Degree Project
Level: Master

FAULT DETECTION FOR SMALL-SCALE PHOTOVOLTAIC POWER INSTALLATIONS

A Case Study of a Residential Solar Power System

Author: Maxim Brüls
Supervisor: William Song
Examiner: Siril Yella
Subject/main field of study: Microdata Analysis
Course code: MI4002
Credits: 15 ECTS
Date of examination: January 20, 2021

At Dalarna University it is possible to publish the student thesis in full text in DiVA. The publishing is open access, which means the work will be freely accessible to read and download on the internet. This will significantly increase the dissemination and visibility of the student thesis.

Open access is becoming the standard route for spreading scientific and academic information on the internet. Dalarna University recommends that both researchers as well as students publish their work open access.

I give my/we give our consent for full text publishing (freely accessible on the internet, open access):

Yes ☒ No ☐
Abstract:

Fault detection for residential photovoltaic power systems is an often-ignored problem. This thesis introduces a novel method for detecting power losses due to faults in solar panel performance. Five years of data from a residential system in Dalarna, Sweden, was applied on a random forest regression to estimate power production. Estimated power was compared to true power to assess the performance of the power generating systems. By identifying trends in the difference and estimated power production, faults can be identified. The model is sufficiently competent to identify consistent energy losses of 10% or greater of the expected power output, while requiring only minimal modifications to existing power generating systems.

Keywords:

# TABLE OF CONTENTS

LIST OF TABLES .................................................................................................................. 3  
LIST OF FIGURES ................................................................................................................ 3  

CHAPTER I: INTRODUCTION ............................................................................................ 4  
0.1. Aim ................................................................................................................................ 4  
1.1. DEFINING CONCEPTS ................................................................................................. 5  
1.1.1. Fault Types and the Goal of a Fault Detection Model .............................................. 5  

CHAPTER II: LITERATURE REVIEW ............................................................................... 6  

CHAPTER III: DATA OUTLINE & EXPLORATION ......................................................... 8  
3.1. Core Variables ........................................................................................................... 9  

CHAPTER IV: THEORY .................................................................................................... 11  
4.1. CONCEPTUAL APPROACH .................................................................................... 11  
4.2. MODEL SELECTION ............................................................................................... 12  
4.2.1. Time Series Modelling ......................................................................................... 12  
4.2.2. Classification ........................................................................................................ 12  
4.2.3. Linear Multiple Regression .................................................................................. 13  

CHAPTER V: EMPIRICAL WORK ....................................................................................... 15  
5.1. DATA PREPROCESSING .......................................................................................... 15  
5.1.1. Time Interactions ............................................................................................... 15  
5.1.2. Solar Position Interactions .................................................................................. 15  
5.1.3. Further Processing .............................................................................................. 16  
5.2. EMPIRICAL MODEL ............................................................................................... 16  
5.2.1. Balance between Regression and Decision Tree ................................................ 16  
5.2.2. Empirical Model ................................................................................................. 17  
5.3. EMPIRICAL METHODOLOGY ............................................................................. 18  
5.3.1. Issues during Winter ......................................................................................... 18  
5.4. EMPIRICAL RESULTS ............................................................................................ 20  
5.4.1. The No-Fault Case ........................................................................................... 20  
5.4.2. Introducing Simulated Power Losses during September (3%, 10%, and 20%) ........... 22  
5.4.3. Nominal Energy Losses ..................................................................................... 25  
5.4.4. Relative Energy Losses ..................................................................................... 25  
5.4.5. Visualizing Nominal and Relative Energy Losses ............................................. 27  
5.4.6. A Real-World Fault ......................................................................................... 29  

CHAPTER VI: DISCUSSION ............................................................................................ 30  
6.1. Discussion of the Strengths and Weaknesses of the Model ...................................... 30  
6.2. Discussion of the Performance of the Model ............................................................ 30  
6.3. Future Improvements ............................................................................................. 31  

CHAPTER VII: CONCLUSION ......................................................................................... 32  
REFERENCES ................................................................................................................... 33  
Other ................................................................................................................................ 33
LIST OF TABLES

- TABLE I: Common Faults in PV Systems 5
- TABLE II: Raw Data Description 8

LIST OF FIGURES

- FIGURE 1: Relationship between Radiation and Power Production 10
- FIGURE 2: Relationship between Low-Level Radiation and Power Production 10
- FIGURE 3: Histogram of the Residual Values of the Testing Dataset 20
- FIGURE 4: Scatterplot of the Residual values of the Testing Dataset 21
- FIGURE 5: Residual Values of Two Identical Datasets: 3% Power Loss 22
- FIGURE 6: Residual Values of Two Identical Datasets: 10% Power Loss 23
- FIGURE 7: Residual Values of Two Identical Datasets: 20% Power Loss 23
- FIGURE 8: GIF (only works in Word) 24
- FIGURE 9: figure of Residuals Scaling Values 27
- FIGURE 10: Nominal Energy Losses – Simulated 28
- FIGURE 11: Relative Energy Losses – Simulated 28
- FIGURE 12: Relative Energy Losses – Real 29
- FIGURE 13: Nominal Energy Losses – Simulated 29
CHAPTER I: INTRODUCTION

Energy from the sun can be used to produced cheap, clean power. According to Euroobserver (2020), European Union total solar power production increased by a factor of sixty-five since 2005, from 2GW to 130GW.\(^1\) Solar energy has become more affordable, widespread, and efficient. As solar power is being adopted, researchers have found increasingly sophisticated ways to detect faults expediently to reduce power losses – a disconnected wire, or even a tree which grew to tall and wide, can reduce power output for a long time before it is detected.

As the scale of industrial photovoltaic (PV) systems grew, so has the accompanying suite of sensors to monitor these systems' performance. We have arrived at a point where fault detection has become quite advanced for large installations. Unfortunately, little attention has been paid to the small-scale residential installations, such as roof-based solar panels. For such systems, automatic fault detection – i.e., fault detection without the direct supervision of a human – is virtually nonexistent. The reason for this is that fault detection requires, so it is presumed, an expensive and sophisticated monitoring system which is not feasible for smaller systems that have a maximum power production capacity of not more than a few kilowatts. A careful study of the existing literature shows that there is a strong preference to create fault detection methods for large systems, while residential systems are ignored (see literature review).

The aim of this research is to create a proof-of-concept that a basic fault detection system for small-scale PV systems can be created without the need for expensive monitoring equipment, and to create and validate a method which can lead to a viable system for residential systems. This research can be beneficial for a host of agents: PV system owners can benefit from increased power output and a greater financial return, while sellers can boast more sophisticated technology. Society, of course, is better off with more efficient sources of clean energy.

The method is based on the observation that solar power output of residential systems can be modeled relatively accurately with minimal data input. Conceptually, it is possible to detect significant power losses due to system faults even with a model that cannot precisely predict the power production for any given situation, but that can make an accurate average prediction. All the model must do is to be consistent in its predictions – in other words, the model must not be highly accurate for any given instant, but the average error of the prediction must not change significantly for different circumstances. In technical terms, the model must produce residual values of the prediction with a constant mean and finite variance. Shifts in the mean error of the prediction then indicate suboptimal photovoltaic performance.

More intuitively it can be said that if the statistical model predicts inaccurately but over- and underpredicts power output of the PV installation at relatively equal rates, then the system is fully functional. However, if the model constantly overpredicts how much power is produced during any given time-period, then it can be assumed that the PV system is not working properly.

1.0.1. Aim

The problem is that little has been done for residential solar power systems. This is an innovative work – supported by referring to related work. The explicit aim of this thesis is to create a statistical model which:

- can reliably detect sizeable power losses due to performance issues in residential and small-scale photovoltaic power installations.
- can do so with minimal data input and requires as little modification as possible to such PV systems.

\(^1\) Comparing year-over-year peak production.
1.1. DEFINING CONCEPTS

This section offers a general introduction into the concept of a fault detection model, and the types of faults that can lead to power losses in PV arrays.

1.1.1. Fault Types and the Goal of a Fault Detection Model

The nature of existing fault detection systems is described in more detail in the literature review. For the purposes of this thesis a “fault detection model” is a model which detects, with some accuracy, any reduction in power output outside normal operation. “Normal operation” must be understood differently between a private, residential setting and a commercial setting. In commercial settings, shading due to buildings blocking the sun are considered “faults”. However, in a private setting there are usually not many placement options, typically somewhere on the southmost-oriented slope of the roof. Due to this, cyclical shading caused by fixed objects, such as houses, are features rather than faults of a residential solar system, and this model need not be able to detect them but take them into account instead.

Solar panels generate, very roughly, one half of their energy from visible light, and the other half from infrared light. Water droplets let most of the visible light through, but almost none of the infrared, leading to power losses. The same can be said of snow: under a thin layer of snow, a few flocks thick, power losses will be roughly half of total output. Turbidity, or the presence of tiny moisture droplets in the air, is a standard variable in PV simulations and affects performance. All these factors need to be incorporated, directly or indirectly, in a fault detection system.

Thus, for the purposes of this thesis, a fault detection model tries to detect power losses which are not inherently caused by the nature of the installations (such as cyclical shading, cold temperatures, rainfall, etcetera). These factors are considered to fall under “normal operation” and losses associated with them will not be detected by the model. Faults outside normal operation are, for example, disconnected wires or inverter malfunctions: mechanical failures which directly cause power losses.

More intuitively, if the fault can be remedied by a solar module maintenance crew (or the owner), then the fault should be detected. If the fault is caused by circumstances beyond human control (say, rainfall), then it will be ignored.

Table I provides a comprehensive, non-exhaustive list about types of different faults (partially based on Chine, et al., 2016).

### TABLE I: Common Faults in PV Systems

<table>
<thead>
<tr>
<th>Type of Fault</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module-based faults</td>
<td>Any fault within the module, most importantly open and short circuit faults.</td>
</tr>
<tr>
<td>Connection</td>
<td>Increased electrical resistance between PV modules, leading to power losses.</td>
</tr>
<tr>
<td>Partial shading</td>
<td>Variable shading conditions are, in the literature, often considered “faults”.</td>
</tr>
<tr>
<td>By-pass diode faults</td>
<td>Every model consists of a number of solar cells, small “squares”. When a panel is partially shaded, electrical resistance in the panel increases, leading to losses. By-pass diodes solve this by providing “shortcuts” for the power to flow between solar cells.</td>
</tr>
<tr>
<td>Inverter faults</td>
<td>Any faults at the inverter (DC-AC conversion).</td>
</tr>
<tr>
<td>Extreme Temperature</td>
<td>Solar panels are generally optimized to be maximally efficient around ~15° – 20°C. Departures from this range lead to sub-optimal behavior.</td>
</tr>
<tr>
<td>Precipitation</td>
<td>Even thin layers of snow can lead to dramatic decreases in power output. Waterdrops, too, decrease power output.</td>
</tr>
</tbody>
</table>
CHAPTER II: LITERATURE REVIEW

The literature on fault detection in photovoltaic power generating systems is quite extensive. Today, research has advanced enough to “solve” the problem: it is already possible to detect faults with a near-perfect degree of accuracy, as will become clear from this review. The problem lies not in the ability to detect faults, but to do so in a practical and economical manner. This literature review will show that the field has focused largely on large-scale PV plants where it is worthwhile to install sophisticated and expensive monitoring equipment. Small, residential installations, of which the only supervision are non-technical owners, are not considered in much of the literature, despite their vast contribution to the world’s energy needs.

The trend in the literature discussed in this section is that they tend to achieve high prediction accuracies – simpler models and more complex ones regularly achieve fault detection accuracies of ninety percent or more. This gives the misleading impression that there are many ways to easily detect faults in PV systems. High accuracies are due to the way research objectives are specified: the most complex models achieve high accuracy to detect and classify even the smallest faults, whereas the simplest ones might achieve a high accuracy for detecting the simplest faults. An example, discussed further in this section, is the paper by Platon (2015), who appear to employ simple regressions to detect faults. In general, the literature shows a clear tradeoff between accuracy and practicality.

Four papers are discussed below, in decreasing order of complexity, to provide a general overview.

One particularly effective, and complex, fault detection setup was devised by Chine (et al., 2016). They have devised a way to detect and identify eight different types of fault, which each have subcategories, using a combination of an algorithm and neural network. First, data is fed into the algorithm, which initially applies a set of conditions to identify more obvious faults in a manner not unlike a classification tree. If a fault is detected but cannot be directly identified by the tree, the data is passed through a neural network, which then classifies it with anywhere from 70% to 90% correct classification rates – an impressive result given that the set of faults they identify include faulty bypass diodes. Unfortunately, the model is plagued by complexity: it requires dedicated sensors for the temperature of every PV module, a radiation sensor, current and voltage input from the strings, as well as Simulink simulation for the current-voltage characteristics of the strings. I also suspect that situations of persistent shading of entire modules would jeopardize the model’s performance. In short, the model works, so long as one is willing to expend time, effort, and money to make it work. It is ideally suited for commercial application, but wholly inapplicable to a residential setup.

Garoudja (et al., 2017) employ a similar, though simpler, strategy. Like Chine, they use a neural network to classify faults, and they also rely on a simulation of the facility. Unlike Chine, its data inputs are limited to radiation, ambient temperature, and inverter current and voltage readers. Lacking as much data as Chine, the model is appropriately more limited to the detection of “obvious” faults such as disconnected strings – situations in which entire models are cut off from the circuit. While it is a capable model, it is included in this review to demonstrate the capabilities of fault detection models decrease swiftly as the data becomes more “general.”

---

2 A bypass diode is a part of the solar panel circuit which prevents voltage drops in cases of partial shading. In particular, it is a low-resistance element which allows current to “take a shortcut” if some cells in the solar panel are partially shaded and resistance has become too high in these areas for the current to flow. A faulty bypass diode is tricky to identify.

3 For example, a structure which casts a shadow over some panels, some of the time.

4 Data comes from strings of PV modules, rather than from every individual module.
The next work is more useful for residential PV setups. Gokmen (et al., 2013) attempt to detect open- or short-circuit faults\(^5\) under different shading conditions. It creates a simulation of PV modules and are then used to create the correct operating voltages given different levels of ambient temperature. Then, faults are simulated, and the voltage changes recorded. The results are stored in a table, which can be used to determine the percentage chance of a given fault occurring. For a given temperature and operating voltage, one can read from the table the probability that it is operating normally, that fault A has occurred, or fault B, et cetera. Remarkably, radiation is not included. The main advantage of this method is that it might be generalizable: presumably, one could compute such a table for any set of PV modules with little effort and a minimal number of sensors. The disadvantage is that for most voltage levels, including those expected under normal operation, the model will always provide some chance of a fault. It is not clear whether this problem is sufficiently large to render the model practically unusable, and the simplicity of the approach is valuable for any research into fault detection for small setups.

The last paper (Platon, et al., 2015) works with data from the AC side of the inverter. This is rather unusual, since there are no apparent advantages to using AC-side data over DC-side data. However, the value in this research lies in its simple, easy-to-replicate approach. They use panel temperature and radiation data to predict power output. They create a model and fit the coefficients – presumably in the form of a simple multivariate regression, though this is not explicitly stated. While they do require temperature sensors from every PV module, they do not, in contrast to the other papers here, use any simulations as a benchmark for module performance. Instead, the model is trained with historical data and then tested, meaning the fitted coefficients cannot be used elsewhere. Their model appears to be a good predictor of power output under normal operating circumstances, i.e., that deviations from the predicted output indicate a fault. Classification accuracy is often close to 80%, depending on the time of the year, and dips as low as 50%. While their model is simple and their accuracy, the simplicity considered, high, the model does not in any way entertain itself with fault identification. An important detail is that they obtain significantly better results with aggregated data – sampled at ten-minute intervals but averaged out over the hour.

It is worth noting that some of the literature aims to achieve something which, at least initially, goes beyond the requirements of residential systems – namely that they not only attempt to detect all faults, but that they also identify the specific type of faults which has occurred. For residential systems, detection of faults large enough to call in a PV technician suffices.

---

\(^5\) An open circuit suffers from zero electrical resistance, meaning no electrical potential (zero voltage) and therefore no power. A short circuit suffers from infinite resistance, meaning that regardless of electrical potential, the electrons cannot flow, and no power is generated.
CHAPTER III: DATA OUTLINE & EXPLORATION

This chapter details the data in raw format. The next section discusses the data pre-processing process and the final form of the data before it is used to create a model.

### TABLE II: RAW DATA DESCRIPTION

<table>
<thead>
<tr>
<th>General Statistics</th>
<th>Sample Size</th>
<th>373,083 observations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sample Date Range</td>
<td>October 2015 – December 2020 (not continuously)</td>
</tr>
<tr>
<td></td>
<td>Sampling Interval</td>
<td>300 seconds</td>
</tr>
<tr>
<td></td>
<td>Format</td>
<td>Time Series</td>
</tr>
</tbody>
</table>

**Original Features: Data Measured Directly from the PV Array**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Included in Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Datetime</td>
<td>YES</td>
</tr>
<tr>
<td>T11M8H</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>T4M8MID</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>T9M8LOW</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>T6M8D</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>T4M8F</td>
<td>Module Temperature</td>
<td>NO</td>
</tr>
<tr>
<td>T1M3H</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>T12M3MID</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>T10M3LOW</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>T7M3D</td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td>*DCPower1</td>
<td>Direct Current Power from System 1</td>
<td>YES</td>
</tr>
<tr>
<td>*DCPower2</td>
<td>Direct Current Power from System 2</td>
<td>YES</td>
</tr>
<tr>
<td>Is1</td>
<td>Voltage (mV) from the string connected to module 1</td>
<td>NO</td>
</tr>
<tr>
<td>Is2</td>
<td>Voltage (mV) from the string connected to module 2</td>
<td>NO</td>
</tr>
<tr>
<td>Is3</td>
<td>Voltage (mV) from the string connected to module 3</td>
<td>NO</td>
</tr>
<tr>
<td>Is4</td>
<td>Voltage (mV) from the string connected to module 4</td>
<td>NO</td>
</tr>
<tr>
<td>Is5</td>
<td>Voltage (mV) from the string connected to module 5</td>
<td>NO</td>
</tr>
<tr>
<td>Is6</td>
<td>Voltage (mV) from the string connected to module 6</td>
<td>NO</td>
</tr>
<tr>
<td>Is7</td>
<td>Voltage (mV) from the string connected to module 7</td>
<td>NO</td>
</tr>
<tr>
<td>Radiation</td>
<td>Radiation Measurement</td>
<td>YES</td>
</tr>
<tr>
<td>VWind</td>
<td>Wind Speed</td>
<td>NO</td>
</tr>
<tr>
<td>T1n</td>
<td>Inside Temperature (at inverter)</td>
<td>NO</td>
</tr>
<tr>
<td>Troom</td>
<td>Room Temperature (at inverter)</td>
<td>NO</td>
</tr>
<tr>
<td>Tamb</td>
<td>Ambient Outside Temperature</td>
<td>NO</td>
</tr>
<tr>
<td>AC Energy1</td>
<td>AC-side Energy for system 1</td>
<td>NO</td>
</tr>
<tr>
<td>AC Energy2</td>
<td>AC-side Energy for system 2</td>
<td>NO</td>
</tr>
</tbody>
</table>

**External Features: Data from other sources**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Source</th>
<th>Included in Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longitude</td>
<td>Coordinate</td>
<td>Google Maps</td>
<td>YES</td>
</tr>
<tr>
<td>Latitude</td>
<td>Coordinate</td>
<td>Google Maps</td>
<td>YES</td>
</tr>
<tr>
<td>Solar Elevation</td>
<td>Height over horizon</td>
<td>PV Lib (Python Package)</td>
<td>YES</td>
</tr>
<tr>
<td>Solar Azimuth</td>
<td>East-west angle</td>
<td>PV Lib (Python Package)</td>
<td>YES</td>
</tr>
<tr>
<td>UTC Time</td>
<td>Universal Time</td>
<td>Google Astral</td>
<td>NO</td>
</tr>
</tbody>
</table>

Note that many “included” variables may be included in a derived or modified shape. Longitude, latitude, elevation, azimuth, and time are not explicitly included in the model. See chapter V.

*Dependent Variable in the model*
Intermittent data from October 2015 to December 2020 was available, with occasional gaps of missing data that range from a week to almost an entire year. The sampling interval is five minutes. Altogether, there are have 373,083 observations to work with.

The data was gathered from two rooftop PV systems which are located close to downtown Borlänge (Dalarna County, Sweden) by Frank Fiedler, Senior Lecturer Energy and Environmental Technology at Dalarna University. Table I gathers the essential information about the data.

The occasional week- or month-long gap in the data is inconvenient but workable. As will become clear in chapter IV, the time-series aspect of the dataset is irrelevant, and variables which explicitly indicate the order of the observations are removed. The order of the observations is unimportant. Instead, it is important that all the different periods of the year (all months, weeks, and seasons) are present at least once or twice. For example, the missing values for December 2016 do not pose a significant issue, since the values of December 2017 may be used to train the model.

3.1. Core Variables

Two core variables require a visual demonstration: Radiation (a regressor) and Direct Current Power (the dependent variable), or DC Power, because their relationship is a key determinant in model choice and performance.

Radiation is a powerful predictor of DC Power: the greater the radiation, the greater the number of Watts generated. However, this this is an indirect relationship. The angle from which the radiation falls onto the solar panels determines its effect. Moreover, the radiation sensor (known as a pyranometer) is placed several meters from the modules. This leads to many situations where either the panels or the radiation sensor is shaded. The relationship between Radiation and DC Power is shown by figure 1. Under ideal circumstances, every dot will fall on the diagonal. Due to differential shading, suboptimal angles, temperatures differences, and a host of other factors, this relationship is distorted. On figure 1, for example, the observations located above the diagonal represent situations where the solar modules were shaded but the pyranometer was not. Conversely, observations below the diagonal represent the opposite situation.

Moreover, even without the shading differential, the relationship between radiation and power generation is not linear at low levels of radiation, as can be seen on figure 2. The reasons for this non-linearity are not immediately apparent, even to experts. Potential explanations are inadequate sensitivity in the measuring equipment, suboptimal behavior of the solar modules at low light levels, or technical defects.

It is important to highlight this relationship because many observations can be found in this small radiation window: nighttime, morning, evening, and wintertime observations have exceptionally low radiation levels. For the most part, these cannot simply be removed from the dataset. Because of this, it is difficult to fit a model which performs well for both low and high radiation values.

Moreover, radiation, is the key independent variable in the model. As such, it is important to understand the difficult relationship between it and the dependent, DC Power.
Figure 1: Relationship between Low-Level Radiation and Power generation.
CHAPTER IV: THEORY

The first part of the chapter explains the model in conceptual terms and provides some intuition. Thereafter follows an in-depth discussion about the choice of the statistical model and why the linear regression proved superior to other, more self-evident candidates.

4.1. CONCEPTUAL APPROACH

The aim is to build a model which estimates the power output of the PV array for a given number of parameters. The comparison between the predicted power output for a given observation \( i \), \( \hat{y}_i \), and the true power output, \( y_i \), permits inference regarding the performance of the PV array. For example, at radiation level \( x_{1i} \) and temperature \( x_{2i} \), the model predicts an estimated power output of \( \hat{y}_i \) watts. The predicted \( \hat{y}_i \) watts can then be compared to the actual power output \( y_i \). If the model perfectly simulates the performance of the PV array, then any difference between predicted and actual power output \( \hat{y}_i - y_i > 0 \) indicates some sort of malfunction, since the system is not producing as much power as it could, given the circumstances.

It is not possible to build such an accurate model with the limited data available. Since it is the aim of this thesis to build a model which can detect performance drops with limited data and sensors, any constructed model will generate significant errors \( \hat{y}_i - y_i \neq 0 \) in its predictions. This approach uses these errors, or residuals, to analyze the performance of the PV array.

Let us designate \( \hat{y}_i - y_i = u_i \), the residual value of the model prediction for a single observation. A model which predicts \( u_i = 0 \) for every observation is a perfect model. A model which is correct on average will suffice. In other words, create a model which yields a distribution of all \( u_i \) such that:

\[
\begin{align*}
  u_i &\sim N(\mu, \text{var}) \\
\end{align*}
\]

In which the mean \( \mu \) is constant for all \( i \). Note that no assumption is made regarding the variance. This is because the variance of the errors is likely to change depending on whether the predictions are made for the summer or winter – power generation is puny in December compared to June, and residuals reflect this trend. However, the variance should nonetheless be as small as possible, since greater variance makes it harder to uncover whether a deviation from the mean is due to random interference or a malfunctioning PV array. If the variance were zero, any deviation from the mean would indicate with certainty that a fault is present.

The following example illustrates the approach: if the residuals in June are, on average, 10 Watts, and just 2 Watts in December, then the variance may change because it has scaled with the amount of predicted power generated. This need not be an issue. What matters most is that the model predicts residuals around a constant mean (ideally, zero, but not necessarily so). However, since in June ten times more power is generated than in December, these average residuals give the misleading impression the PV array performed worse in June (10 Watts lost) than in December (2 Watts lost). The opposite is the case: relative to the amount of sunlight, the array in June lost half as much power as in December. This issue is discussed and remedied in Chapter V (Relative Energy Losses).

This approach allows for the identification of a power loss and its nominal size. Assume that the model has been correctly and consistently “predicting” power output such that:

\[
  u_{2020} \sim N(2 \text{ Watts}, \text{var}_{2020})
\]

However, in the first month of 2021, the model generates the following output:
\[ u_{2021-01} \sim N(15 \text{ Watts}, \text{var}_{2020-01}) \]

Under the assumption that the PV array has been functioning correctly for the entire duration of 2020, we can conclude that there is a malfunction which has led to an energy loss of \((15-2)\text{Watts} \times 24\text{h} \times 31\text{days} = 9.672 \text{kWh}\). This is the nominal power loss. Because the variance, presumably, changes throughout the year, a determination about the relative power loss cannot be made – i.e., a percentage loss.

Empirical and theoretical testing shows that the approach described in this section is viable. However, smaller power losses are hard to detect since there is some inherent inaccuracy in the model.

### 4.2. MODEL SELECTION

I considered three different approaches to model the Direct Current Power output of the PV array:

- Time Series Modelling
- Classification
- Linear Regression

This section outlines the different strengths and weaknesses of each option, and the reasons for selecting the linear regression with a small decision tree.

#### 4.2.1. Time Series Modelling

ARIMA, Prophet, and Exponential Smoothing (Holt-Winters) models were considered for the model. However, these were abandoned early on because of three reasons:

Time series models assume that there is value to be found in the order of the observations (serial correlation). An observation is influenced by the previous one, or by an observation two weeks ago, or the past three observations, and so on. This is extremely restrictive, and it means that any observation can only be determined by a limited number of fixed past observations, predetermined by the model fit. An in-depth discussion concerning this issue can be found in Greene’s *Econometric Analysis* (2012, Chapter 20). One can get around this by reducing the influence of the serial correlation and introduce external regressors, but it is then sometimes better to drop the time-series approach altogether. Moreover, the data contains gaps, which further hampers the use of time-series models. For time-series modeling, the order and closeness of observations matters. For data from PV arrays this assumption does not hold strongly: a cloud, which is an almost entirely random phenomenon, does not care about the order of observations. The same can be said for temperature, and other factors which influence the performance of solar panels. The data available is far less cyclical than one might assume – sometimes, the daytime is similar to nighttime, and at other times, power generation drops when solar elevation increases. The lack of apparent cyclical patterns hampers the use of time-series models.

Moreover, elaborate autoregressive models can take significant amounts of time to compute, especially if one uses such functions as grid_search (sklearn) to determine the ideal number of lags.

A time-series based approach was swiftly ruled out. However, it may be useful in places with fewer clouds and more cyclical weather patterns.

#### 4.2.2. Classification

A classification model can be used to create residuals by classifying an observation within a certain interval of values, and then take the average of this interval to calculate the residual. Although most used for qualitative variables, decision trees can be applied to regression problems as well (Gareth, et al, 2013, p. 303 – see Chapter 8 for a general discussion). This type of model has the potential to be a powerful predictor of PV array performance because it allows for logical relationships between features. For example:
A linear regression model with a linear variable denoting month [1:12] and hour [1:24] cannot establish this relationship without significant feature modifications, even though such features are important determinants of the power output of a PV array. A classification model based on a decision tree, on the other hand, can easily do this. Moreover, decision trees swiftly deal with the non-linear effects of a feature on the outcome variable. This is especially valuable since the most important feature, radiation, has such a non-linear effect.

Nonetheless, there are drawbacks. For one, there is some difficulty involved when classifying a linear outcome variable (DC Power, measured in Watts). The outcome variables need to be divided (binned) in intervals, which is a highly subjective task. Not only the size of the intervals needs to be determined, but also whether to make them equally wide across the entire range of the outcome variable. The classification model can be powerful, depending on how narrow or broad the classification intervals are. However, the precise determination has too much influence on the performance of the model: slight changes in the width of the interval can make large changes in predictive ability. The main reason for why a pure classification model was not created, was because it could not be guaranteed that this model can also be used on datasets taken from other PV arrays.

Moreover, there is an inherent difficulty in determining intervals. If a linear regression predicts 9 Watts, and the true value is 7 Watts, the residual is 2 Watts. A classification model would categorize the prediction as belonging to the [0:10] Watts interval, leading to a residual of -2 Watts (derived from the mean value of the interval, 5: 5 - 7 = -2). This may or may not be a problem, but the uncertainty does not motivate the use of this model. Making many, narrow intervals can solve this but that would require an enormous decision tree, which introduces overfitting problems.

Moreover, there is an inherent difficulty in determining intervals. If a linear regression predicts 9 Watts, and the true value is 7 Watts, the residual is 2 Watts. A classification model would categorize the prediction as belonging to the [0:10] Watts interval, leading to a residual of -2 Watts (derived from the mean value of the interval, 5: 5 - 7 = -2). This may or may not be a problem, but the uncertainty does not motivate the use of this model. Making many, narrow intervals can solve this but that would require an enormous decision tree, which introduces overfitting problems.

4.2.3. Linear Multiple Regression

The linear regression model is the workhorse of econometrics. For the purposes of modeling a solar panel, the linear regression suffers from two main weaknesses: it does not deal well with features that have a non-linear effect on the outcome, and it cannot create a logical relationship between the features themselves.

The strengths of the linear regression are that, by its inherent design, the mean of the squared residuals of the training set is 0 by definition\(^6\), which usually translates to a zero mean for the absolute residuals, and that the residuals are generally scattered around the mean (this depends on the distribution of the data, but generally holds). The zero mean assumption must be fulfilled since, in the least-squares case, the constraint against which the regression is solved is given by \(\sum_{i=1}^{n} u_i^2 = 0\). This guarantees that any fitted model will generate a useful set of residuals, at least on the training data. This in turn can be used to validate the consistency of the model on one or several test sets: if the distribution of the residuals is similar on several test sets, it can be assumed that the model works well over a range of different datasets. The linear regression and derived attributes are discussed in-depth in Greene (2012, Chapters 3, 4).

Secondly, linear regressions make numeric predictions: the residuals can be inferred directly from the difference between the predicted and the true value, thereby negating the problems caused by classifying the outcome in intervals.

Thirdly, the linear regression is flexible and can be intuitively interpreted: its performance can be greatly enhanced by extensive feature manipulation, it is computationally fast, and it deals well with large datasets and large sets of features.

Most importantly, some of its shortcomings can be worked around: logical relationships between variables can be constructed through interactions terms, non-linear effects can be somewhat mitigated by variable transformations – in effect, applying functions which transform the relationship between regressor and

\(^6\) When Ordinary Least Squares or Absolute Deviations are used as loss functions. These are most commonly used.
coefficient from linear to any desired form (the only restriction is that the relationship must be molded to fit the empirical equation, which can be mathematically tricky). Again, refer to Greene (2012, Chapter 7). Lastly, the linear regression is compatible with decision tree setups.

For these reasons, a regression model fitted at the end of a modest decision tree was developed: the decision tree takes care of some non-linear effect of \textit{radiation} on \textit{DC Power}, while interaction terms compensate for the logical relationships between other features.
CHAPTER V: EMPIRICAL WORK

This chapter details the preprocessing which was required to ensure that the linear regression can “interpret” the data. It details transformations of time and solar position variables made to ensure that the regression can correctly recognize relationships between regressors.

5.1. DATA PREPROCESSING

Data preprocessing has proven to be the greatest hurdle in constructing an adequately performant model. The precise shape of time and solar data variables have made the difference between a model which is hardly consistent and one which manages to predict yearly power generation with an accuracy of about one percent. For this reason, the preprocessing will be discussed in detail. Two categories of linear variables were identified which required significant modification before they could be used in the linear regression:

- Time
- Position of the sun

Note that the precise form of the interaction terms was guided not only by statistical concerns (like how many additional features the sample size warrants), but also by performance. Overall, roughly a thousand separate dummy variables were added to the dataset. Statistically, this could be doubled or tripled without jeopardizing the performance of the linear regression, given the sample size, but this leads to severe memory issues. For these reasons, the size of the dataset was limited to around 2.5 GB.

5.1.1. Time Interactions

Time is an important feature because it incorporates obvious cyclical effects. For example, one of the two PV arrays that generated the data is shaded by a house at certain times. These times change over the course of a year because the position of the sun changes. The panels may be shaded at 9 AM in November, but not at 9 AM in June.

The original time variable in the dataset consists of a datetime format variable, accurate to the minute. Self-evidently, this needs to be converted to some numeric value. Dummy variables for year, month of the year, and hour of the day were created. The year dummy is included for data-selection purposes, but not for model estimation. Dummies alone, however, are not enough. At 9 AM in November the solar panels generate almost no power, but at 9 AM in June they do. This relationship can be created by introducing interaction terms between month and hour dummies.

This creates a set of $24 \times 12 = 288$ interaction terms, each in the form of a dummy. Any individual observation has a value of 1 for one of these dummies, and a value of 0 for the others. Since at 9 AM in November some solar panels are shaded, the corresponding interaction ($\text{INT}_\text{month}_\text{11}_\text{hour}_9$) terms reflects this by penalizing the predicted power output for these observations.

These time-based interactions have proven to be quite powerful tools to mitigate the statistical difficulties introduced by cyclical shading.

5.1.2. Solar Position Interactions

The position of the sun is included for much the same reason that the time interactions are included. However, simply inserting the linear features solar elevation (north-south angle) and solar azimuth (east-west angle) does little to improve the model. Solar elevation is, counterintuitively, not strongly related to the power output of the PV array. For high power output, the sun must be as perpendicular as possible to the plane of the PV array. A regression model cannot “understand” this relationship between the solar elevation
and the solar azimuth. Again, variables which explicitly state this relationship can be introduced by creating a large set of interaction terms.

This is best thought of as dividing the sky in small regions, the boundaries of which are determined by some arbitrary number of degrees of arc given by solar elevation and solar azimuth. Each square is represented by a dummy variable. At any given moment (i.e., for any given observation), the dummy of the square containing the sun will have a value of 1, all the other dummies a value of 0.

There is no perfect way to determine the size of these squares: as many as the data and hardware allows. Ideally, these squares would measure half a degree on each side—the rough angular size of the sun—but this results in approximately 130,000 interaction terms. Another option would be to organize the squares such that maximal variation in the data is allowed, i.e., by creating smaller regions in areas where the sun passes more often. This, however, requires a good understanding of orbital mechanics and constitutes a complicated affair from the data-preprocessing point of view.

The solution was to limit to regions to only those areas of the sky where the sun ever passes in Borlänge, roughly between \([0°:55°]\) elevation and \([-150°: +150°]\) azimuth. The size of each area is 5° elevation by 5° azimuth, resulting in \(\sim 750\) interaction terms. This is good enough to identify the position of the sun, and just barely within hardware capabilities.

### 5.1.3. Further Processing

Two other variables have been slightly reworked as well: **DCPower** (the measure of Direct Current Power from the panels, in Watts), and **Radiation**. For **DCPower**, all values smaller than 10 were set equal to zero, mainly to avoid the issues stemming from the non-linear relationship between power production and radiation during the dimmer moments of the day. **Radiation** is included twice: once in logged form, and once in its original form.

Linear variables denoting the **hour of the day**, **week of the year**, **day of the year**, and **month of the year** were also included to give the decision tree some additional splitting options. A randomized ID and universal time variable were also included, mostly for selection and dataset matching purposes.

Finally, basic preprocessing included the replacement of missing values by zero (the measuring equipment registers a missing value where it should measure zero), radiation values below 10 were set to 0, variables were converted, where possible, to integers to free up memory, and converting any time-based observations to UTC time for easier matching with external data.

### 5.2. EMPIRICAL MODEL

The statistical model consists of an expansive linear regression, containing over 1000 features, and a “modest” decision tree. First, the balance between the decision tree and regression model will be discussed. Subsequently, the precise form of both will be laid out. Details on decision trees are found in *Elements of Statistical Learning* (Hastie, et al, 2009, chapter 15.3).

#### 5.2.1. Balance between Regression and Decision Tree

A discussion on the choice of a “large” regression model atop a “modest” decision tree, and not a vast decision tree with a small regression model, is merited here as it has strong implications for the form of the model.

Most importantly, the linear regression easily handles large numbers of features. A decision structure would have to create huge trees with many nodes to achieve the same level of detail as the interaction terms. This can lead to overfitting issues, which also became apparent in preliminary tests. Secondly, the interaction
terms force the regression model to estimate the effect of all conceivable situations: any solar position, and the influence of time, and any level of radiation. This creates a certain “stability” in the predictions: the model is wrong often, but it is rarely very wrong. Thirdly, the only culprit of nonlinearity, the regression’s main weakness, is the *radiation* variable, but the dummies and interaction terms manage to compensate for this.

Put briefly, the regression works well overall, and its few flaws can be compensated for by a decision tree, but, conversely, a regression cannot compensate for the overfitting risk of a decision tree. Therefore, the decision tree segments the data to ensure good fits, after which the regression handles the bulk of the predictive effort.

### 5.2.2. Empirical Model

The fitted regression model is represented by the following equation. A generalized matrix version can be found in *Econometric Analysis* (Greene, 2012, p. 66-67). The most important element is the residual, \( u_i \):

\[
DC\ Power_i = \hat{\alpha} + \hat{\beta}_1 \cdot \text{Radiation}_i + \hat{\beta}_2 \cdot \log(\text{Radiation}_i) + \overline{D} \cdot \text{Sun} + \sum \hat{\gamma}_j \cdot \text{Linear Time}_j + \\
+ \sum \hat{\delta}_j \cdot \text{Time Interactions}_j + \sum \hat{\theta}_j \cdot \text{Sun Position Interactions}_j + \hat{\tau} \cdot \text{Random ID}_i + u_i
\]

\( i \) refers to a moment in time, which in turn identifies a single observation. The coefficients \( \hat{\beta}_j \) incorporate the effect of radiation, \( \overline{D} \) indicates whether the sun is above or below the horizon, \( \hat{\gamma}_j \) inserts a number of linear time variables (*week of the year, month of the year, hour of the day, and day of the year*) to give the decision tree some additional options for splitting, \( \hat{\delta}_j \) comprises the coefficients of the time interaction terms, \( \hat{\theta}_j \) comprises the coefficients of the solar position interaction terms, \( \hat{\tau} \) has no statistical significance or impact on the model as *random ID* is a random identifier (it is included to be able to identify specific observations throughout the modelling process), and \( u_i \) is the residual value (alternatively written \( \hat{\varepsilon}_i \), or estimated error term). The constant \( \hat{\alpha} \) creates a zero mean for the residuals \( u_i \), which is useful for the intuitive interpretation of the residual values of a testing set.

The predicted power output is given by:

\[
\hat{\gamma}_i = DC\ Power_i - u_i
\]

This regression is estimated using the *decisiontree regressor* function from sci-kit learn. For regression problems it is recommended to allow all features (or a large proportion) to be considered for splitting nodes, but an element of randomness is preferred to ensure that the tree is robust, so the number of random features it can consider at any node is 5. The maximum depth is 25. This creates a truly random forest which provides consistent results. Higher values for both parameters create more accurate trees, at the cost of performance and overfitting risks. An inspection of the most important features reveals that both radiation features account for approximately 90% of node impurity reduction. At first sight, this seems strange since most nodes have only a tiny chance of being split on any of the two *radiation* variables.

This makes sense intuitively: except for the *radiation* and *linear time* parameters, the tree can really only split on interaction terms. These interaction terms take the value of 1 for a few hundred observations and the value of 0 for the remainder. Every time the tree splits on a time interaction, for instance, the side of the node where the time interaction has the value of 1 can no longer be split further for any time interaction. The side of the node where the split is 0 is limited by the depth of the three (25 splits). The same goes for the solar position interaction terms. As a result, the radiation parameters show up far more often than the settings of the tree would suggest, since at least one side of most nodes eliminates several hundred interaction terms.

Lastly, the decision tree can create nodes based on linear time variables. These were included to separate periods that may be especially hard to fit the model on, like winter, but do not contribute much.

Overall, this setup returns stable results across different samples, on both PV systems.
5.3. EMPIRICAL METHODOLOGY

This section discusses the choice of training and testing data, issues with modelling the winter periods, and the empirical procedure.

The model is trained on data from the years 2016, 2017, and 2019. Because the data contains many gaps, these three combined years of gathered data corresponds, roughly, to a continuous two-year dataset. Data from the year 2018 is used as a test set, since this is the single most complete year in the data, including the longest continuous stretch of uninterrupted data gathering. In total, the data contained in the training set corresponds to 645 full days of data.

Despite this, the training dataset remains unbalanced. For some months (e.g., February), there is substantially more data available than for others (e.g., July). Moreover, nighttime observations do not contain any useful information to train the model – once the sun is down, power generation drops to 0. The model understands this. The distribution of information containing observations across different months can be seen in the last column of table III. The test dataset contains the data from the year 2018, for a total of 290 days or 112,527 observations. The remaining 75 days are missing, mostly in March and June.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>18124</td>
<td>7.24%</td>
<td>71%</td>
</tr>
<tr>
<td>February</td>
<td>26064</td>
<td>10.41%</td>
<td>65%</td>
</tr>
<tr>
<td>March</td>
<td>24488</td>
<td>9.78%</td>
<td>51%</td>
</tr>
<tr>
<td>April</td>
<td>27133</td>
<td>10.83%</td>
<td>42%</td>
</tr>
<tr>
<td>May</td>
<td>23888</td>
<td>9.54%</td>
<td>29%</td>
</tr>
<tr>
<td>June</td>
<td>21516</td>
<td>8.59%</td>
<td>20%</td>
</tr>
<tr>
<td>July</td>
<td>13401</td>
<td>5.35%</td>
<td>22%</td>
</tr>
<tr>
<td>August</td>
<td>22425</td>
<td>8.95%</td>
<td>33%</td>
</tr>
<tr>
<td>September</td>
<td>24941</td>
<td>9.96%</td>
<td>51%</td>
</tr>
<tr>
<td>October</td>
<td>17562</td>
<td>7.01%</td>
<td>56%</td>
</tr>
<tr>
<td>November</td>
<td>17116</td>
<td>6.83%</td>
<td>66%</td>
</tr>
<tr>
<td>December</td>
<td>13796</td>
<td>5.51%</td>
<td>81%</td>
</tr>
<tr>
<td>Totals</td>
<td>250454</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

5.3.1. Issues during Winter

Winter months are especially hard for the model to deal with, for the following reasons:

1. Few daytime observations: In December, for example, the sun rises above the horizon only 19% of the time. This renders the remaining 81% useless, since nighttime observations do not contain any information which can be used to train the model.

2. Low radiation levels: the non-linear relationship between power output and radiation at low levels makes the model less accurate.

3. Snow: the panels may or may not be covered by snow. There is no explicit feature in the data pertaining to snowfall, and it is thus impossible to know with certainty whether an observation was taken from a snow-covered array.

These problems need not be overstated. The lack of sunlight means that truly little power is generated during winter, and detecting faults is therefore far less important. In the dataset, the power generated during the winter months (December, January, and February) is only 10% that of the power generated during the summer months (June, July, and August). In other words, it is more important for the model to detect an 11% power loss in summer than to detect an 89% power loss in winter. In addition, these “winter problems”
are only severe in the far North. At Southern latitudes, they resolve themselves due to the way the data changes.

Moreover, it will be shown that the model is more than capable of detecting a 25% power loss in November, indicating that it performs adequately even during the darker months.
5.4. EMPIRICAL RESULTS

First, the residuals from a no-fault system are analyzed. Subsequently, faults are introduced and compared against these original residuals.

5.4.1. The No-Fault Case

The model has been trained, and the prediction on the test dataset have been made. Figure 3 shows the distribution of the residuals $u_i$. For the purposes of visualization, the distribution does not show the true residual but, for any given observation, the average of the residuals of the nearest 72 observations timewise, with descending weights the farther the residual is from the observation (a triangular rolling window). This makes the figure “shorter” and ensures that the distribution of the data is visually discernible. Figure 3 shows that for most predictions the model is perfectly accurate. This has little to do with the performance of the model per se since even a bad model predicts nighttime observations perfectly. Most importantly, the non-zero residuals are relatively symmetrical, indicating that over the course of 2018 the model has overpredicted as much as it has underpredicted. Moreover, it is encouraging that the model never predicts a negative power production (this is not shown on the graph), although nothing inhibits it from doing so.\footnote{Regression models ignore "logical" values for their predictions. After predicting power output on the test data there were, somewhat surprisingly, no negative values for $\hat{y}_i$. This is, presumably, a consequence of the decision tree.}

![Figure 2: Histogram of the residual values $u_i$ of the testing dataset.](image_url)
The residuals can also be plotted against a time series, to visualize how the model performs at various moments throughout the year. This is shown in figure 4 (without rolling window modification), which contains 112,527 observations. As can be seen, the residuals are more-or-less evenly distributed around 0. The winter periods, toward the left and right of the graph, are an exception, presumably for the reasons discussed in chapter V. However, at various times during winter, the residual values are either zero or positive. This is a strong indicator that the PV array was at least partially covered by snow: if it were, it would produce no power when the model expects it to produce at least some power. The result is a set of positive residuals, which can be seen on figure 4 during the months of January, February, and December. If the solar panels were, in effect, covered by snow during these months, then the model has just uncovered a “fault” in the PV array. Nonetheless, this is speculative, since there is no data indicating whether the panels were snow-covered or not.

Besides the winter months, the residuals behave as discussed in chapter IV: they appear to have a constant mean of zero across the year and are packed closely together around the mean (limited variance).

Figure 3: Scatterplot of the residual values $u_i$ of the testing dataset

---

8 The average temperature during these months is roughly, -5°C. Moreover, the model will virtually always predict some power output during the day, whether snow is present or not, since the Radiation variable carries significant weight in the estimations.
While the model overestimates power production in some weeks and underestimates it in others, its average performance is quite good: it correctly predicted an average power output of 228 Watts, whereas the real average power output was 226 Watts (a difference of 0.7%).

5.4.2. Introducing Simulated Power Losses during September (3%, 10%, and 20%)

In this instance, the dependent variable, DCPower, was reduced by three percent during the month of September in 2018. September was chosen because it is one of the more moderate months, with relatively average temperatures and radiation levels. The same pattern demonstrated below is found for any period during which one introduces a simulated fault in the system. Figure 5 plots two set of residuals: those derived from predicting power output on a fully functioning system (grey), and those derived from a power output predicted on the same system, but with a 3% power loss in September (red). In all months except September, the residuals are precisely the same (they are offset a little for visualization purposes). In September, the red residuals are significantly more positive than the grey ones, indicating that the model has overestimated the power production – which is consistent with the presence of a fault. Figures 6 and 7 demonstrate the change of residuals at power losses of, respectively, 10% and 20%.

Figure 4: Residual values ui of two identical testing datasets: one without power losses (grey), and one with power losses (red, 3% loss in September).
Clearly, as the power losses increase in magnitude, a distinct pattern reveals itself: the residuals become increasingly positive. For power losses over 10%, visual inspection can swiftly identify a loss.

In this case, however, losses may be swiftly identified because we have access to the performance of the system without fault (the gray data) is available. In reality, there is no knowledge about whether a fault has occurred, and only the data provided by the array (the red data) can be used. In this case, identifying smaller faults, such as a 3% power loss, becomes impossible. Over the course of an entire year, the model has proven to be accurate to within a single percent, but for any individual week or month, let alone a single day, this does not hold. The shorter the duration of the fault, the larger the fault needs to be to be reliably detected.

So far, it has been shown that clear patterns exist when introducing simulated power losses. The next section describes a method for simplifying and clarifying the data.

The GIF on the next page9 (figure 8) demonstrates the change in residuals as the size of the fault increases.

However, such figures are neither clear nor intuitive. Moreover, the residuals measure power, not energy. For practical purposes, energy is preferred.

---

9 The GIF only works in Microsoft Word, not in PDF.
Figure 7: GIF of the residual values $u_i$ of two identical testing datasets: one without power losses (blue), and one with power losses (red, 1, 5, 10, 20, and 30% loss in September).

Period of Malfunction = SEPTEMBER

Size of Malfunction = 1% POWER LOSS

Malfunctioning System

Operational System
5.4.3. Nominal Energy Losses

The accuracy of the model decreases with the timescale of prediction: the average yearly prediction is highly accurate, but the average daily prediction is not. This means that large losses of energy will be noticed, even if momentary power is not accurately predicted.

Power cannot be used to measure how much production was lost – a conversion from power to energy is required. Energy is defined as the product of power and time. On any given set of power data, the energy loss between two moments, \( i \) and \( j \), can be derived trivially from the SI definition of Watts:\(^\text{10}\)

\[
TotalEnergyLoss = \int_i^j \text{Power Loss} \; dt
\]

Given the dataset, \( i \) is merely the date and time of an observation, while \( dt \) is the five minutes surrounding time \( i \). The model predicts how much power shall be generated, \( \hat{y}_i \), and compares it to the actual power generated, \( y_i \), to compute the residual, \( u_i \). Equation (4) is the empirical counterpart to equation (3), modified because the sampling interval is not infinitesimal, and expressed in Watt-hours of energy:

\[
TotalEnergyLoss = \sum_{i}^{j} \frac{\hat{y}_i - y_i}{12} = \sum_{i}^{j} \frac{u_i}{12}
\]

Where \( i \) and \( j \) are any two observations in the dataset between which the total energy loss is to be determined. The denominator is necessary for the correct conversion to Watt-hours (every observation contains five minutes, or one twelfth of an hour), and is the practical equivalent of \( dt \). Equation (4) is used to create figures 10 and 12.

5.4.4. Relative Energy Losses

Developing a graphical way to clearly identify a fault has proven challenging. Because energy production in summer is significantly higher than in winter, a plot which shows residual values is rather confusing: a 30% percent power reduction in winter is, in terms of power lost, equivalent to a 3% reduction in summer because roughly ten times more energy is produced. Thus, unmodified, it is not possible to visually distinguish a large wintertime power reduction from a small summertime reduction (for more intuition, compare figures 10 and 11, or 12 and 13), which in turn makes it difficult, if not impossible, to assess the true performance of the PV array. In addition, presenting percentual energy losses is impossible because the dimension change from nominal to relative inhibits the use of any smoothing techniques – a power reduction in the morning would receive the same weight as one during noon in the smoothing process, which is nonsensical. Without smoothing techniques, clear visualization of energy losses becomes a near impossible task. This section deals with the scaling of residuals such that performance in winter and summer can be compared in a straightforward manner, and smoothing techniques can still be applied.

To present a clearer picture the nominal values must be presented but scaled to account for the different levels of power production in summer and winter. This creates problems of its own: the precise power production in winter is unknown. The predicted (fitted) power output, \( \hat{y}_i \), was computed, and there is data on the actual power output, \( y_i \). These are, however, susceptible to weather changes and significantly change year-over-year on anything smaller than a monthly scale. Moreover, \( y_i \) cannot be used since, again, it is unknown whether this data includes a fault – trying to scale data for the purposes of identifying a fault based on data which itself may contain the fault, is counterproductive. Scaling the residuals based on the power production in the training data is also a biased process: the training set may contain an exceptionally bright winter, or a dim summer, or other anomalous weather.

The alternative way is to scale the residual values with the amount of daytime – i.e., the percent of the time that the sun was over the horizon at any given day. This can be computed because solar position data is

\(^{10}\) The International System of Units defines Joule (Energy) as the product of Watts (Power) and Seconds (Time). Equations (3) and (4) are derived from this.
available, which is itself exogenous and unbiased. The subsequent issue is that daytime does not scale linearly with power production, and the datetime scalar itself will need to be scaled to mediate this. This is illustrated by the fact that daytime in December is about 25% that of June, but power production in December is only 10% that of June. In this case, a logistic cumulative distribution function is used to correct this non-linearity\(^\text{11}\). By using these transformations, scaled, but still nominal, residuals are created and smoothing methods may still be applied. The next paragraph describes the creation of the \(\text{Daytime}\) scalar, referencing the ratio of daytime to nighttime observations in the surrounding 24-hour period, and the design of the formula used to introduce non-linearity.

Define \(\text{Sun}_i\) the dummy variable with value 1 when the sun is over the horizon, and value 0 when it is not (see the empirical model in chapter V). For any given period \(j\) containing \(n\) observations, the proportion of daytime is given by:

\[
\text{Daytime}_j = \frac{\sum_{i=1}^{n} (\text{Sun}_{ij} = 1)}{n}
\]

For a period with length one day, \(n = 288\) since an observation encompasses a five-minute period. Linearly scaling residuals such that winter residuals are scaled up and summer residuals are scaled down requires a minor modification:

\[
\text{Daytime}_j = 1.5 - \frac{\sum_{i=1}^{n} (\text{Sun}_{ij} = 1)}{n}
\]

The value of 1.5 ensures that the scalar has values in the range \([0.5:1]\) for the six months with more daylight, and \([1:1.5]\) for the remainder. This ensures that winter residuals are “scaled up” and summer residuals are “scaled down,” while spring and fall residuals remain relatively untouched. Inserting this \(\text{Daytime}\) variable in a logistic cumulative probability function gives:

\[
\text{ScaledDaytime}_j = \left[ 1 + \left( \frac{\text{Daytime}_j \cdot [1 - \mu]}{\mu \cdot [1 - \text{Daytime}_j]} \right)^{1/\alpha} \right]^{-1}
\]

Where, \(\mu\) is the mean of the distribution, and \(\alpha\) controls the penalty for values deviating from the mean \(\mu\). On a yearly basis, the ratio of daytime over nighttime is roughly 0.5.\(^\text{12}\) Coincidentally, setting \(\mu = 0.5\) removes it from the equation, such that:

\[
\text{ScaledDaytime}_j = \left[ 1 + \left( \frac{\text{Daytime}_j}{1 - \text{Daytime}_j} \right)^{1.3} \right]^{-1}
\]

The value of \(\alpha\) is subjective: at value 0, it sets all \(\text{Daytime} = 1\), and at value 1, \(\text{Daytime}\) is unaffected. Between 0 and 1, winter daytime is scaled down while summer daytime is scaled up, and for \(\alpha > 1\), the inverse. \(\alpha = 1.3\) was chosen after some testing, though this is arbitrary. Ideally, \(\alpha\) would be calibrated statistically, by simulating errors and solving the \(\alpha\) such that an error in winter is as visible on the figure as a similarly sized error (in relative terms) in summer, or calculated precisely by experts. The scaled residual \(i\) during any 24-hour period \(j\), \(u_{\text{Scaled}_{ij}}\) becomes:

\[
u_{\text{Scaled}_{ij}} = u_{ij} \cdot \text{ScaledDaytime}_j
\]

The relative energy loss during this 24-hour period, \(j\):

\(^{11}\) The specific function I used was taken from Werner Antweiler’s website (see references).

\(^{12}\) Technically, the atmosphere scatters sunlight even if the sun is just below the horizon, so the ratio should be slightly greater than 0.5. For the purposes of this thesis, the mathematical simplification gained by setting \(\mu = 0.5\) is preferable to the additional accuracy of the transformation.
\[ \text{RelativeEnergyLoss} = \sum_{i=1}^{288} u_{\text{Scaled},ij} \] (8)

The result of this equation has no interpretation in kilowatt-hours. The complete equation used to transform the residuals from nominal to relative values is given by:

\[ u_{\text{Scaled},ij} = u_{ij} \cdot \left[ 1 + \left( \frac{\sum_{n=1}^{\text{Sun}_{ij}}}{n} \frac{\sum_{n=1}^{\text{Sun}_{ij}}(\text{Sun}_{ij} = 1)}{n} \right)^{1.3} \right]^{-1} \] (9)

This precise equation was used to transform the data residuals from those seen on figure 5, 6, or 7, to those seen in figures 11 and 13 in the next section. Figure 9 shows the values of the scalar (y-axis), plotted against the percentage of daytime:

Figure 8: figure depicting with which value (y-axis) a residual will be scaled, contingent on the percentage of daytime within the 24-hour period (x-axis) surrounding that residuals. This function results from equation (6).

A brief example: In December, the sun rises above the horizon roughly 20% of the time. A residual value from December will thus be multiplied by, approximately, 1.35. A residual in June has the opposite: 80% daytime and will be scaled down by multiplying it by approximately 0.65. This transformation can be used to compare energy losses easily and intuitively, in winter and summer. Moreover, the use of this transformation does not inhibit the use of rolling windows as smoothing methods. Lastly, this method deals relatively well with gaps in the data, since it counts the nearest 288 observations, not merely observations that fall within the preceding and succeeding 12 hours.

5.4.5. Visualizing Nominal and Relative Energy Losses

So far, two equations were created: one which converts power to energy (equation (4)), and one which scales the residuals to account for the varying levels of energy production during different seasons (equation (9)). These equations each serve a different purpose: one identifies the total loss of energy, the other scales the losses relative to the amount of power that can be generated given the changing weather conditions during the year. The purpose of the former is merely to assess the total amount of energy lost, while the purpose of the latter is to get an intuitive understanding of the losses and compare the performance of the model during different periods of the year. Figures 10 and 11 show the losses in nominal and relative forms. In each figure, two 50% power losses were introduced. These are the highlighted curves in June and
October. The winter period is highlighted as well, since it is assumed that there were significant energy losses due to snowfall. Both figures have been smoothed by a 3-day rolling window.

The spike in late July is a wrong prediction by the model. Either there is an undocumented malfunction in the system, or it is a large error on the part of the model. Given the limited training data for July, it is likely the latter.

Figure 10 allows us to quickly determine the amount of energy lost. Recalling equation (4), the total energy lost during a period is simply the area under the curve during that period. Using a back-of-the-envelope calculation, the energy loss in June must be roughly $0.015 \cdot 30 \text{ days} \cdot 288 \sim 130 \text{ kWh}$, corresponding to a financial impact of about 260 Swedish Crowns, at average 2019 prices. However, this figure gives the misleading impression that the photovoltaic modules suffered from a much greater failure in June than in October. In both months, the loss was 50% of expected production.

Figure 11 remedies this shortcoming. By scaling the energy losses such that the different seasonal energy production levels are considered, according to the equation (9), it can be observed that the PV array suffered from a similarly sized malfunction in June and October alike. Note that both figures 10 and 11 are built off the same data which underpins figures 4 through 8. The data has merely been modified so as to obtain better visualizations.
5.4.6. A Real-World Fault

So far, only results using simulated faults have been shown. This is due to the absence of recorded faults in the system. Nonetheless, some data from October and November 2021 in which one solar module (out of four total) in the PV array of system 1 was completely disconnected, is available to us. The result of this is an energy loss of approximately 25%. Note that the loss cannot be precisely determined: different modules can contribute unevenly to the total energy production for a variety of reasons (such as different shading patterns). Nonetheless, the educated guess for the size of the loss is roughly 25%.

On figures 12 and 13, the simulated (green) and real-world (red) power losses are plotted side-by-side, since they occurred on the same PV array. Other data for 2020 is missing, so the real-world fault (in red) must be shown in complete isolation. The simulated faults (still at 50%) are shown in green. On figure 12, the total energy loss is visible (the area under the curve) and minimal compared to the simulated losses. This is because the fault occurred late in the year when the sun shone dimly.

On figure 13, on the other hand, the relative size of the malfunction becomes clear. The real-world fault, estimated to be 25%, has about half the height of the 50% simulated fault. This is encouraging, since this is the result one would expect to see if the model were functioning correctly.
CHAPTER VI: DISCUSSION

This chapter offers a more general and reflective discussion of the method and the results obtained, as well as how they connect to the stated aim in chapter I. Specific recommendations for potential improvements are given at the end of this chapter.

6.1. Discussion of the Strengths and Weaknesses of the Model

The method is designed to fulfill three goals: to create a model which deals well with nonlinearity, provide a usable set of residuals, and create a system for visualizing the results such that they can be interpreted quickly and intuitively. These goals have been fulfilled to varying degrees of success, but have been fulfilled, nonetheless. Despite this, obvious problems remain.

First, the model requires a large dataset for training. Under the best circumstances, one full year’s worth of continuous uninterrupted data. This can be problematic for two reasons: there is no fault-detection for the first year of power generation, and, as it is assumed the during this first year, no significant fault occurs. This leads to the second problem.

Second, installation faults, i.e., faults introduced during the construction of the PV array, cannot be detect under any circumstances. The model relies entirely on the assumption that the “training year” represents the system at peak performance. How strongly this assumption holds, is up for debate, but it will be a vital consideration in any practical application.

Third, the model works reasonably well but does require a pyranometer (radiation sensor), which is not standard equipment for residential solar arrays. One the one hand, this is a weakness, but on the other hand it is a strength too, as the only modification required to a residential system is said pyranometer. The dataset included data from a host of thermometers, a windspeed meter, and voltage meters at many individual parts of the array. All of these improved the accuracy of the model, allowing it to detect smaller faults more reliably. However, including these in the model would require too many modifications to residential systems and render the effort unfeasible and irrelevant. Therefore, prediction accuracy was sacrificed to stay as close to the original aim as possible. Unfortunately, removing the pyranometer proved a bridge too far. It need not be a major problem: pyranometers are cheap, and internet-connected data collection systems are already in place at residential systems. The training and testing can then be done in the cloud.

6.2. Discussion of the Performance of the Model

The results are satisfying but not perfect: the model predicts yearly average power production very accurately, but on the weekly or even monthly scale it can break down. This is evidenced by the sudden jump in the residuals during the month of July (observe figures 10, 11, 12, and 13). This is the largest error the model has made. I have estimated it to be approximately 10 kWh of energy. It is fair to say that the model cannot reliably detect faults smaller than this.

Moreover, there is a limited amount of available testing data. Ideally, the model should be trained and tested over several years and many different PV arrays. This is outside of my control. It was tested with different training samples on different test sets, and the model works well overall, but it tends to find significant energy losses without there being any evidence of a fault having occurred.

There are no reasons to think that the model would behave fundamentally differently on datasets from other PV systems. Certainly, different circumstances can lead to different results, but the model uses three layers (radiation, time, and solar position) to estimate power output and should be stable. Moreover, the current setup is far from “ideal”: lots of shading, cold temperatures, snow is common… if anything, nearly any place South of Borlänge will provide better training data.

In the real world, faults are not as “temporary” as they have been shown in this thesis. A disconnected cable does not reconnect itself, after all. Rather than temporary jumps in the curve, as shown in figures 10, 11,
12, and 13, the jump would be permanent, and therefore far more visible. Currently, the model only offers visual output to evaluate the performance of the PV array. Over time, this element could be replaced by a more sophisticated system. A model could be applied on the residual values to learn when it is not functioning properly, and then escalate a message to the user. Alternatively, some expert-made rules can be established.

Overall, I estimate that this model can reliably detect consistent power losses which are greater than five or ten percent. This would be more than enough to justify the cost of an additional pyranometer.

### 6.3. Future Improvements

This section compiles several recommendations for improvements in possible future work. It lists several logical next steps and general insights.

- Use data from more moderate latitudes and sunnier regions: data from Northern regions makes for a poor test case, as the radiation values are rather extreme, and the prolonged periods of overcast weather do nothing but hamper the performance of the model.
- Construct a crude simulation of the solar array. Tools for python are available in the *PV Solar* package. This has the potential to increase the consistency of the model.
- Create a system which automatically detects faults: I have devised a visual way of identifying faults but have not created a program which does so itself. The results from this model might be used in a neural network or clustering setup.
- Attempt to partially “pre-train” the model using data from similar arrays, such that the on-site training only needs to incorporate local, cyclical effects. Removing the need for on-site training greatly benefits the practical viability of the model.
- Use data with a narrower sampling interval, ideally every second – more data, more interactions.
- Let it be a cooperation between an engineer and a data scientist. I have been consistently hampered by my lack of technical understanding regarding photovoltaic technology and engineering in general.
- Try to gather data which includes documented faults for model performance testing.
CHAPTER VII: CONCLUSION

The aim of this thesis has been to develop a model which can reliably identify faults which occur in residential PV arrays and lead to power losses. It has also been the aim to ensure that this model can be implemented in practice with minimal modification to current PV array infrastructure. In both regards, the model mostly achieves these aims.

It can detect modest to large power losses reliably, but it has difficulty detecting power losses which only occur at timescales shorter than one or two weeks, or which are no bigger than five or ten percent of the array’s optimal output. It also has the major drawback that it requires at least one full year’s worth of training data and operates on the assumption that the array performs optimally during this time.

Minimal modification to current or future residential systems would be required, as infrastructure for collecting and uploading data is available. All that is required is the addition of a pyranometer. Both time and DC power output are already being recorded by current residential systems.

The methods described in this thesis also lead to results in a format which can be interpreted intuitively. Both the size of any power loss and the relative performance of the PV array can be deducted swiftly from the graphical output.

Most importantly, this work has shown that it is practically feasible to create a reliable, low-cost monitoring system for residential PV arrays. This can increase the electricity generated by an environmentally sustainable source and provide a financial benefit while doing so.
REFERENCES


Other:

Main data: Fiedler Frank, Högskolan Väst
Location data: Google Maps