharshir, S. W., Abd Elaziz, M., Kabeel, A. E., Guilan, W., & Haiou, Z. (2019). Modeling of solar energy systems using artificial neural network: A comprehensive review. *Solar Energy*, *180*, 622–639. https://doi.org/10.1016/j.solener.2019.01.037

European Commission. (2024, February 14). *Photovoltaic Geographical Information System (PVGIS)*. https://joint-research-centre.ec.europa.eu/photovoltaic-geographical-information-system-pvgis_en

Facebook's Core Data Science team. (2023). *Prophet*. Prophet. http://facebook.github.io/prophet/

Feng, M., Bashir, N., Shenoy, P., Irwin, D., & Kosanovic, B. (2020). *SunDown: Model-driven Per-Panel Solar Anomaly Detection for Residential Arrays* (arXiv:2005.12181). arXiv. https://doi.org/10.48550/arXiv.2005.12181

Hong, Y.-Y., & Pula, R. A. (2022). Methods of photovoltaic fault detection and classification: A review. *Energy Reports*, *8*, 5898–5929. https://doi.org/10.1016/j.egyr.2022.04.043

Ibrahim, M., Alsheikh, A., Al-Hindawi, Q., Al-Dahidi, S., & ElMoaqet, H. (2020). Short-Time Wind Speed Forecast Using Artificial Learning-Based Algorithms. *Computational Intelligence and Neuroscience*, *2020*, e8439719. https://doi.org/10.1155/2020/8439719

*Imputation of missing values*. (2024). Scikit-Learn. https://scikit-learn/stable/modules/impute.html

Ineichen, P., & Perez, R. (2002). A new airmass independent formulation for the Linke turbidity coefficient. *Solar Energy*, *73*(3), 151–157. https://doi.org/10.1016/S0038-092X(02)00045-2

Iyengar, S., Lee, S., Sheldon, D., & Shenoy, P. (2018). SolarClique: Detecting Anomalies in Residential Solar Arrays. *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, 1–10. https://doi.org/10.1145/3209811.3209860

Iyengar, S., Sharma, N., Irwin, D., Shenoy, P., & Ramamritham, K. (2017). A Cloud-Based Black-Box Solar Predictor for Smart Homes. *ACM Transactions on Cyber-Physical Systems*, *1*(4), 21:1-21:24. https://doi.org/10.1145/3004056

Josef Perktold, Skipper Seabold, Kevin Sheppard, ChadFulton, Kerby Shedden, jbrockmendel, j-grana6, Peter Quackenbush, Vincent Arel-Bundock, Wes McKinney, Ian Langmore, Bart Baker, Ralf Gommers, yogabonito, s-scherrer, Yauhen Zhurko, Matthew Brett, Enrico Giampieri, yl565, … Yaroslav Halchenko. (2024). *statsmodels/statsmodels: Release 0.14.2* (Version v0.14.2) [Computer software]. Zenodo. https://doi.org/10.5281/ZENODO.593847

Koehrsen, W. (2018, January 10). *Hyperparameter Tuning the Random Forest in Python*. Medium. https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

Leo Breiman & Adele Cutler. (n.d.). *Random forests*. Retrieved August 24, 2024, from https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#ooberr

Li, B., Delpha, C., Diallo, D., & Migan-Dubois, A. (2021). Application of Artificial Neural Networks to photovoltaic fault detection and diagnosis: A review. *Renewable and Sustainable Energy Reviews*, *138*, 110512. https://doi.org/10.1016/j.rser.2020.110512

Long-Dong, B., Wu, Y.-K., & Pham, M.-H. (2021). Fault identification and diagnosis methods for photovoltaic system: A review. *2021 7th International Conference on Applied System Innovation (ICASI)*, 126–129. https://doi.org/10.1109/ICASI52993.2021.9568414

Mansurova, M. (2023, October 8). *Interpreting Random Forests*. Medium. https://towardsdatascience.com/interpreting-random-forests-638bca8b49ea

Mellit, A., Tina, G. M., & Kalogirou, S. A. (2018). Fault detection and diagnosis methods for photovoltaic systems: A review. *Renewable and Sustainable Energy Reviews*, *91*, 1–17. https://doi.org/10.1016/j.rser.2018.03.062

Mohan, R., Cheng, T., Gupta, A., & Garud, V. (2014). *Solar Energy Disaggregation using Whole-House Consumption Signals*. https://www.semanticscholar.org/paper/Solar-Energy-Disaggregation-using-Whole-House-Mohan-Cheng/edcfb4c8fde24e2d75b9da57f05c7ff2dbc07933

Perez, R., Ineichen, P., Moore, K., Kmiecik, M., Chain, C., George, R., & Vignola, F. (2002). A new operational model for satellite-derived irradiances: Description and validation. *Solar Energy*, *73*(5), 307–317. https://doi.org/10.1016/S0038-092X(02)00122-6

Pillai, D. S., & Rajasekar, N. (2018). A comprehensive review on protection challenges and fault diagnosis in PV systems. *Renewable and Sustainable Energy Reviews*, *91*, 18–40. https://doi.org/10.1016/j.rser.2018.03.082

Rapaport, S., & Green, M. (2021). *The Use of Advanced Algorithms in PV Failure Monitoring*. IEA-PVPS. https://iea-pvps.org/key-topics/the-use-of-advanced-algorithms-in-pv-failure-monitoring/

Schölkopf, B., Hogg, D. W., Wang, D., Foreman-Mackey, D., Janzing, D., Simon-Gabriel, C.-J., & Peters, J. (2016). Modeling confounding by half-sibling regression. *Proceedings of the National Academy of Sciences*, *113*(27), 7391–7398. https://doi.org/10.1073/pnas.1511656113

scikit-learn. (2022). *Permutation Importance vs Random Forest Feature Importance (MDI)*. Scikit-Learn. https://scikit-learn/stable/auto_examples/inspection/plot_permutation_importance.html

scikit-learn. (2023). *3.1. Cross-validation: Evaluating estimator performance*. Scikit-Learn. https://scikit-learn/stable/modules/cross_validation.html

Tabone, M., Kiliccote, S., & Kara, E. C. (2018). Disaggregating solar generation behind individual meters in real time. *Proceedings of the 5th Conference on Systems for Built Environments*, 43–52. https://doi.org/10.1145/3276774.3276776

Tina, G. M., Cosentino, F., & Ventura, C. (2016). Monitoring and Diagnostics of Photovoltaic Power Plants. In A. Sayigh (Ed.), *Renewable Energy in the Service of Mankind Vol II: Selected Topics from the World Renewable Energy Congress WREC 2014* (pp. 505–516). Springer International Publishing. https://doi.org/10.1007/978-3-319-18215-5_45

Triki-Lahiani, A., Bennani-Ben Abdelghani, A., & Slama-Belkhodja, I. (2018). Fault detection and monitoring systems for photovoltaic installations: A review. *Renewable and Sustainable Energy Reviews*, *82*, 2680–2692. https://doi.org/10.1016/j.rser.2017.09.101

Van Gompel, J., Spina, D., & Develder, C. (2023). Cost-effective fault diagnosis of nearby photovoltaic systems using graph neural networks. *Energy*, *266*, 126444. https://doi.org/10.1016/j.energy.2022.126444

Wager, S., Hastie, T., & Efron, B. (2014). *Confidence Intervals for Random Forests: The Jackknife and the Infinitesimal Jackknife* (arXiv:1311.4555). arXiv. https://doi.org/10.48550/arXiv.1311.4555

# H. Appendix

## 1 Project definition

*Please see a copy of the project definition attached at the end of this thesis.*

## 2 Code

*The code is directly available online in the following git repository, or by contacting its author :*

https://github.com/maximecharriere/TM_PV_ADC

## 3 List of duplicate date

*This is a confidential data is only accessible to advisors and the partner company.*

See file `Appendix/duplicate_dates.csv`

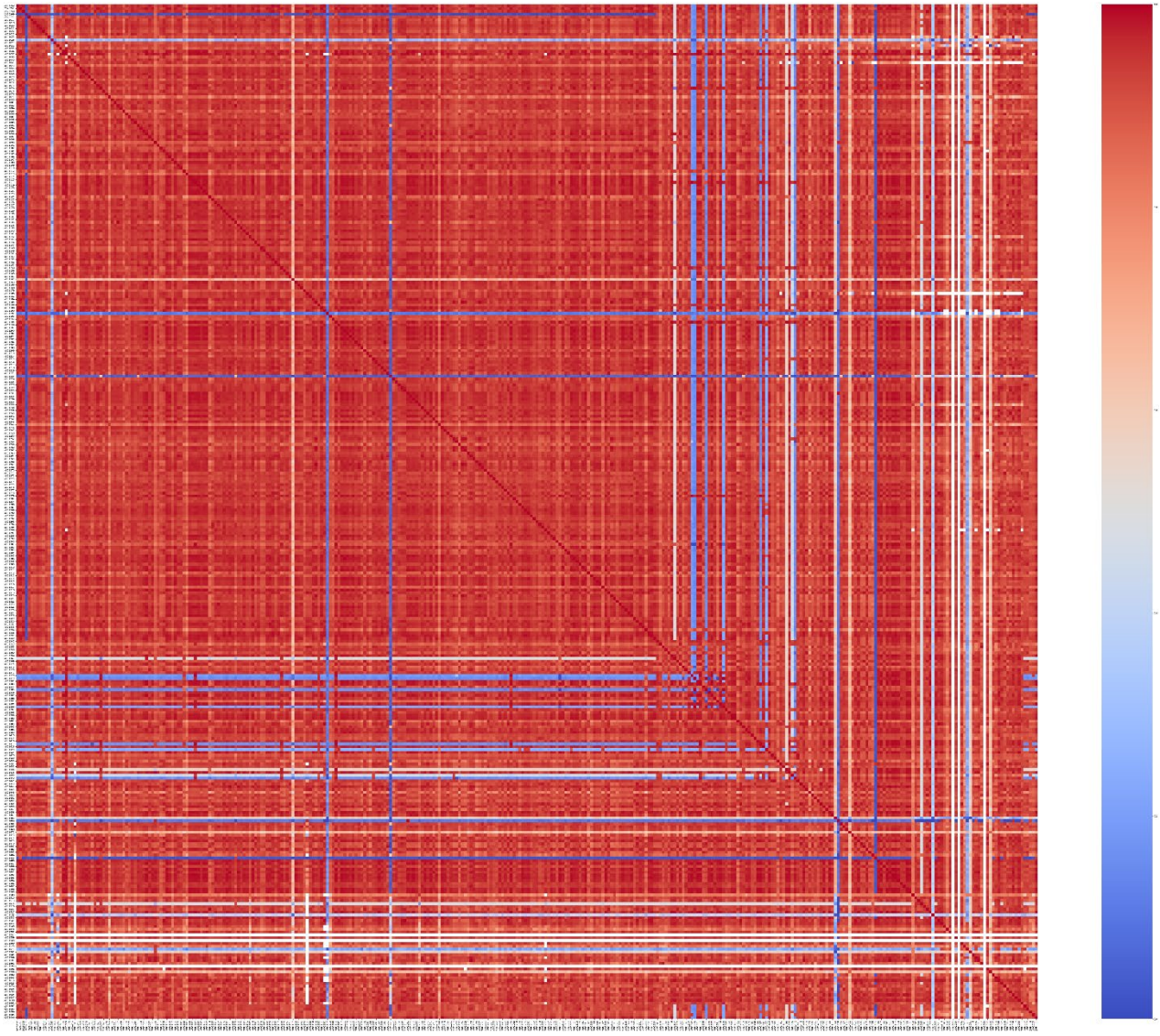
## 4 Filtered out PV Systems

*This is a confidential data is only accessible to advisors and the partner company.*

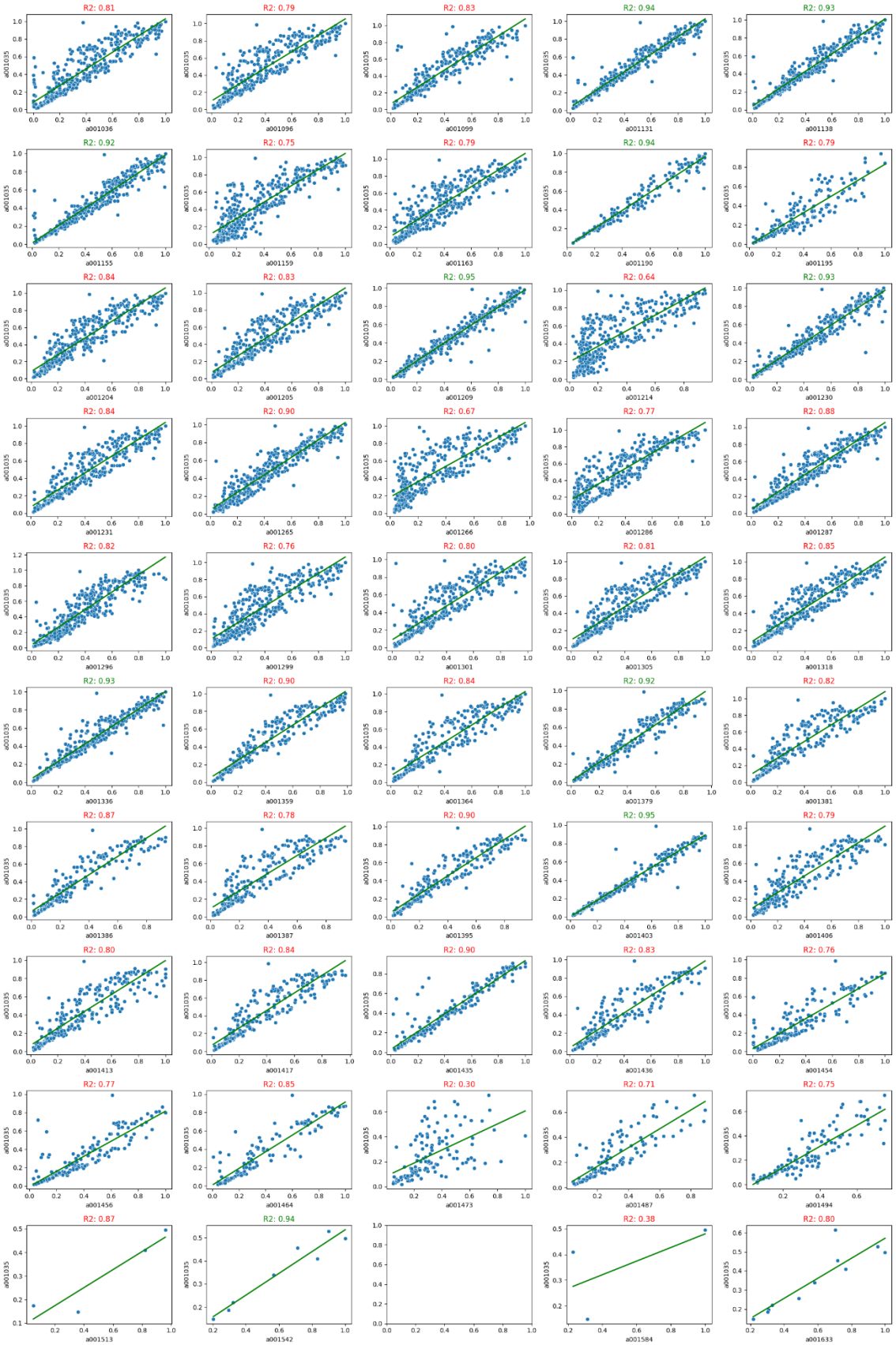
See files:

- `Appendix/erroneous_metadata.txt`
- `Appendix/not_enought_data.txt`
- `Appendix/unfitted_systems_by_normalizer.txt`

# 5 Pearson Correlation

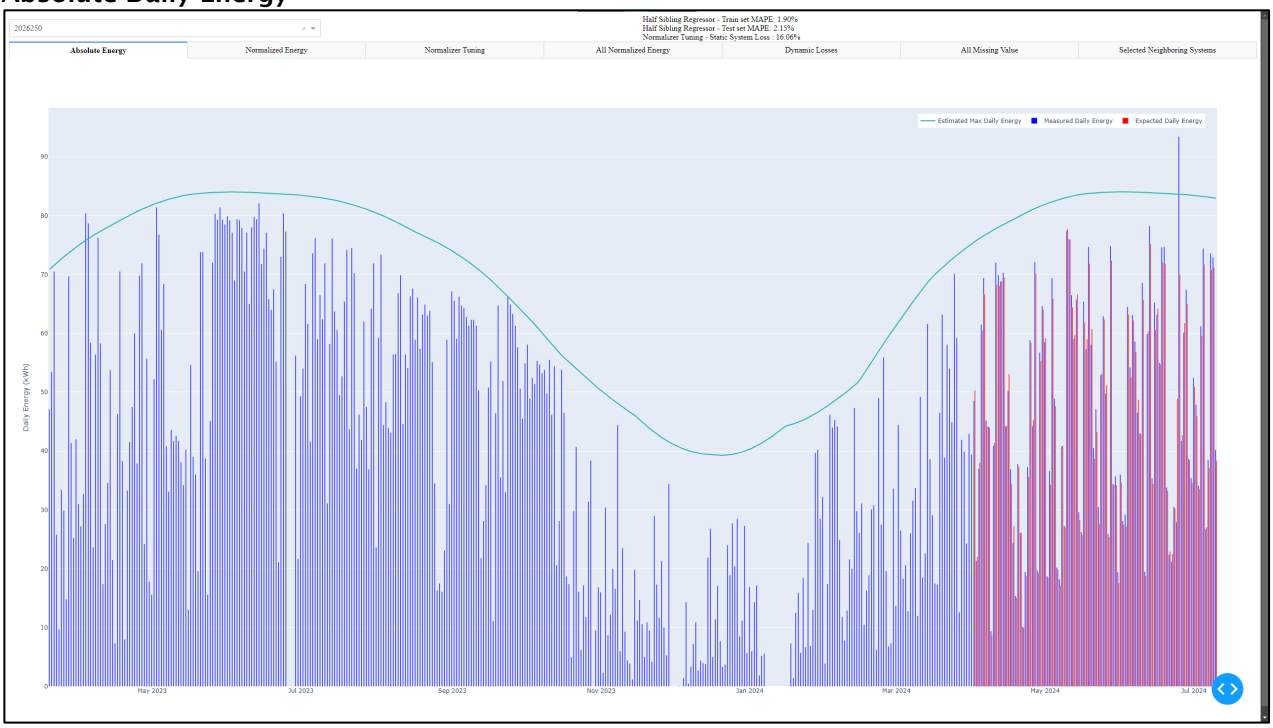


# 6 All Pairwise Linear Regression

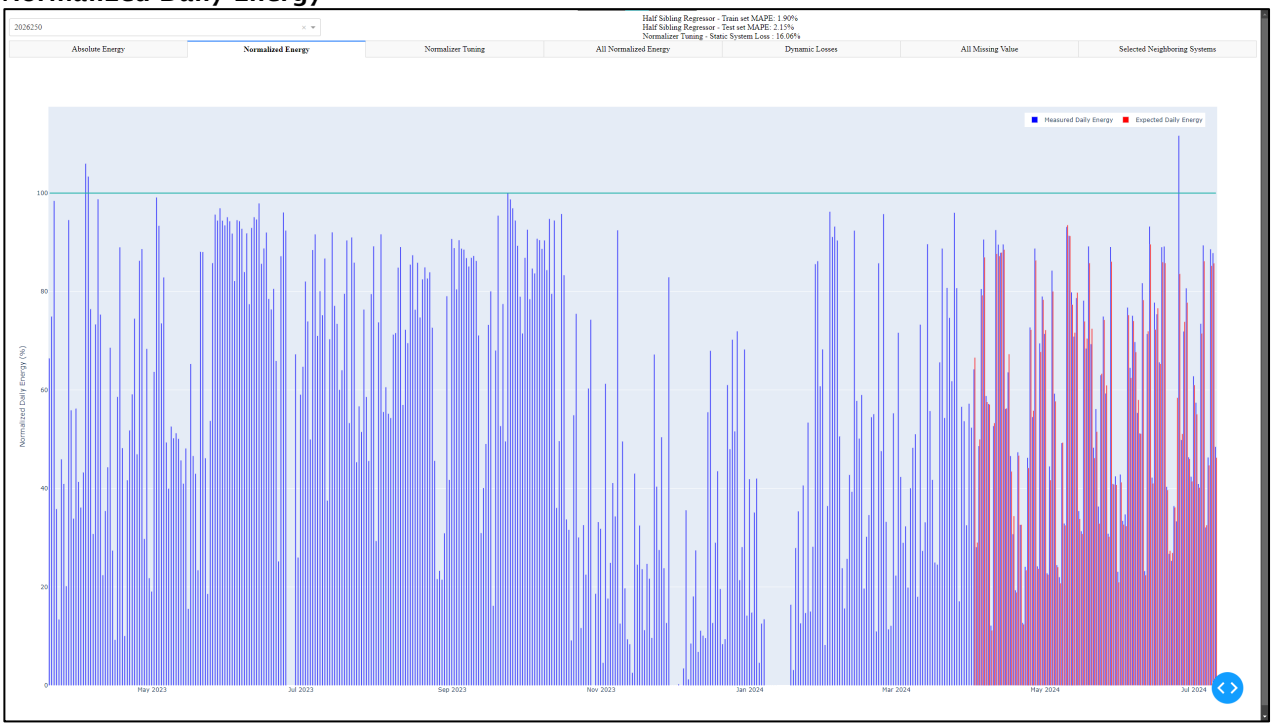


# 7 Application – User Interface Screenshots

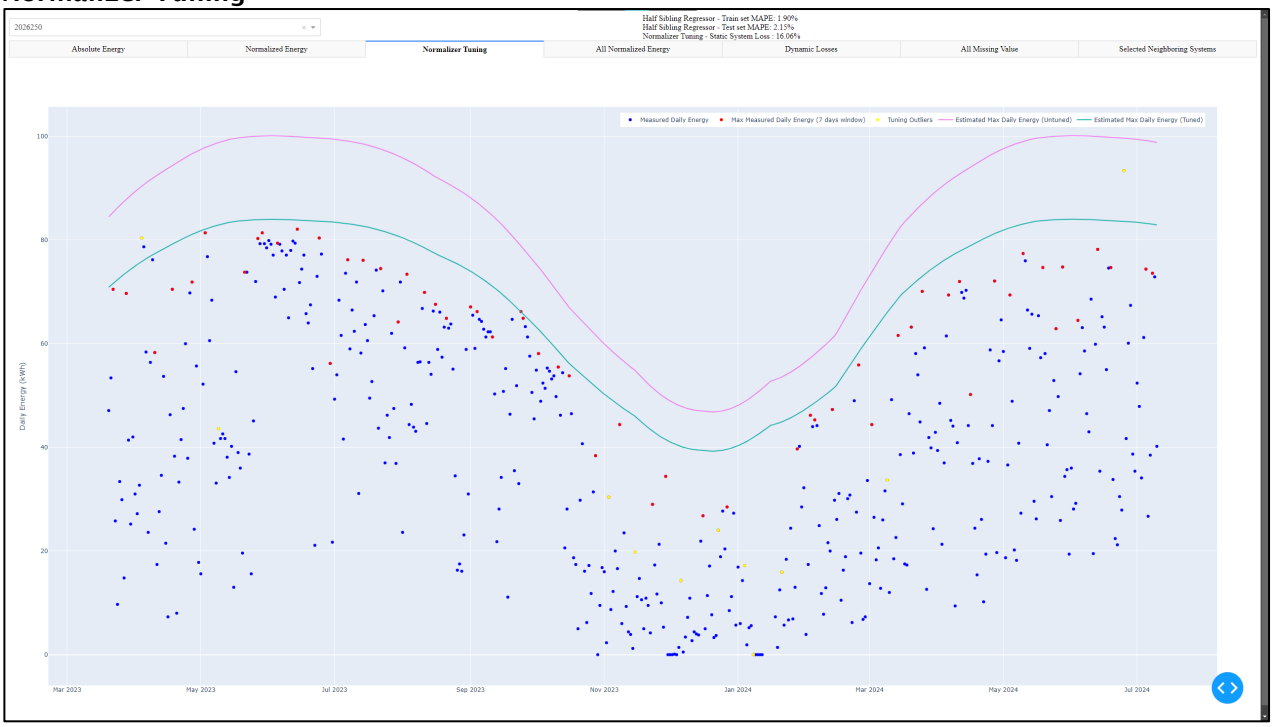
## Absolute Daily Energy



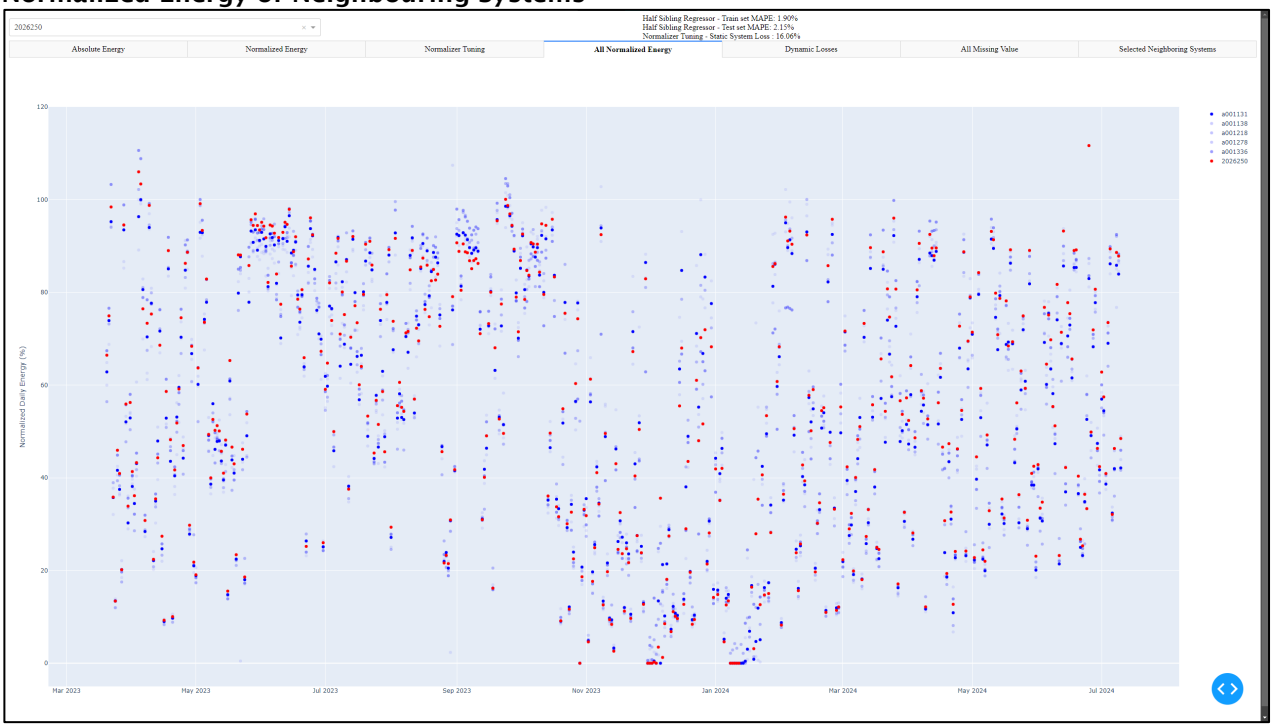

## Normalized Daily Energy



## Normalizer Tuning

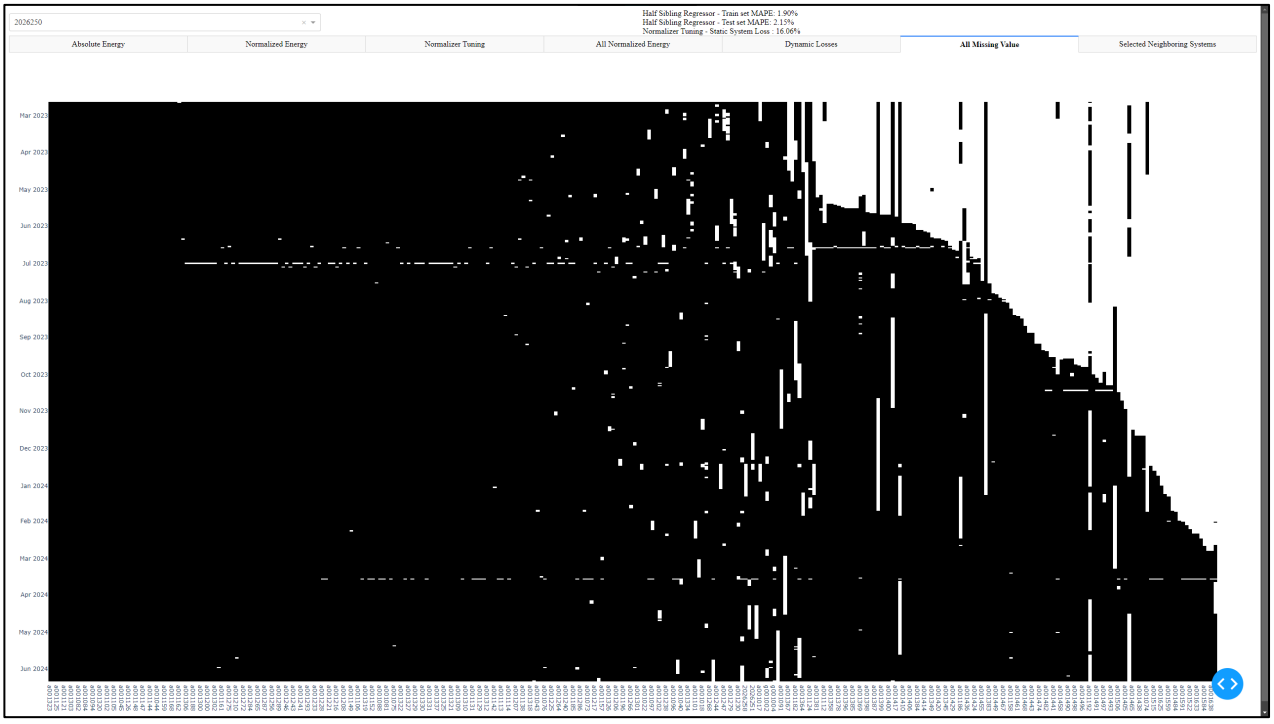



## Normalized Energy of Neighbouring Systems



## Underproduction, with detected anomalies


## Missing Values



## Selected Neighbouring Systems