Fault Detection and Performance Visualisation for a Grid-Connected Photovoltaic Power Plant Using Sensor Data

by

Wayne Peter Dyamond

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (E&E) in the Faculty of Engineering at Stellenbosch University

Supervisor: Dr. Arnold J. Rix

December 2019

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: ..... December 2019 .....

Abstract

Fault Detection and Performance Visualisation for a Grid-Connected Photovoltaic Power Plant Using Sensor Data

W.P. Dyamond

Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MEng (E&E)

December 2019

The rising energy demand and need for alternatives to fossil fuel based power generation have increased the utilisation of photovoltaic (PV) power plants. The reliable operation of PV power plants will maximise energy delivery, boost public opinion on PV technology and lead to financial gains for investors. Accurate fault detection and effective plant performance reporting could significantly reduce system downtime, power loss and safety hazards. The work presented in this document aims to investigate improvements to fault detection and performance visualisation for an utility-scale PV power plant using measured sensor data. 560 GB of operational data from a 75 MWp capacity solar power plant is obtained for the research project. Data pre-processing and cleaning results in a 167 GB dataset containing measured values for 12 595 different signals over the period of three years. A fault detection procedure based on the comparison of modelled and measured string-pair current is proposed. The expected current is modelled using the single diode electrical model. The Euclidean distance between the measured and expected values is calculated for all string-pairs in the power plant. Events are flagged as possible faults when the corresponding Euclidean distance is considered an outlier. The fault detection procedure is tested on the dataset and a sample accuracy of 94.67% is achieved. A visualisation tool based on the performance comparison of all string-pairs is developed. The visualisation is used to verify events detected during the fault detection procedure as well as visualise average performance and degradation differences between string-pairs. An average DC degradation rate of 0.38% per year is observed during string-pair degradation analysis.

Uittreksel

Ontdekking van Stelsel Fout Gebeurtenisse en Werkverrigting Visualisering in 'n Netwerkverbindte Fotovoltaïse Kragsentrale deur die Gebruik van Sensor Data

("Fault Detection and Performance Visualisation for a Grid-Connected Photovoltaic Power Plant Using Sensor Data")

W.P. Dyamond
Departement Elektries en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MIng (E&E)
Desember 2019

Die toenemende energie aanvraag en behoefte om alternatiewe bronne vir energieopwekking te gebruik, het gelei tot die ontwikkeling van meer fotovoltaïese (FV) kragstasies. Betroubare werksverrigting van FV-kragsentrales sal die energie-opbrengs verhoog, die publiek se mening oor FV-tegnologie verbeter en tot hoër winste vir beleggers lei. Akkurate foutopsporing en effektiewe verslagewing van aanleg werksverrigting kan stelsel stiltand, kragverlies en veiligheidsrisiko's aansienlik verminder. Die navorsing wat in hierdie dokument uitgelê word, mik om verbeteringe aan foutopsporing en visualisering van stelsel uitset vir 'n netwerkverbindte FV-kragstasie te ondersoek. 560 GB se gemete sensor data van 'n 75 MWp sonkragaanleg word in hierdie navorsingsprojek ondersoek. Verwerking van die data verminder die grote tot 167 GB. Die datastel bevat meetingswaardes vir 12 595 verskillende bronne vir drie jaar. 'n Foutopsporingprosedure, gebaseer op die vergelyking van gemodelleerde en gemete stringpaarstroom waardes, word aangebied. Die verwagte stroom word gemodelleer met behulp van die enkeldiode elektriese model. Die Euklidiese verskil tussen die gemete en verwagte waardes word bereken vir alle stringpare in die kragsentrale. Uitskieters word as moontlike foute geïdentifiseer. Die foutopspooringprosedure word op die datastel getoets en behaal 'n steekproef akkuraatheid van 94.67%. 'n Visualiseringstoepassing, wat die werkverrigting van alle stringpare vergelyk, word ontwikkel. Die visualisering word gebruik

om gebeurtenisse wat tydens die foutopspooringprosedure geïdentifiseer is, te bevestig. Die toepassing word ook gebruik om die verskille in gemiddelde uitset en agteruitgang van drywing tussen stringpare te visualiseer. 'n Gemiddelde gelykstroom-agteruitgangskoers van 0.38% per jaar word waargeneem tydens die analise.

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations:

- Heavenly Father for the ability to pursue my calling and the abundance of blessings He provides.
- Dr. Arnold J. Rix, my project supervisor, for the endless support and guidance.
- Scatec Solar for providing funding towards my Masters.
- My family and Rose for their love and support.
- My friends and peers in the Media Lab. A special thanks to Armand du Plessis for his guidance and contributions as well as Carel Landman for the encouragement and laughter.

Dedications

This thesis is dedicated to Rose.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

a-Si	Amorphous Silicon
AC	Alternating Current
AM	Air Mass
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
ARIMA	Auto Regressive Integrated Moving Average
c-Si	Crystalline Silicon
CdTe	Cadmium Telluride
CEC	California Energy Commission
CIGS	Copper Indium Gallium Selenide
CSD	Classical Seasonal Decomposition
CSV	Comma-Separated Values
DC	Direct Current
DTW	Dynamic Time Warping
ECM	Earth Capacitance Measurement
GB	Gigabyte
GUI	Graphical User Interface
HMI	Human-Machine Interface
LOESS	LOcally wEighted Scatterplot Smoothing
LR	Linear Regression
MPPT	Maximum Power Point Tracking
MSE	Mean of Squared Errors
MTU	Master Terminal Unit
MW	Megawatt
MWp	Megawatt peak
NAS	Network-Attached Storage
NREL	National Renewable Energy Laboratory
O&M	Operations and Maintenance

OECD	Organisation for Economic Cooperation and Development
PLC	Programmable Logic Controllers
POA	Plane-of-Array
PR	Performance Ratio
PV	Photovoltaic
RDBMS	Relational Database Management System
REIPPPP	Renewable Energy Independent Power Producer Procurement Programme
RMSE	Root Mean Square Error
RTU	Remote Terminal Units
SCADA	Supervisory Control And Data Acquisition
SQL	Structured Query Language
SSM	Sunny String-Monitor
STC	Standard Testing Conditions
TDR	Time Domain Reflectometry
TS	Transformer Station
TSK-FRBS	Takagi-Sugeno-Kahn Fuzzy Rule-Based System
UV	Ultraviolet

Constants

k	Boltzmann's constant ($1.3806503 \times 10^{-23}\,\mathrm{J/K}$)
q	Electron charge ($1.60217646 \times 10^{-19}\,\mathrm{C}$)

Chapter 1

Introduction

1.1 Background

The global energy demand is ever increasing due to population growth and power requirements associated with technological advancements [1]. Global energy consumption experienced annual growth of 2.9% in 2018 [2]. Traditional energy generation methods are largely dependent on burning fossil fuels like coal, natural gas and oil. These methods account for roughly 64% of the total electricity generation market [2]. The major drawbacks of using fossil fuels include greenhouse gas emissions produced during combustion and limited availability of the resources. Carbon emissions from energy use increased by 2% in 2018, the highest growth rate in seven years, which makes the need for cleaner alternative energy sources clear [2]. The renewable energy sector has seen rapid growth due to technological improvements, cost reductions and supportive government policies [3]. Hydroelectricity, solar energy and wind power are some of the popular technologies that are implemented worldwide. Hydroelectricity accounts for about 16% of the global energy generation and non-hydro renewable energy sources have shown a significant growth from 3% to 9.3% in the past decade [2]. Figure 1.1 shows the electricity production by source, between 1974 and 2018, for countries part of the Organisation for Economic Cooperation and Development (OECD). Solar, wind and geothermal electricity production have increased substantially in the last ten years. Solar photovoltaic power generation is expected to dominate the renewable capacity growth in the next few years [4]. Solar power is an attractive solution since the cost of PV modules has dropped significantly, while the efficiency has improved. The United States National Renewable Energy Laboratory (NREL) reported a 66% decline in cost per Watt DC from 2010 to 2018 [5].

South Africa has also seen an increase in renewable energy generation which is driven by strained electricity supply and favourable weather conditions [7]. The South African Renewable Energy Independent Power Producer Procure-

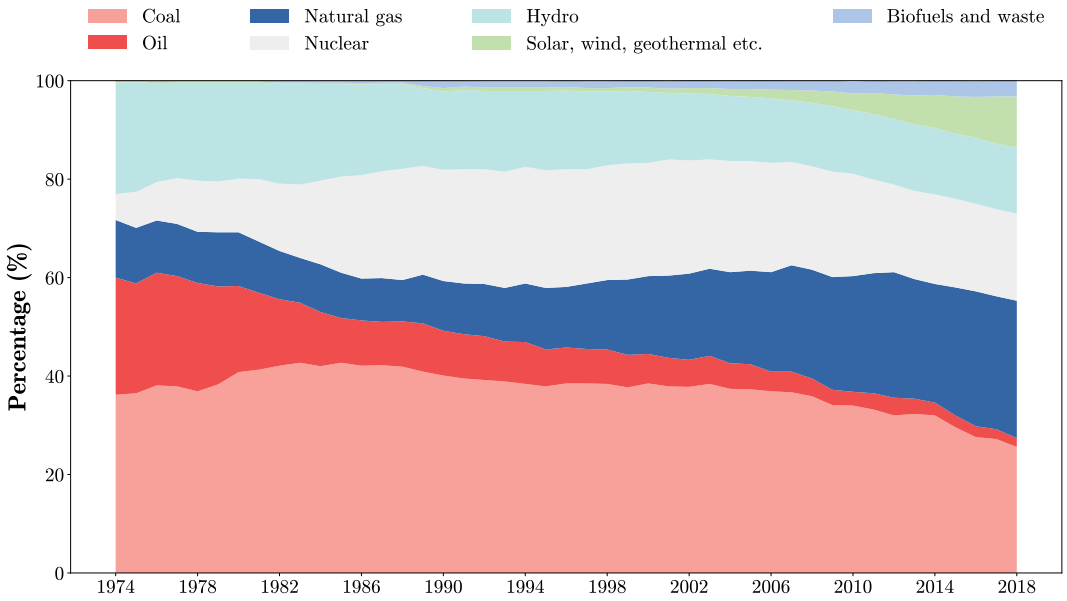


Figure 1.1: Electricity production by source (data acquired from [6]).

ment Programme (REIPPPP) was launched to increase private investment into renewable energy generation. The programme has resulted in the procurement of wind, solar photovoltaic and other renewable energy adding up to 6328 MW in generation capacity [8]. Along with the REIPPPP, South Africa has also committed to the Integrated Resource Plan, drawn up in 2010, which aims to achieve 9600 MW of solar power capacity by 2030 [7]. The adoption of solar PV technologies plays a fundamental part in accommodating rising energy demands, while also limiting the use of fossil fuels. Unfortunately, due to the dependence on weather, PV electricity supply is variable. An increased share of electricity supply by PV power plants, therefore, necessitates the reliability of PV power generation during operating hours.

1.2 Problem Statement

Faults and failures that occur in a PV system can result in energy loss, system shutdown or serious safety hazards [9]. Effective detection and location of system faults may accelerate response times for corrective measures. Multiple researchers have investigated fault detection and diagnosis techniques for PV systems. The methods proposed in previous work was mainly tested in simulated environments or implemented on small- to medium-scale systems. Considering the influx of large PV power plants, the need for investigating fault detection in utility-scale PV systems arises. Along with accurate fault detection, effective plant performance reporting is needed to ensure that operations and maintenance (O&M) personnel are informed about unexpected system behaviour. Implementation of accurate fault detection and plant performance

evaluation tools could ultimately improve the reliability and stability of large-scale PV systems. Better stability and minimised downtime also correlates to financial gain for independent power producers and investors.

1.3 Aim and Objectives

Solar power plants generate considerable amounts of data from sensor measurements. This project aims to explore the use of data analysis for fault detection and performance evaluation of a utility-scale PV power plant. The main research question is formulated as: how can measured operational data be used to improve fault detection and performance evaluation of a large-scale PV power plant? The effective communication of failures and overall performance to O&M teams is expected to improve the stability of the power generation. The proposed work aims to address the following objectives:

- Process the obtained measured sensor data into a clean, workable dataset. Investigate the data quality and missing data. Deal with outliers appropriately, while also maintaining the integrity of the data. Ensure that the processed dataset is centralised and accessible to the few researchers that will be using the data.
- Use the processed data and model the expected behaviour of specific PV devices in the power plant.
- Develop a fault detection procedure based on the comparison of measured and modelled values. Test the procedure on measured data from a large-scale solar power plant to evaluate the accuracy.
- Create a visualisation tool which compares the performance of subsystems in the power plant. Analyse average performance, and change in performance over time, using the visualisation.

1.4 Thesis Outline

Using the objectives as a guideline, this thesis aims to address the research question posed in Section 1.2. The investigation into data processing, fault detection and performance analysis for a solar power plant using measured sensor data is structured as below:

- **Chapter 2:** covers theory relating to the building blocks of a photovoltaic system. The basic concepts of PV devices from solar cells to module arrays are described. The combination of PV devices, power converters and a monitoring system is defined as the basis for a grid-connected PV power plant. After the basics regarding the building blocks

of a PV system are established, the modelling of devices in a PV system is reported. The use of the single diode model for calculating the current and voltage of a PV device is outlined. A method for estimating the five parameters of the single diode model equation is documented. Two methods for solving the equation are also described. Finally, the chapter concludes with the implementation of the modelling method using a Python library. The comparison between modelled and measured current values is used to test the modelling method. The modelling is found to be accurate and validated for use in the rest of the project.

- **Chapter 3:** contains an investigation into the data that was obtained for the project. A brief overview of the measurements that are available and the layout of the power plant is given. The manner in which the data was processed is described. The data processing involves combining raw data files into a single database, removing redundant columns or measurements and finally omitting any values that are considered erroneous. A data pipeline for integrating future data is also described. The chapter concludes with an analysis of missing data in the processed dataset.

- **Chapter 4:** includes the development of the fault detection procedure. A review of common faults that occur in PV systems and fault detection approaches presented in literature is covered. Considering the work identified in literature, a fault detection procedure based on the comparison of modelled and measured behaviour is presented. The fault detection procedure is developed to be implemented on string-pair current data, as this is the lowest level measurement available in the dataset. Time series comparison is investigated, whereafter Euclidean distance is decided as the appropriate similarity measure. The fault detection procedure is tested on three years of measured data and the results are presented. An improvement to the proposed algorithm is implemented and the fault detection algorithm achieves a 94.67% sample accuracy.

- **Chapter 5:** describes the use of visualisation to relay information about the performance of the power plant. A tool is proposed that compares and visualises string-pair currents for the entire plant. The use of this visualisation for validating faults detected in Chapter 4 is demonstrated. An alteration of the first visualisation tool is proposed for analysis of long-term performance. The average power produced by each string-pair is used as a performance indicator and represented visually. The chapter also covers an overview of performance degradation at string-pair level. The implementation of the visualisation tool in real-time, and long-term performance evaluation is considered to be valuable for decreasing fault response time and therefore improving output stability of the power plant.

- **Chapter 6:** is the concluding chapter for this project. Closing remarks for all previous chapters are stated. Finally, possible future improvements and research directions are outlined.

Chapter 2

Modelling Photovoltaic Devices

This chapter provides an overview of basic photovoltaic devices and the fundamental building blocks of a PV system. Following a brief introduction to PV systems, the modelling of PV devices is covered. The single diode model equation and the implementation thereof are discussed in detail. An implementation of the single diode model in Python is used to model the power behaviour of a string-pair. The modelled current is compared to a baseline parametric model as well as the measured current values. The modelled string-pair current is used extensively in the following chapters.

2.1 Building Blocks of a Photovoltaic Power Plant

A large-scale photovoltaic system can be designed to supply power to the utility grid. This system is often known as a PV power plant or utility-scale system. A PV power plant consists of various smaller components, the basics of which are discussed in this section.

2.1.1 PV Module Arrays

PV cells are used to convert sunlight directly into electricity. When the cell is exposed to solar energy from photons, excited electrons in the material are able to flow into conductive carriers located on the cell. The most common PV cells use either crystalline or thin-film technology. Crystalline cells are constructed from a thin layer of semiconductor material, often crystalline silicon (c-Si), and can be produced to employ a monocrystalline or polycrystalline lattice structure. Monocrystalline cells are slightly more efficient, but polycrystalline cells are less expensive to manufacture. Crystalline silicon technology accounts for 90% of the market share in terms of power produced annually [10]. This technology is popular due to relatively high efficiency, reliability and life span.

Thin-film cells follow similar working principles to crystalline cells. The cells are constructed from layers of photovoltaic materials deposited onto a glass or plastic surface. Cadmium telluride (CdTe), copper indium gallium selenide (CIGS) and amorphous silicon (a-Si) are three of the common thin-film technologies, with CdTe being the most popular [10]. Thin-film technology allows the cells to be flexible and light weight, but can also be sandwiched between glass to obtain a rigid structure [11]. Thin-film cells are often less expensive to manufacture than crystalline silicon cells, but are less efficient and have a shorter life span than the c-Si counterpart.

A module is constructed from multiple interconnected cells. Typical silicon-based cells can only produce an open-circuit voltage of about 0.5 volts. Cells in a module are therefore connected in series to increase the total voltage output. A PV module often consists of more than 60 cells in series. In this project, 'module' is used when referring to a PV device consisting of multiple interconnected cells.

The term array or module array refers to multiple connected modules configured to obtain a specific power output. Modules are connected in series (known as a string) to increase voltage, and in parallel to increase current output. A PV system can be built to contain multiple arrays and configured to meet the designed output voltage and current. The term string-pair is often mentioned in the rest of the work. String-pair refers to an array configuration of two strings connected in parallel. Figure 2.1 shows an illustration of a PV cell, module and array. Modules in a PV system can be mounted in a fixed, single-axis tracking or dual-axis tracking mounting structure. Fixed structures are stationary supports that have a fixed orientation and tilt optimised for maximum solar irradiance exposure. Fixed mounting structures are cost-effective, durable and space efficient, but result in lower overall power output. Single-axis tracking uses a mounting structure with a fixed orientation; however, the tilt is automatically adjusted according to the position of the sun. Dual-axis tracking allows variable orientation and tilt of the PV modules which maximises direct normal solar irradiance. Tracking structures results in improved overall performances due to better irradiance exposure. These structures are, however, more expensive, require more maintenance and use more land area than fixed-tilt systems. PV modules produce direct current (DC), but since the the power grid utilises alternating current (AC), power conversion is needed.

2.1.2 Power Conversion

Inverters are devices that convert input DC power to AC power. Inverters may also include maximum power point tracking (MPPT) which sets the power point of all arrays connected to the inverter. The MPPT control system adjusts

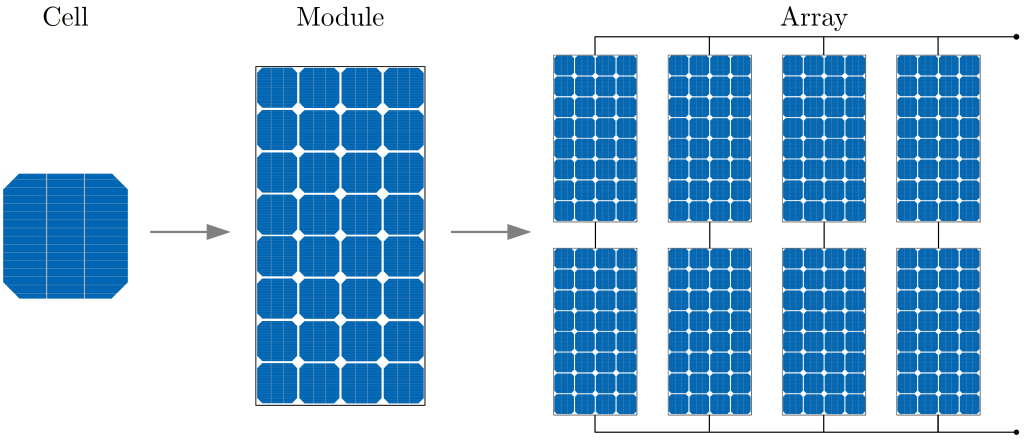


Figure 2.1: The elements of a solar array.

the internal resistance of the inverter slightly until the maximum power output is observed. This behaviour is important to accommodate changing weather conditions. Inverters are classified as centralised- or string inverters depending on the input capacity. Centralised inverters have a high input capacity and convert the power of multiple strings simultaneously. Since many strings are connected to a single inverter, if a centralised inverter fails, a large power loss ensues. String inverters have a lower capacity and convert power at string level. MPPT is more effective at string level, since variation in solar irradiance is limited to a smaller area, and power loss is minimised if a string inverter fails. Inverter output voltages are usually much lower than the grid voltage, therefore transformers are used to increase the voltage from inverters to match the grid-voltage.

2.1.3 Monitoring System

A comprehensive solution for the control, supervision and fault detection of a PV plant is termed as the monitoring system [9]. Along with fault diagnosis techniques, the monitoring system is crucial to maximise the operational reliability of a PV system [12]. Many commercially available monitoring systems share the following main features:

- Real-time monitoring and performance tracking of subsystems in the power plant.
- Collect and store operating data from main components in system including weather data if applicable.
- Control remote functions of connected inverters.
- Generate graphic-based reports of plant performance.

- Identify system malfunction based on thresholds and deviance from normal operating conditions.

The monitoring system is often built on a control architecture called Supervisory Control And Data Acquisition (SCADA). SCADA is used for data collection and control at the supervisory level for large-scale power plants [13]. Figure 2.2 shows the different levels of a typical SCADA control system. Sensors and control devices interface directly with field level hardware. In the case of a PV power plant, this could be an inverter for example. Programmable logic controllers (PLC) or other remote terminal units (RTU) can receive data from sensors and send control signals to the control devices. A master terminal unit (MTU), connected to the RTUs, hosts the monitoring system software and establishes the central control for the entire system. Control inputs, for instance an inverter remote shutdown command, can be passed to the MTU through a user interface, also known as the human-machine interface (HMI). Data from low-level sensors is routed through the SCADA system to be displayed in graphical reports and can be stored on a database. Similarly acquired data from sensors in a large-scale PV power plant is the focus in this project.

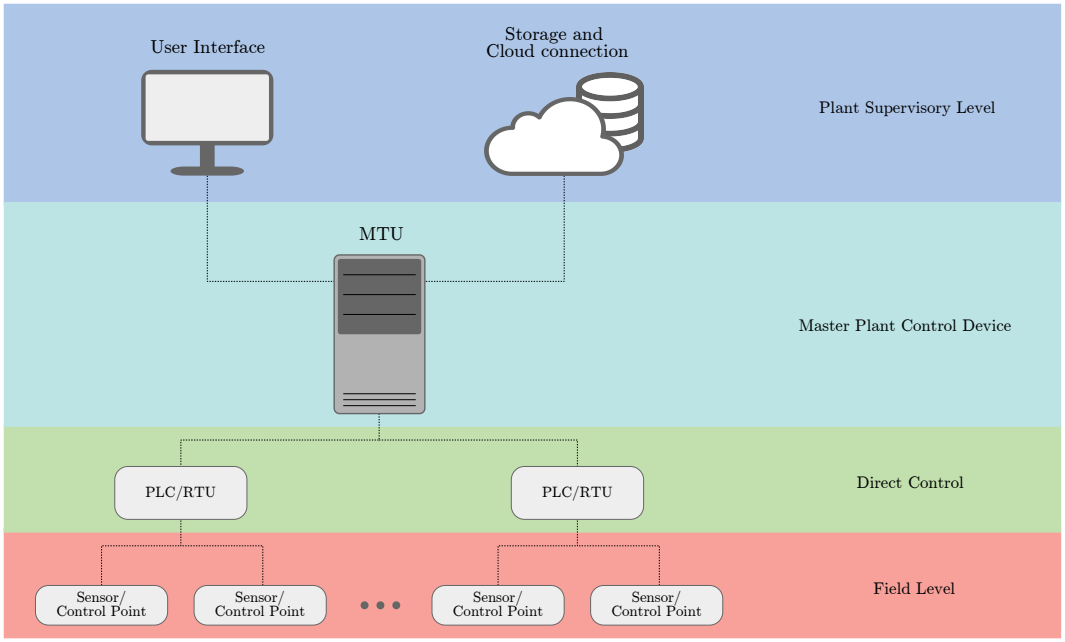


Figure 2.2: Overview of a SCADA system.

Knowing the expected behaviour of devices in a PV system is an important part of fault detection. Given a number of inputs, a model can be used to calculate the expected output for a PV device. The following section provides a review of modelling methods.

2.2 Modelling Expected Power

This project uses a comparison between expected and measured power in order to detect PV system faults. The power output from a photovoltaic module depends on the received solar radiation, the cell temperature and the load resistance [14]. An accurate model of expected power is needed for the comparison procedure during fault detection. Relevant literature shows the use of parametric models [15; 16], neural networks [17] or electrical models [14; 18; 19] for calculating expected power. These models rely on meteorological data including solar irradiance, temperature and wind. Weather data can be obtained on-site using measurement equipment such as pyranometers and temperature sensors or off-site from neighbouring weather stations. In some cases, satellite observed irradiance derivations are used to eliminate the need for ground measurements [15]. Electrical models are advantageous because the model requires only the configuration information of the PV system. Parametric and artificial intelligence based models are trained on historic measured data. The suitable method for calculating expected power of a PV system depends on the available data, complexity of the implementation and accuracy needed. This project investigates the use of an electrical model for the relatively simple implementation, independence on system configuration and high accuracy.

2.2.1 Single Diode Model

Figure 2.3 shows the single diode equivalent circuit of a PV device. The ideal PV cell has no shunt (R_{sh}) or series (R_s) resistance. Equation 2.1 is used to calculate the output current.

$$I = I_L - I_D \tag{2.1}$$

The current I_D can be calculated using the Shockley diode equation, resulting in Equation 2.2.

$$I = I_L - I_o[e^{\frac{qV}{nkT_c}} - 1] \tag{2.2}$$

Where I_L is known as the light current, I_o is the diode reverse saturation current, n is defined as the diode ideality factor, T_c refers to the cell temperature, and q and k are the electron charge and Boltzmann's constant respectively. Any practical PV device (cell, module, string or array) can be modelled by adding the effect of the shunt and series resistances [14]. The output current I for a practical PV device can be calculated as a function of the output voltage as seen in Equation 2.3.

$$I = I_L - I_o\left[e^{\frac{V+R_sI}{a}} - 1\right] - \frac{V + R_sI}{R_{sh}} \tag{2.3}$$

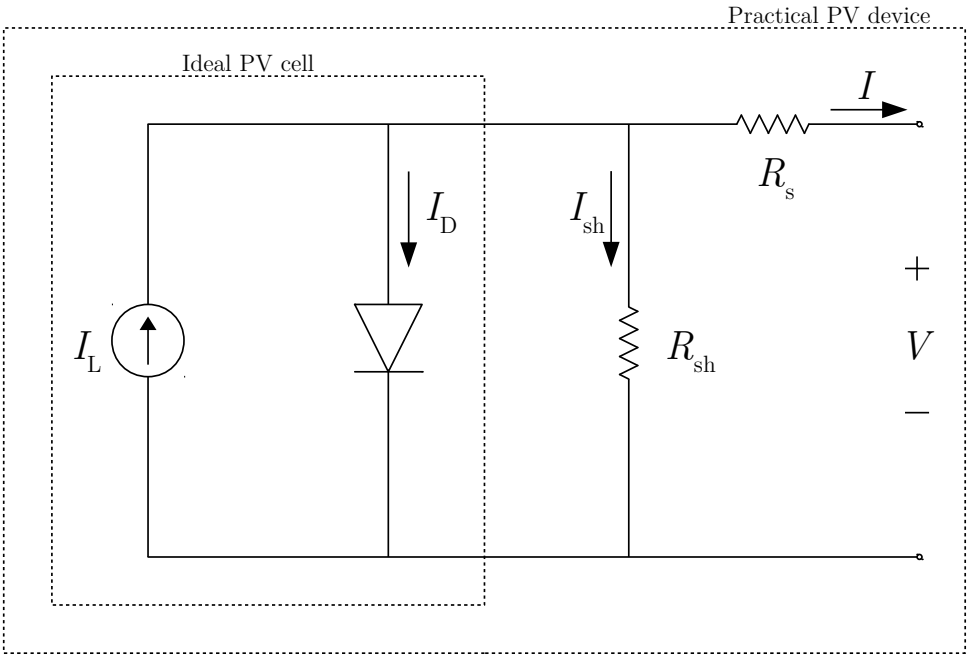


Figure 2.3: Single diode equivalent circuit of a PV device [14; 20].

The modified diode ideality factor of the device is $a = N_s nkT/q$ with N_s the number of cells connected in series. Equation 2.3 contains five unknowns namely: I_L, I_o, a, R_{sh} and R_s. These parameters are to be calculated before the equation can be solved.

Different methods for estimating the five parameters arise in literature. These methods can be classified into three major categories: non-iterative, numerical and optimisation approaches [21]. Numerical methods depend on solving a system of equations relating to 2.3 numerically [14] or iteratively [20]. Numerical methods tend to have high accuracy, but can be computationally expensive and converge to suboptimal solutions. Non-iterative methods [21] employ simplifications of the equations resulting from numerical methods to explicitly calculate the parameters. Non-iterative methods are easy to implement, but often return lower accuracy. Optimisation techniques [22] adjusts parameter values to fit the model output to a set of measured data points, usually on the I-V curve. Curve fitting and other optimisation methods result in high accuracy and are independent of the module technology, but can become computationally expensive and require additional measured data [21]. Some methods depend only on information listed by the manufacturer in the module datasheet. Module manufacturers typically provide the open-circuit voltage (V_{oc}), short-circuit current (I_{sc}), maximum power current (I_{mp}) and voltage (V_{mp}), and temperature coefficients for open-circuit voltage and short-

circuit current ($\beta_{V_{oc}}$ and $\alpha_{I_{sc}}$, respectively) [14]. These measurements are taken at standard testing conditions (STC) where the irradiance (G) is 1000 W/m^2 and the cell temperature (T_c) is 25 °C with an air mass (AM) number of 1.5.

The numerical method as proposed in [14] is the focus of this project as it is widely used in literature and is implemented in the PVLIB-Python library used in the fault detection procedure.

2.2.2 Estimating the Five Parameters of the Single Diode Model

De Soto et al. provides equations for calculating the five unknown parameters as a function of the parameters at STC [14]. These reference parameters and the temperature and irradiance dependencies are used to define numerical equations for calculating the parameters at conditions other than STC. The modified ideality factor is modelled as a linear function of cell temperature in Equation 2.4.

$$a = a_{\text{ref}} \frac{T_c}{T_{c,\text{ref}}} \tag{2.4}$$

$T_{c,\text{ref}}$ and a_{ref} are the cell temperature and modified ideality factor for reference conditions, while T_c and a are the cell temperature and modified ideality factor parameters for the new operating conditions. The diode reverse saturation current, I_o, is influenced by change in temperature as noted in Equation 2.5.

$$I_o = I_{o,\text{ref}} \left[\frac{T_c}{T_{c,\text{ref}}}\right]^3 \exp\left[\frac{1}{k}\left(\left.\frac{E_g}{T}\right|_{T_{\text{ref}}} - \left.\frac{E_g}{T}\right|_{T_c}\right)\right] \tag{2.5}$$

E_g is the material band gap energy, where $E_{g,\text{ref}} = 1.121$ eV for crystalline silicon cells [14]. The band gap energy also shows small temperature dependence and is adjusted using Equation 2.6.

$$E_g = E_{g,\text{ref}}(1 - 0.0002677(T - T_{\text{ref}})) \tag{2.6}$$

The light current, I_L is dependent on the effective irradiance (G), the cell temperature and the short-circuit temperature coefficient ($\alpha_{I_{sc}}$). Equation 2.7 then shows the relation of the light current at operating conditions to the reference conditions.

$$I_L = \frac{G}{G_{\text{ref}}}[I_{L,\text{ref}} + \alpha_{I_{sc}}(T_c - T_{c,\text{ref}})] \tag{2.7}$$

Note that Equation 2.7 differs from the equation for light current in [14]. The single factor effective irradiance (G) replaces the product of absorbed solar irradiance (S) and air mass modifier (M) as used in [14].

Equations for estimating a, I_o and I_L are shown in 2.4 - 2.7. The parameter R_s is assumed to stay constant at the reference value [14]. The effective shunt resistance increases as a function of irradiance shown in Equation 2.8. Using this numerical approach, the five unknown parameters can be calculated at operating conditions that are different than the STC.

$$R_{sh} = R_{sh,\text{ref}} \frac{G_{\text{ref}}}{G} \tag{2.8}$$

Reference parameters are required for estimating the model parameters using the methods shown above. These reference parameters may be obtained using the open-circuit voltage at a temperature close to 25 °C and using the equations proposed in [14]. Non-iterative methods like thoroughly investigated in [21] could also be used to estimate the reference parameters. Appendix A.1 shows the implementation of such a non-iterative approach. Alternatively, if the module I-V curves at STC are available, [22] shows parameter estimation using I-V curve data. Some reference parameters for commercial modules may also be obtained from the California Energy Commission (CEC) module database as proposed in [23].

2.2.3 Solving the Single Diode Model

Once the parameters for Equation 2.3 are determined, the equation can be solved to determine the expected output power of the PV device [20]. The single diode equation can be solved either by numerical methods, like the Newton-Raphson method, or by using the Lambert W function to obtain an explicit solution [21].

2.2.3.1 Newton-Raphson Method

The Newton-Raphson method provides an iterative approach to solving the roots of a function $f(x)$ [24]. An initial guess towards the values of the root is needed, after which the method repeatedly finds values closer to the real value of the root. This iterative process is defined in Equation 2.9.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2.9}$$

In the case for solving Equation 2.3, the function can be defined as in Equation 2.10 and the derivative results in Equation 2.11.

$$f(I) = I_L - I_o\left[e^{\frac{V+R_sI}{a}} - 1\right] - \frac{V + R_sI}{R_{sh}} - I \tag{2.10}$$

$$f'(I) = -\frac{I_oR_s}{a}\left[e^{\frac{V+R_sI}{a}}\right] - \frac{R_s}{R_{sh}} - 1 \tag{2.11}$$

Using Equation 2.9 and substituting $f(I)$ and $f'(I)$, the Newton-Raphson method can be applied. An initial guess for the value of I is used in the first iteration. A threshold comparing the previous value to the current calculated value can be implemented to stop the process in the case of convergence. A maximum number of iterations is also used to prevent an endless non-convergence loop. Some drawbacks to using the iterative Newton-Raphson method is dependence on initial guess values, possible non-convergence and high computational cost. The Lambert W function provides an explicit formulation of the single diode equation and could be used to mitigate some of these drawbacks.

2.2.3.2 Lambert W Function

The Lambert W function $W\{x\}$ is defined as the inverse of the function we^w [21]. Given $we^w = x$ the solution of w is the Lambert W function $W\{x\}$. Since real valued positive numbers are considered as the solution for the single diode equation, the principal branch, W_0, is used. Implementation of the Lambert W function to obtain explicit formulations of output current and voltage are shown in equations 2.12 and 2.13.

$$I = \frac{R_{sh}(I_L + I_o) - V}{R_s + R_{sh}} - \frac{a}{R_s} W_0\left\{\frac{R_s R_{sh} I_o}{a(R_s + R_{sh})} e^{\frac{R_s R_{sh}(I_L + I_o) + R_{sh} V}{a(R_s + R_{sh})}}\right\} \tag{2.12}$$

$$V = R_{sh}(I_L + I_o) - (R_s + R_{sh})I - aW_0\left\{\frac{R_{sh} I_o}{a} e^{\frac{R_{sh}(I_L + I_o - I)}{a}}\right\} \tag{2.13}$$

These equations are calculated either numerically or by an approximation formula [21]. Using either the Newton-Raphson method or Lambert W function, the current for different values of voltage: $0 \leq V \leq V_{oc}$ is calculated. An example resulting I-V curve is shown in Figure 2.4.

The maximum power point is found at the point on the graph where the product of I and V is a maximum. The expected power output for a PV device at different irradiance and temperature values can be determined using the methods covered in this section. This expected power will serve as the comparison to the measured values in the fault detection procedure. The modelling of PV devices is a well-studied problem and therefore some software solutions have been developed.

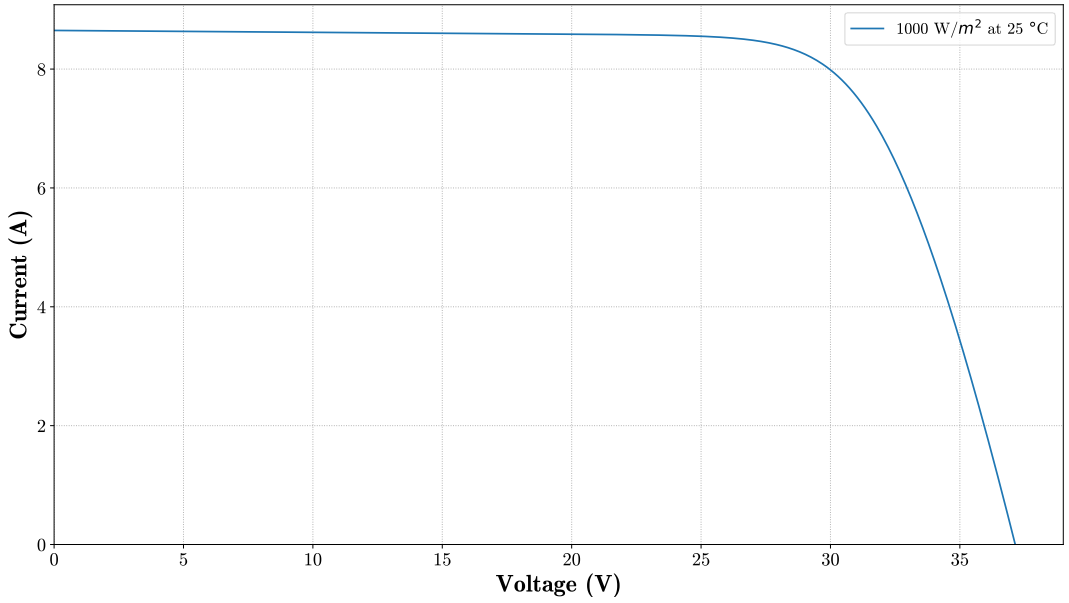


Figure 2.4: Example of I-V curve obtained with standard testing conditions.

2.3 PVLIB-Python

PVLIB-Python is an open source library written in Python that provides most of the functionality of PVLIB-Matlab [25]. The library contains a set of functions that allow users to model and simulate various aspects of a PV system [26]. The PVLIB project was started at Sandia National Laboratories in an attempt to standardise PV modelling functions and analysis methods [26]. Gurupira et al. [27] compared the PVLIB-Python package to the commercially available modelling software PVSyst. It was found that the Python package is a suitable open-source alternative for accurately modelling PV systems. PVLIB-Python is used in this project for modelling the expected power of a string-pair given measured meteorological conditions.

The '*PVSystem*' class contains the attributes associated to the configuration of the PV system being modelled. Attributes like module- and inverter information, tilt- and zenith angles and array configuration are all customisable during initialisation. Various methods contained in the class then allows the user to model the PV system performance. The '*retrieve_sam*' function can be used to retrieve module- and inverter information from CEC and Sandia databases. Segment 2.1 shows the Python code snippet for retrieving module parameters from the CEC database using PVLIB. The 'BYD 240P6-30' module is used in the power plant relating to this project. The reference parameters estimated in Appendix A.1 compare closely to the reference parameters in the CEC module database.

Segment 2.1: Module parameters retrieved from the CEC module database.

```
In [1]: from pvlib import pvsystem

        # CECMod database: BYD_Company_BYD_240P6C_30
        module_list = pvsystem.retrieve_sam(name='CECMod')
        module = module_list['BYD_Company_BYD_240P6C_30']

        module
```

```
Out[1]: BIPV                    N
        Date            9/18/2014
        T_NOCT               47.5
        A_c                 1.627
        N_s                    60
        I_sc_ref             8.65
        V_oc_ref            37.14
        I_mp_ref             8.12
        V_mp_ref            29.57
        alpha_sc         0.004559
        beta_oc          -0.07911
        a_ref              1.4356
        I_L_ref             8.661
        I_o_ref          4.97e-11
        R_s                 0.408
        R_sh_ref           324.26
        Adjust              47.21
        gamma_r            -0.411
        Version            NRELv1
        PTC                 216.6
        Technology     Multi-c-Si
        Name: BYD_Company_BYD_240P6C_30, dtype: object
```

The CEC module database has reference parameters for many popular modules, including the 'BYD 240P6-30' module, predetermined. The '*PVSystem*' class can be initialised to use the 'BYD 240P6-30' module and the array configuration is specified to match the power plant. In this case a string is constructed from 24 modules in series and strings are connected in parallel pairs to form string-pairs. Once the class attributes are specified, the modelling is fairly simple. PVLIB implements parameter estimation based on the work by De Soto et al. as covered in Section 2.2.2. The method '*calcparams_desoto*' accepts plane-of-array irradiance and cell temperature as arguments and returns the estimated parameters for the single diode model. The method '*singlediode*' accepts the five parameters and calculates I-V curves and maximum power point values using an implementation of the Lambert W function. Segment 2.2 shows example code for the PV system configuration of a single string-pair. I-V curves at five different irradiance values are modelled which results in Figure 2.5.

Using the '*PVSystem*' class and the modelling methods mentioned, the estimated current is obtained for a string-pair at given irradiance and temperature values. An accurate prediction of expected current is needed in the fault detection algorithm covered later in the project. Comparing the modelled- and measured current can give insight into the accuracy of the model. The current for a string-pair connected to inverter one will be used as reference. Using the

Segment 2.2: PV system configuration and I-V curve modelling using PVLIB.

```
In [1]: # Desoto
system = pvsystem.PVSystem(module_parameters=module, modules_per_string=24,
                           strings_per_inverter=2)

irradiance = [200, 400, 600, 800, 1000]
temp_cell = [25, 25, 25, 25, 25]

# calculates parameters for module with de soto method
I_L, Io, R_s, R_sh, nNsVth = system.calcparams_desoto(pd.Series(irradiance),
                                                      pd.Series(temperature))

single_diode_out = pvsystem.singlediode(I_L, Io, R_s, R_sh,
                                        nNsVth, ivcurve_pnts=100)
```

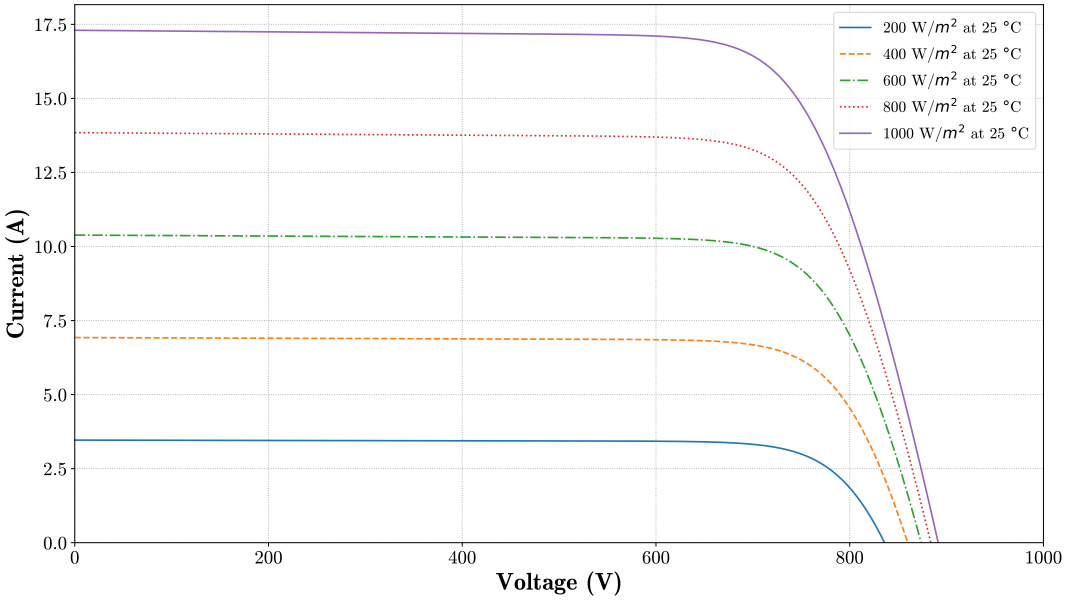


Figure 2.5: Resulting modelled array I-V curves at varying irradiance values.

in-plane irradiance and module temperature at a weather station close to the inverter, the '*calcparams_ desoto*' and '*singlediode*' methods are used to calculate expected current. Figure 2.6 shows the measured- and expected current for string-pair one connected to inverter one for two different days with distinctive cloud cover intensities. The PVLIB modelled current accurately resembles the measured current which boasts confidence in the modelling approach.

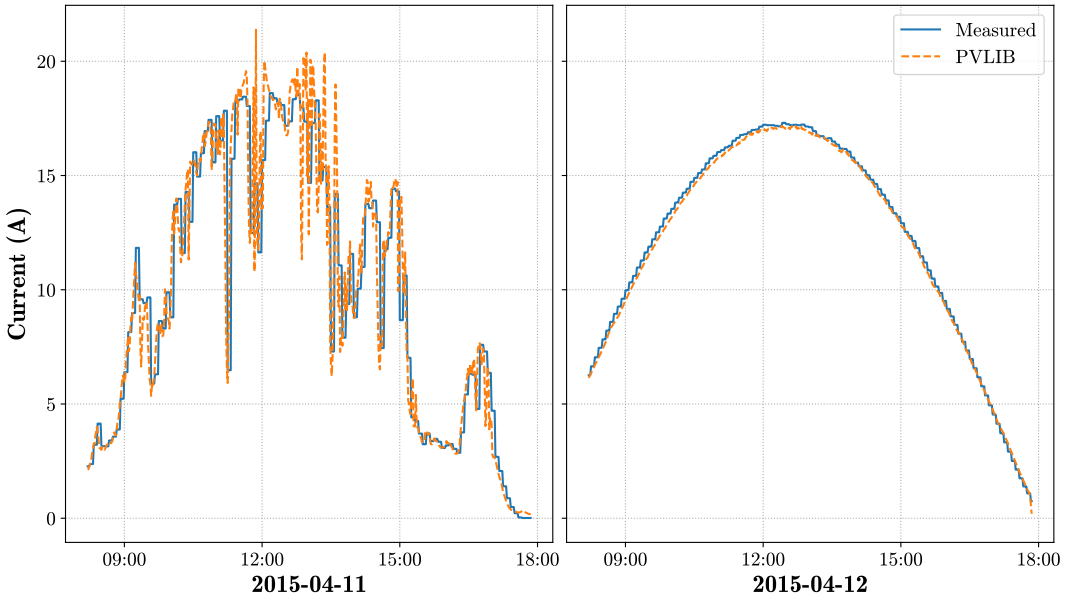


Figure 2.6: Comparison of measured and modelled current output for a single sting-pair connected to inverter one.

The accuracy of a model is often quantified by using the root mean square error (RMSE). This error corresponds to how far the predicted values are from the observed values. A low RMSE indicates that the modelled values lies close to the measured value. The RMSE is calculated using Equation 2.14 where m is the measured value, p is the predicted value and N is the number of compared samples. The RMSE value is not really meaningful unless it is compared to another modelling method.

$$RMSE = \sqrt{\frac{\sum_{i=0}^{N}(m_i - p_i)^2}{N}} \tag{2.14}$$

Segment 2.3: Calculating the RMSE of 2015 for string-pair one connected to inverter one.

```
In [2]: import numpy as np
        from sklearn.metrics import mean_squared_error

        # root mean squared error
        rmse = np.sqrt(mean_squared_error(data['measured'], data['expected']))
        rmse

Out[2]: 1.9274583641121974
```

The parametric model mentioned in [16] is used as comparison. This model defines an equation for the output power of a PV device given in-plane irradiance and module temperature. Since power and current are strongly correlated, in this case, the model is used to calculate expected current. See Appendix A.2

for the implementation of this model. Figure 2.7 shows the comparison of the PVLIB and parametric model. Both modelling methods produce output values that lie close to the measured current values. The RMSE is calculated for 2015 on the modelled current for the same string-pair shown in Figure 2.6. The code segment for obtaining the resulting error is shown in Segment 2.3. The error of PVLIB is compared to the parametric model and shown in Table 2.1. The RMSE error for both models are similar, with the parametric model only slightly smaller. The modelling accuracy of PVLIB is confirmed and deemed suitable for the use case in this project. The PVLIB model is also preferred since the parametric model needs to be trained on historic data for each string-pair, where the electrical model is consistent for all string-pairs in the power plant.

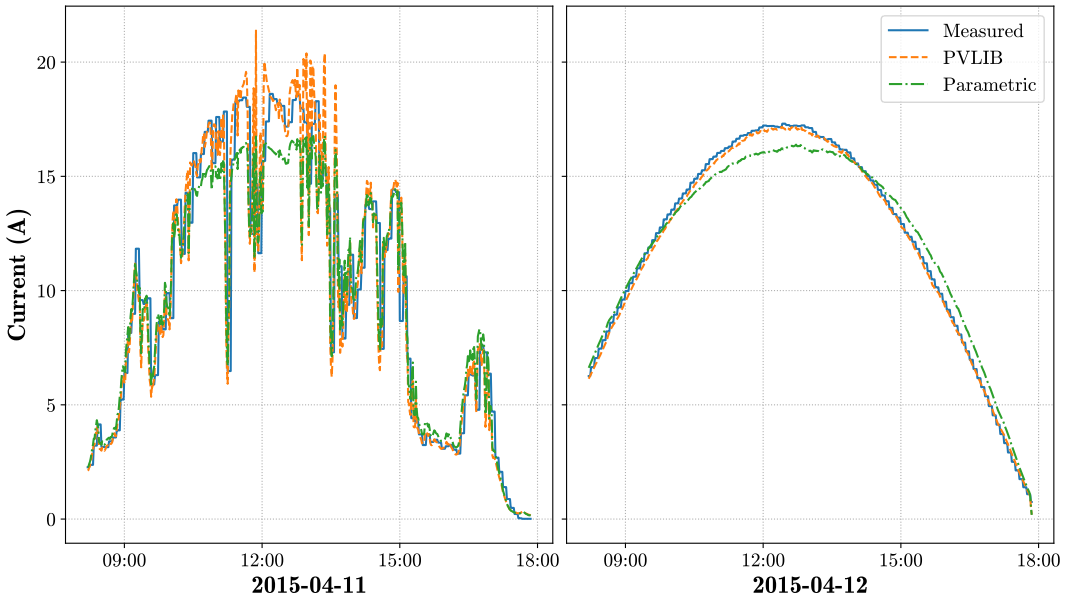


Figure 2.7: Comparison of measured, PVLIB- and parametric modelled current for a single sting-pair connected to inverter one.

Table 2.1: Table showing RMSE comparison of PVLIB and the parametric model for 2015.

Modelling Method	RMSE
PVLIB	1.91
Parametric	1.83

2.4 Summary

Chapter 2 covers the building blocks of a PV system and methods for modelling PV devices. A typical PV power plant includes solar module arrays, power converters and a monitoring system. Details regarding modelling the power output of a PV device given irradiance and temperature are discussed. The implementation of a single diode electrical model in Python is reviewed. Using the software package, the expected current is compared to the measured current for a single string-pair. As a verification of accuracy, the modelled current is tested against a parametric model proposed in literature. The fundamentals of a PV system and the modelling of arrays in a power plant are used for fault detection in Chapter 4. Comparison between measured and modelled behaviour is the core concept on which the detection procedure is based. Analysis of the measured data used in the present work is explored in the following chapter.

Chapter 3

Data Analysis

Operational data from a 75 MWp capacity solar power plant situated in the Northern Cape is obtained for the research presented in this work. Large amounts of sensor data are generated by the utility-scale PV power plant. The main focus of this project is to investigate the use of sensor data for fault detection and performance evaluation. This chapter provides an overview of the available data and includes the methodology for processing the data. Raw data files are combined into a centralised database, whereafter the data is processed and cleaned. Finally, an analysis of missing and unacceptable data is conducted. This process is essential to gaining knowledge about the working dataset.

3.1 Overview of Available Data

Raw data is recorded by the SCADA system each minute and includes measurements for 12 595 different signals. These measurements range from measured power to environmental conditions. The three main sources of data are the inverters, weather stations and transformer stations. Figure 3.1 shows roughly how the power plant and the data sources are connected.

The 'SMA Sunny Central 800CP XT' inverter installed in the solar power plant is capable of measuring operating data and the data is transferred to the master SCADA system using the MODBUS communication protocol. MODBUS is a popular method used for information transmission over serial lines between electronic devices [29]. Each inverter receives DC current measurements from 10 'Sunny String-Monitors' (SSM). A string-monitor is a DC sub-distributor to which several strings can be connected in parallel [30]. Each SSM measures the current for up to 8 string-pairs. The inverter also relays operational data, for example measured AC power, through the SCADA system. The inverter controls the power point at which the string-pairs are driven using the on-board SMA maximum power point tracking procedure called OptiTrac. The

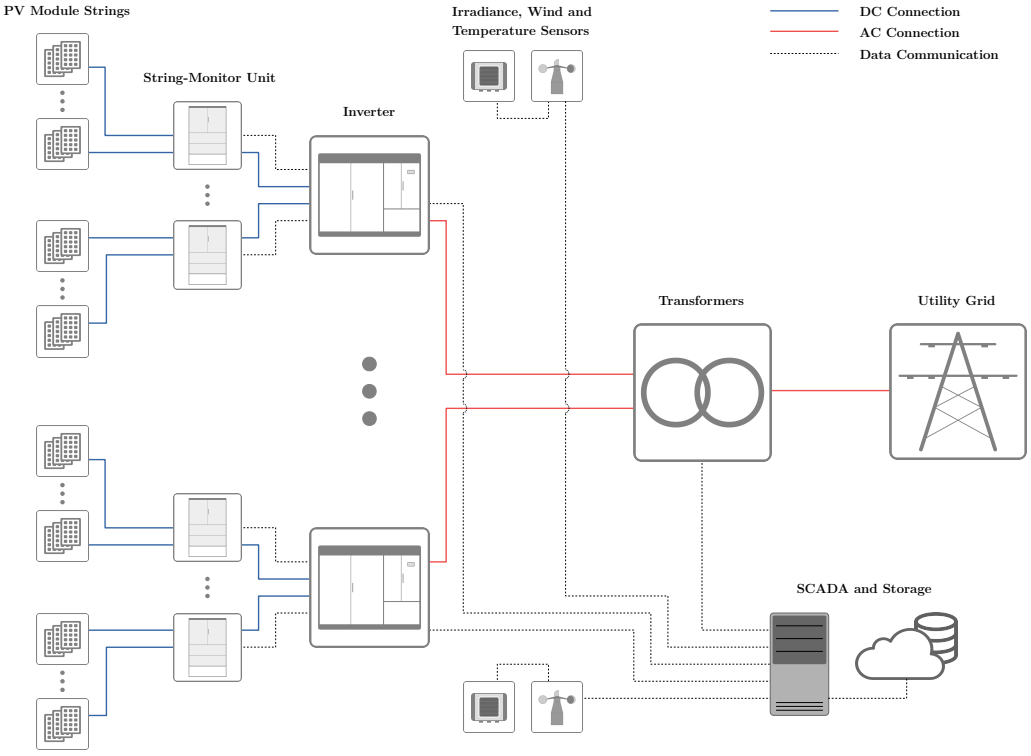


Figure 3.1: Diagram of a typical utility-scale solar PV power plant (adapted from [28]).

measured string-pair current and inverter operating data is used throughout the rest of the project.

Analog sensors are used for measuring active- and reactive energy and power at the medium and high voltage transformers. Sensors at the transducers measure a range of different operating details including line current and voltage, power and frequency. MOXA switches are used to convert the analog measurements to digital when transmitting the data to the SCADA logging system. Meteorological data is recorded at weather stations within the power plant. Table 3.1 shows the number of data sources from the power plant used in the present work.

Table 3.1: Data sources available from the solar power plant investigated.

Data source	Quantity
String-pairs	6511
String-monitors	840
Inverters	84
Medium-voltage transformers	8
High-voltage transformers	4
Transducers	2
Weather stations	5

3.1.1 Weather Data

The power plant under investigation in this project has a total of 5 weather stations. The weather stations measure solar irradiance, temperature and wind speed in real-time. These measurements are recorded at one minute intervals by the SCADA system. One weather station is located at the control building while four other weather stations are positioned throughout the power plant. Table 3.2 and Table 3.3 list the parameters measured at the control building and the four other weather stations. The plane-of-array irradiance and module temperature measurements are used in modelling the expected PV power as detailed in Chapter 2. The control building only measures horizontal irradiance and has no temperature measurement, therefore the in-plane irradiance is calculated using the PVLIB method '*irradiance.poa_horizontal_ratio*'. This method uses the surface tilt, surface azimuth, solar zenith and solar azimuth angles to determine the ratio of the in-plane irradiance to the horizontal irradiance. Weather station 2 is the closest other station to the control building. Therefore, when modelling the power of sections close to the control building, the module temperature measured at weather station 2 is used.

Table 3.2: Table showing weather measurements available from control building.

Measured Parameter	**Unit**
Pyranometer Irradiance Horizontal	W/m^2
Wind Direction	°
Wind Speed	m/s
Absolute Air Pressure	hPa
Rain Intensity	mm/h
Relative Humidity	%

Table 3.3: Table showing weather measurements available from on-site stations.

Measured Parameter	**Unit**
Pyranometer Irradiance Horizontal	W/m^2
Pyranometer Irradiance Incline	W/m^2
Environment Temperature	°C
Module Temperature	°C
Sunny SensorBox Irradiance	W/m^2
Wind Speed	m/s

During the fault detection procedure, the power for each inverter is modelled separately. This step requires the environmental conditions from the weather station closest to the inverter in question. In order to determine which weather station data each inverter should use, the plant layout was studied. The closest weather station for each inverter is identified based on proximity. Figure 3.2 shows the distribution of weather stations for the solar power plant. The location of each weather station is indicated with a different shape and colour. Note that transformer stations (TS), where pairs of inverters are located, are also shown. The inverter-pairs are indicated on the figure with the same shape (slightly smaller) and colour as the nearest weather station.

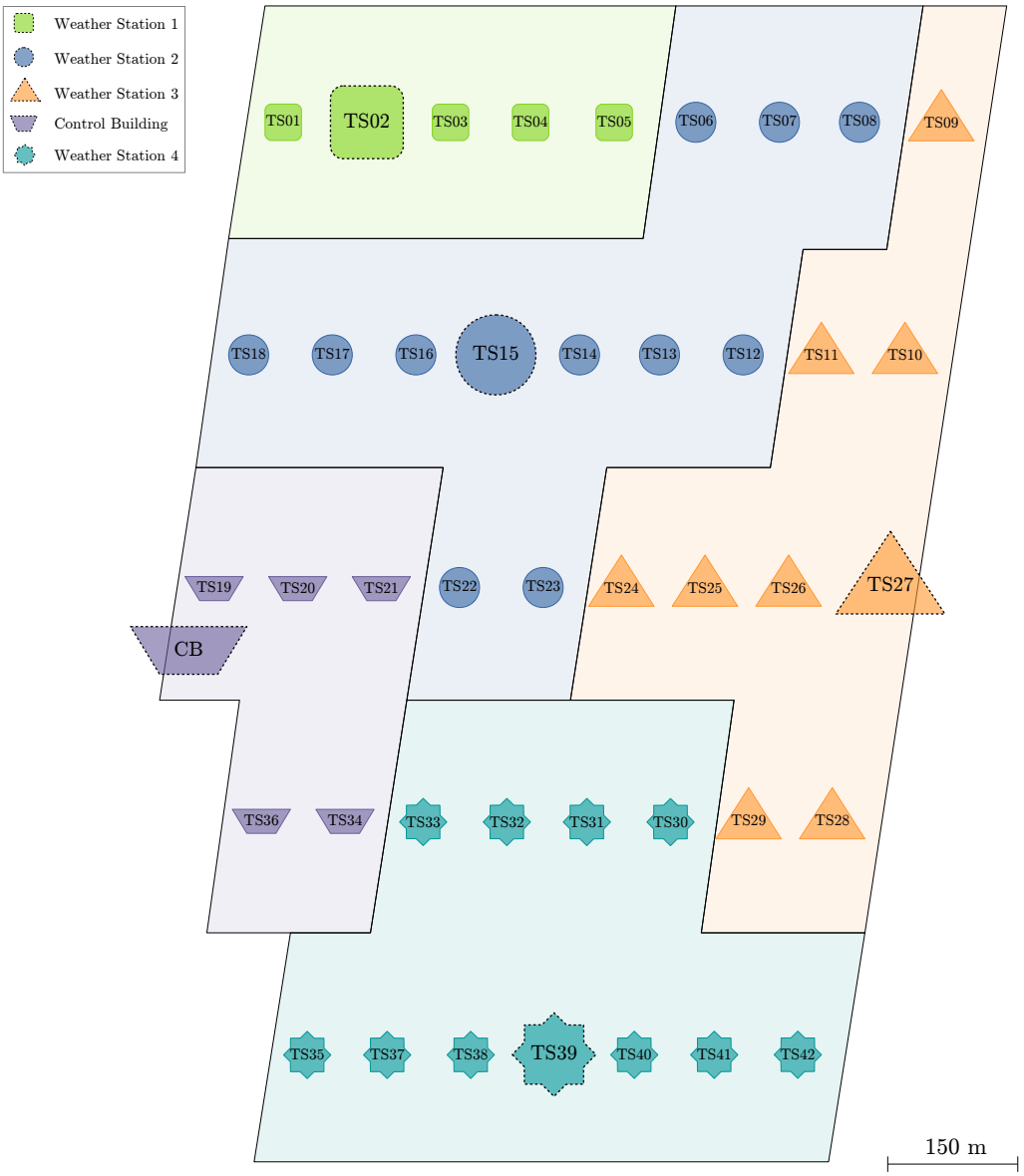


Figure 3.2: Weather stations and corresponding inverters.

3.2 Data Pre-processing

It is clear that a tremendous amount of operating data is generated by the power plant. The one-minute logged data from '2013-09-01' to '2019-05-31' acquired for this project equates to over five years of sensor measurements stored in roughly 560 GB of comma-separated values (CSV) files. The data was acquired in two instances, where the first batch ranges from '2013-09-01' to '2018-03-31' and the second from '2018-01-01' to '2019-05-31'. The raw data is divided into measurements from inverters, weather stations, medium- and

high voltage transformers and transducers. Each data source is further divided into multiple subsections and these subsections are divided into chunks of CSV files each containing data for one month. Since the raw data is split into many separate files, a database could be used to host all the data and mitigate the need for tedious file handling. Processing and cleaning the data is a crucial step in the analysis process. The following section describes the steps taken to make the data accessible.

3.2.1 Database

Organisation of large datasets is an important initial step to data analysis. A database is used to store the large amount of data which also allows the use of powerful queries during data extraction. Structured Query Language (SQL) is used for managing data in a relational database management system (RDBMS). Since the data will be accessed by more than one researcher in the research lab, the database is stored on a network-attached storage (NAS) server. MariaDB is the RDBMS running on the NAS server. SQL statements are used to extract specific columns, rows or values matching a conditional query. A Python script was used to assist in populating the database with the raw data. The script steps through all the CSV files in a directory, creates the necessary tables corresponding to the current file and writes the contents to the table. Figure 3.3 shows a flowchart of the process used. Once all the raw data is written to the database, SQL statements can be used to extract segments of data given specific conditions. The use of a network connected database allows the data to be remotely accessed by multiple researchers, queried and manipulated for use in their respective projects. The raw database uses 565 GB of storage space and executing queries for data extraction can take up to 40 seconds. Cleaning the data and removing redundant values may drastically reduce the size and increase performance.

3.2.2 Cleaning the Data

3.2.2.1 Replacing Missing Data

The first batch of data is formatted slightly different than the second batch. In the first dataset, the SCADA system records an input quality based on the availability of the data. Each data column in the raw database therefore has an associated ‘Quality’ column storing the availability of the sensor reading at a given time. Cases where the sensor data is available and being recorded by the SCADA system, the ‘Quality’ column contains the description ‘calculated:good’. The ‘Quality’ column contains ‘no data:bad’ when the data for a given time is not available. This mainly occurs when an inverter is off-line and the sensors are not recording data or when the data transmission from sensors to the master terminal unit fails. When the data is unavailable, the SCADA

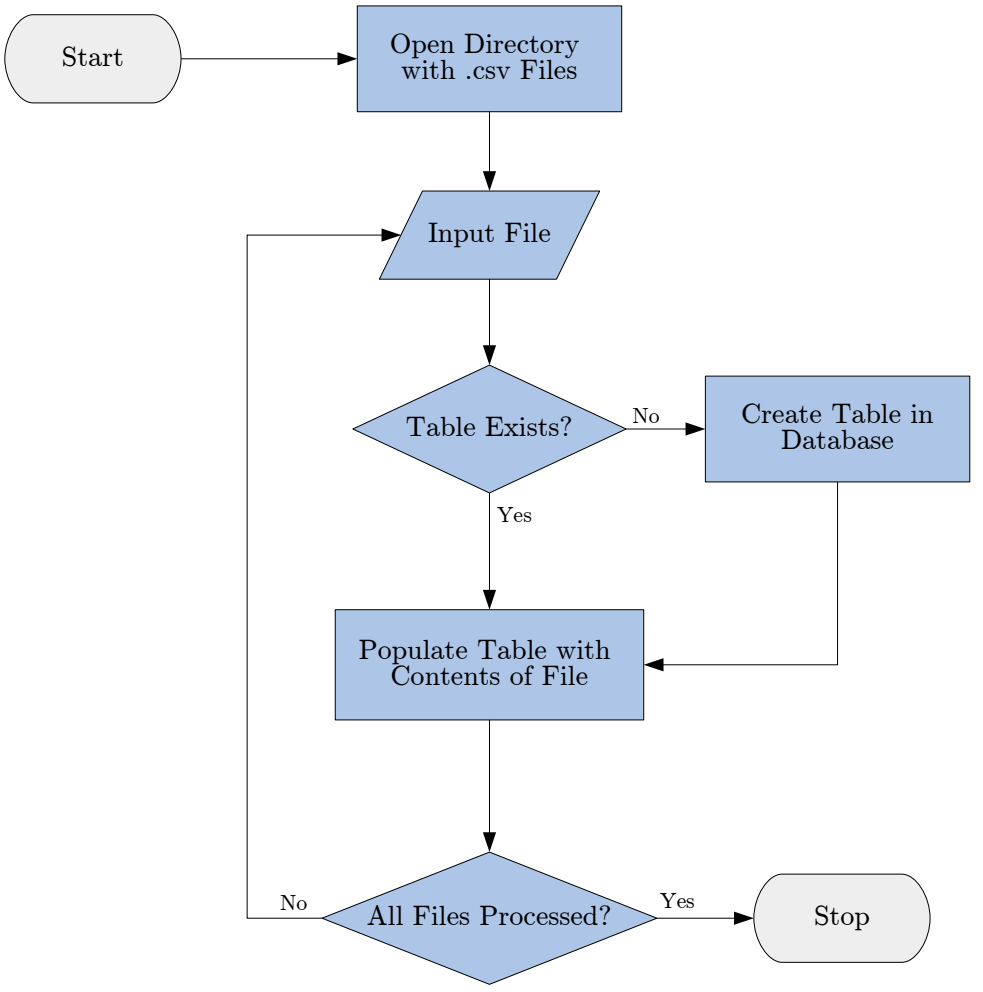


Figure 3.3: Flowchart illustrating the process of populating the database from CSV files.

system records a zero in the data column. Since each value column has a separate 'Quality' column, the size could be drastically reduced if the 'Quality' column is removed. This column can be removed if the value availability is added to the data column. A Python program was developed in order to achieve this. The values in the data column is replaced with NULL when the 'Quality' column contains 'no data:bad', indicating missing data. Since the data column now inherently stores a value when data is available, and NULL when data is missing, the 'Quality' column is removed from the database.

The second batch of data is obtained with interpolation, which is used to replace values where data is missing. This approach removes the need for the 'Quality' column as missing data is interpolated, which greatly saves storage

capacity. However, during fault detection, the interpolation of missing data could cause a decrease in accuracy and lead to false positive detections. Since data could always be interpolated after storage, a better solution might be to store NULL values when no data is received.

3.2.2.2 Replacing Far-out Values

Initial analysis of the data showed that some columns contained outlier values. These data points are values that lie outside of what is considered normal operation from the rest of the data. As an example, a value of 1279400 for measured horizontal irradiance is clearly impossible. Further investigation shows that these far-out values often corresponds to the maximum- or minimum values for a given signal. An example of such a value is either the maximum unsigned 32-bit number (4294967295) or the signed 32-bit numbers (-2147483648 or 2147483647). The occurrence of these numbers indicate measurement error, data overflow or missing values. The solution to handling missing or erroneous data is often application specific. Since the first dataset already contains missing data (NULL values), it is decided that the far-out values should also be replaced by NULL and treated as missing data. In order to remain consistent, far-out values in the second set are also replaced with NULL. Another possible solution for handling missing data is imputation. Imputation refers to the substitution of missing values with a minimum or maximum number, the average of the data range or an estimated value that is calculated using regression or interpolation techniques.

Since the manual detection and replacement of these outliers is infeasible given the size of the dataset, a Python script was developed to assist in the cleaning process. The program extracts minimum, average and maximum values for each column and logs the results to a spreadsheet. The spreadsheet can then be checked for values that are considered extreme outliers (typically overflow). Another function was developed that accepts a column name and value as arguments and replaces each occurrence of the specified value in the column with NULL. Using this method, the dataset could be cleaned much more efficiently than a purely manual approach.

3.2.2.3 Data Pipeline

Receiving a second dataset, necessitates the need for a data pipeline. A data pipeline automates, or at least simplifies, the process of adding new data to the existing dataset. The script developed in Figure 3.3, already automates the population of the database from raw data in CSV files. The data cleaning process is automated by defining minimum and maximum values for each signal. Considering measured environment temperature for example, the minimum and maximum is set to -100 °C and 100 °C respectively. The idea is

not to remove outliers from the data, but rather values that are considered far-out. Therefore, the minimum and maximum limits are set with enough margin to allow normal operating outliers, but all far-out values are removed. The process of defining the allowable ranges for each column is time-consuming and requires system knowledge. However, once the range for each column is defined, when new data is acquired the cleaning process is automated. Firstly, the script which replaces missing data with NULL, if applicable, is executed. Another script then compares the values in each column with the pre-defined minimum and maximum values. If any value in the column falls outside the range, it is replaced with NULL. Figure 3.4 shows the flow of data through the data pipeline. This method allows seamless integration of any new data into the existing database.

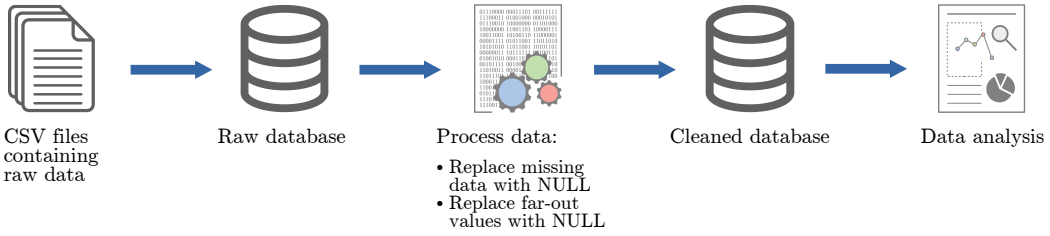


Figure 3.4: Data Pipeline.

The cleaning process drastically reduces the size of the database. Table 3.4 shows the size and performance comparison of the raw- and clean databases. The data pre-processing is deemed successful since the data size is decreased, query performance is increased and the data contains no unacceptable entries.

Table 3.4: Table showing the comparison of the raw- and clean database.

Database	**Size (GB)**	**Query time (s)**
Raw	567.7	46.21
Clean	145.7	17.54

3.3 Missing Data Analysis

Since all missing and far-out values in the clean database are replaced with NULL, an analysis of data availability could quantify the amount of missing data. The SQL statement, shown in Figure 3.5, is used to calculate the percentage of NULL values for each column. A Python program runs this query for each column in the database and logs the results. A slightly altered query

also allows the analysis of missing data per month and year. Figure 3.6 shows the percentage of missing values per year for the data available. The figure shows that the majority of data before '2015-05-01' is NULL values. The reason for the loss of data for the first two years is unknown, but the remaining data is still considered sufficient for analysis.

```
SELECT ((SELECT COUNT(*) FROM I01 WHERE column IS NULL)/(SELECT COUNT(*) FROM I01) * 100) AS percent_missing;
```

Count number of NULL values in column

Count total number of rows in table

Returns number of NULL values as a percentage of total entries in column

Figure 3.5: Example SQL statement to calculate percentage of missing data per column.

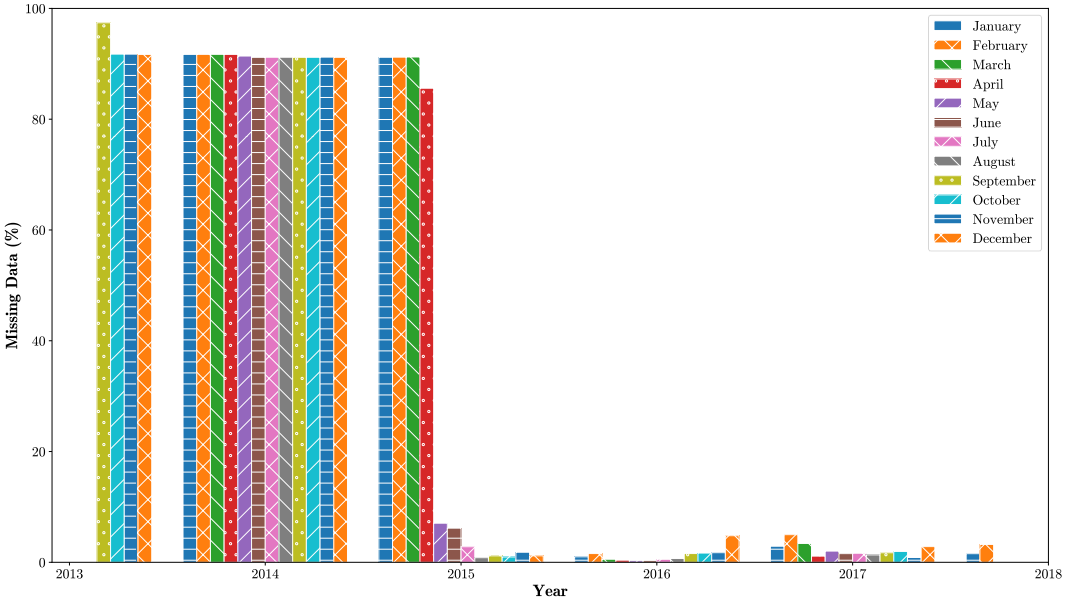


Figure 3.6: Bar graph showing percentage of missing values per year.

The missing data statistics are used to determine the data availability per inverter after '2015-05-01'. Figure 3.7 shows the analysis of missing data for each inverter considering days after April 2015. Three inverters clearly show a higher percentage of missing data. Inverter 33 and 34 are known to have been damaged by lighting and was decommissioned for three months. Inverter 60 also shows a high percentage of missing data. Communication failure from the string-monitor connected to the inverter is the most likely cause, since a high percentage of string-pair current measurements was found to be missing.

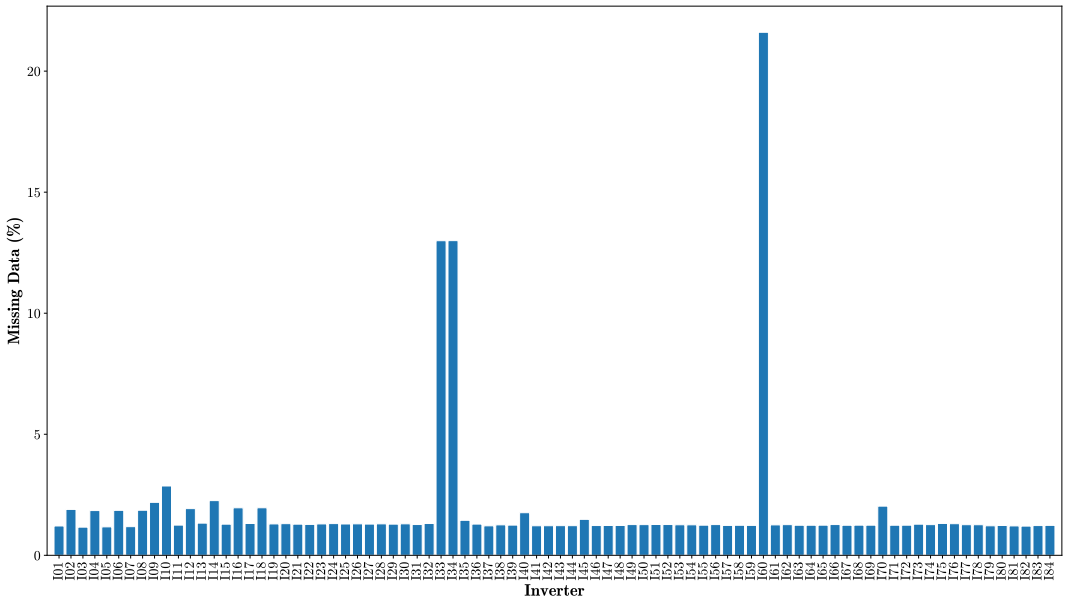


Figure 3.7: Bar graph showing percentage of missing values per inverter for data after April 2015.

3.4 Summary

Chapter 3 provides analysis of the available data and techniques used for processing the raw data into a workable dataset. Raw data from 12 595 measured signals in the power plant is obtained as multiple sets of CSV files. The 560 GBs of data is written to a centralised, network connected database for further processing. Redundant columns and far-out values in the dataset were replaced, which drastically reduced the size. An automated processing procedure is proposed and allows simple integration of future data into the existing dataset. The size and query execution time is reduced by a factor of three as a result of data processing. Analysis of missing data shows that the first one and a half years of data is unavailable. A substantial amount of the data for inverter 33, 34 and 60 after this period is also missing. The dataset used for the rest of the research project ranges from '2015-05-01' to '2018-03-31'. The second dataset was obtained after most of the research was already conducted and is thus excluded from the analysis. The processed dataset used in the rest of the work equates to almost three years of operational data. This is considered sufficient for conveying the results of the procedures and investigations covered in the following chapters. Measured data can be extracted from the database using SQL queries and is used during the fault detection procedure described in the next chapter.

Chapter 4

Fault Detection

Faults in a PV system (stand-alone or grid connected) can affect the efficiency, energy yield and reliability of the plant. Persistent or undetected faults could also lead to safety hazards and a risk of fire [9]. A study conducted on two test PV systems reported annual energy losses, due to various faults, of up to 18.9% [31]. Accurate fault detection in PV systems allow operations teams to perform corrective measures promptly [16]. Quick response times prevent prolonged under-performance and power losses of the system. Overall system performance and reliability can be greatly improved if faults are automatically detected. Literature regarding common faults, fault detection and fault diagnosis is reviewed in this chapter. A fault detection procedure based on the comparison of measured- and modelled string-pair current is then proposed. Implementation of the algorithm is discussed, whereafter the detection procedure is tested on the dataset described in Chapter 3. The chapter concludes with the results of the utility-scale fault detection procedure.

4.1 Common Photovoltaic Power System Faults

PV system faults can be defined as either temporary or permanent. Temporary faults are often caused by shading effects and module soiling. Permanent module faults include: delamination, bubbles, yellowing of cells, scratches and burnt cells [9]. Permanent faults are eliminated by replacing or repairing the damaged modules. Serious failures of PV arrays include line-to-line, ground- and arc faults [9]. Other factors that could lead to production losses include maximum power point (MPP) tracking error, electrical disconnection, wiring losses and faulty equipment [16]. Faults in a PV system can be classified as module-, string- or array faults depending on which component of the PV system is affected. Some of the common faults that occur in a PV system are mentioned below. Madeti et al. [12], provides an extensive table listing PV system faults and descriptions.

4.1.1 Hot-spot Fault

Module hot-spots can occur when individual cells are shadowed or broken. These cells produce considerably less current than other cells in the string and can become reverse biased, which leads to power dissipation instead of generation [32]. This phenomenon affects crystalline silicon (c-Si) cells and is usually a result of soiling, shading, damaged cells or damaged bypass diodes. The hot-spot cells dissipate power which causes the surface temperature to increase and therefore hot-spot faults are diagnosed predominantly using infrared and thermal analysis [9]. Persistent hot-spots can cause damage to the solar cells, bypass diodes and lead to open-circuit faults [9].

4.1.2 Degradation

Module degradation results in lower power output over time. Degradation faults could be identified by voltage and current patterns that are above threshold limits, but still significantly less than expected. DC degradation is covered in more detail in Chapter 5.2.

4.1.3 Partial Shading

Shading faults are caused when a subsection of a module or PV system receives less solar irradiation due to obstruction and shadows. Shading can be diagnosed by looking for unexpected drops in current [19]. A shading event yields similar results to open-circuit strings, but are most often temporary.

4.1.4 Open-Circuit Fault

Open-circuit faults refer to disconnection-faults in PV subsystems. This includes disconnection of cells in a module, modules in a string or strings in a PV array [19]. Diagnosis on an array level can be achieved by inspecting voltage and current indicators. The PV array voltage stays constant; however, the fault results in current decline. Open-circuit faults can be caused by cell damage, faulty diodes and wiring defects among other factors [9].

4.1.5 Short-Circuit Fault

Similar to open-circuit faults, short-circuit faults can occur in different subsystems of the PV plant. Short-circuited modules in a string will result in a significant drop in array voltage, while the array current increases slightly [19]. The same effect is seen when a short-circuit occurs between two strings in an array. An experimental study in [19] shows that short-circuit faults between modules are more detrimental to the system output voltage than short-circuits between strings.

4.1.6 Ground Fault

Ground faults are considered the most common fault in PV systems. The fault refers to the accidental electrical short-circuit between a current-carrying conductor and ground [9]. This fault is predominantly caused by insulation failure of cables. Unresolved ground faults could pose serious safety hazards from DC arcs generated at the fault point, electric shock due to live ground connections and increased risk of fire [9]. Modern inverters often include ground fault detection units, but ground faults on the grounded terminal cannot be detected reliably [33].

4.1.7 Arc Fault

The unintended flow of current through air or another dielectric is known as an arc fault [9]. Arc faults can occur between a discontinuity in an electrical conductor and between conductors with different potentials [9]. Electrical arcs in a PV system could lead to serious fire hazards.

4.1.8 Line-to-line Fault

A line-to-line fault refers to a short-circuit fault between conductors in the PV system. Line-to-line faults can be caused by insulation failure of cables and mechanical damage [9].

4.2 Fault Detection and Diagnosis Methods in Literature

Numerous monitoring and fault detection methods have been proposed in literature. Methods vary in detection speed, implementation complexity, sensor requirements and capability of fault identification and location [9]. Fault detection methods are used to identify abnormal system behaviour, whereas fault diagnosis methods specifically aim to identify different types of faults and the location of faults within in the PV system. Successful fault diagnosis is highly dependent on the measured data that is available. Existing fault detection and diagnosis methods include visual inspection and image analysis, electrical signal analysis, artificial intelligence algorithms and mathematical model analysis methods [19].

4.2.1 Visual and Thermal Analysis Techniques

Visual inspections and analysis of infrared and thermal imaging for fault detection and location are classified as non-electrical methods. These methods

do not require measured electrical data from the PV system. Visual and thermal methods are used specifically to detect discolouration, browning, surface soiling, hot-spots, breaking and delamination of PV modules [9]. Visual inspection of modules is time-consuming and unsuitable for large-scale PV systems. Infrared and thermal fault detection methods usually depend on expensive equipment (thermal or infrared cameras, unmanned aerial vehicle, etc.) and detection speeds are based on the frequency of plant surveillance. These methods have been shown to be successful [34; 35], but are more suitable for small scale PV plants [9].

4.2.2 Electrical Signal Analysis

Signal processing methods are based on the analysis of the PV system response after signal injection. Electrical signal analysis used for fault location include Time Domain Reflectometry (TDR) and Earth Capacitance Measurement (ECM) [9]. These two methods for fault location were studied experimentally in [36]. Results include successful location of disconnected modules with ECM and detection of impedance change due to degradation with TDR [36]. An advantage to using electrical signal analysis is that no measured data is required. These methods are, however, most suitable for small scale PV plants, uses expensive equipment and requires the PV system to be turned off during diagnosis [9].

4.2.3 Statistical Approach

A statistical approach towards AC fault detection is proposed in [37] and experimentally studied on a grid-connected 20 kWp PV system. The procedure is based on ANOVA (Analysis of Variance) and Kruskal-Wallis tests on the measured inverter energy data. The two statistical tests provide box-plots corresponding to the mean and variance of the data for each inverter. Inverter energy data is gathered over a period of three months, whereafter a comparison of the box-plots is used to identify under- and over performing inverters. Statistical methods are useful as they do not rely on weather data or a model of the PV system [37]. The method does, however, result in a slow fault detection rate due to the three month data acquisition period.

4.2.4 Parametric Model Methods

A fault detection and diagnosis method proposed in [15], uses solar irradiance derived from the data of the meteorological satellite Meteosat-8. Based on the satellite derived irradiance the expected yield of a PV system can be simulated using the PVSAT-2 procedure [38]. This method reduces costs by eliminating the need for on-site weather sensors. Low cost hardware is used to measure the actual plant output and the output is compared to the simulated results.

The result from this analysis is compared to a list of predefined error patterns corresponding to possible failures. Using the fault detection method, error patterns for shading effects, degradation, module and string defects, MPPT faults and total blackout can be detected on a daily basis [15; 38]. The automated failure detection routine was studied on 100 small (less than 70 kWp) PV systems across Europe and proved successful. The detection was, however, shown to be highly dependent on the accuracy of the satellite derived irradiance data [15].

An online fault detection approach is developed in [16]. The fault detection is based on a comparison between the measured and expected AC power production [16]. The study was performed on data from a 120 kWp PV plant in Canada. Initial data analysis was used to identify and remove data corresponding to faulty operation. A low complexity parametric model for the expected system AC power production was developed. The model is a variation of similar models presented in [15] and [39] (see Appendix A.2). The model is dependent on the hourly averaged parameters G, the solar irradiance in the module plane and T_m, the module temperature. Operation was considered faulty if the ratio of *measured/modelled* power falls outside three standard deviations of the mean training data ratio. The model achieved a fault detection rate of 81% with a false positive rate of 1%.

4.2.5 Artificial Intelligence Methods

Recent advances in machine learning have lead to various studies into PV fault detection using artificial intelligence techniques. An array level fault detection algorithm using the Takagi-Sugeno-Kahn Fuzzy Rule-Based System (TSK-FRBS) is proposed in [40]. The TSK-FRBS is trained on solar irradiance, temperature and DC power output data from a PV system. The trained model is then used to estimate DC power output for normal operation given measured irradiance and temperature values. The estimated power is compared with the real power and an alarm signal is generated if the difference becomes larger than a specified threshold [40]. The method was implemented in a simulated environment and shows a DC fault detection rate of more than 90%.

Mekki et al. introduces an Artificial Neural Network (ANN) based model for detecting shading losses in a PV module [17]. The ANN is used to model the expected voltage and current for a PV module. The network was trained on solar irradiance, module temperature and corresponding module voltage and current data. Using external meteorological data, the model predicts the PV module performance. The simple error ($E_i = |y_i - y_i^*|$) and mean of squared errors ($MSE(n) = \frac{\sum_{i=1}^{i=n}(y_i - y_i^*)^2}{n}$) are used to detect whether a PV module is

shaded. The method is experimentally studied on a single PV module and results show effective detection of power losses due to shading.

Another use of artificial neural networks is shown in [41]. The authors propose using I-V characteristic analysis for fault detection and an ANN for fault diagnosis. Simulated I-V characteristics are obtained from a Simulink model based on values of solar irradiance and module temperature [41]. The difference between simulated and measured power is compared to a threshold set by calculating the model uncertainty. If faulty operation is detected, a second algorithm implementing an ANN is used to distinguish between eight different predefined faults. Voltage and current indicators based on simulated and measured I-V curves are used as inputs for the neural network. The method is studied in a simulated Matlab/Simulink environment and results show successful detection and identification of faults in a PV array.

4.2.6 Voltage and Current Characteristics Analysis

Similar to [41], the use of voltage and current characteristics for fault detection is presented in [19]. The research investigates faults relating to open circuit, short circuit, partial shading and degradation of a PV array in the PV system. Voltage and current is simulated using the single diode model of a solar cell [20]. Indicators for voltage and current based on ratios of simulated and measured values (similar to [18]) are defined and thresholds for normal operation are calculated [19]. The DC faults under investigation can be diagnosed by comparing the different indicators to threshold values. This method is validated in a simulated Matlab/Simulink environment.

4.2.7 Power Loss Analysis

A fault detection and diagnosis technique is developed in [18]. This method uses solar irradiance and module temperature data to predict the PV array output with an equivalent circuit model [14]. Two indicators corresponding to thermal and miscellaneous power losses are defined. The detection algorithm recognises faulty system operation when the measured losses are much greater than the simulated losses. Indicators based on ratios of the simulated and measured current and voltage values are then used to identify either string defects or shading faults [18]. The proposed method is tested using operational data from a 3.2 kWp grid-connected PV system.

4.3 Implementation of Fault Detection

Various methods of fault detection and diagnosis for PV systems are proposed in literature. In this project, a fault detection algorithm is proposed

that combines some of the ideas in literature; however, the method is applied on a much larger scale. The main idea of the detection algorithm is to compare the expected- and measured behaviour of the PV system. This concept is used in [16], [17], [18], [19] and [40]. The main difference between works in literature is the aspect of the PV system that is compared. Modelled power, power losses and voltage- and current characteristics are some of the different aspects that are compared. The method for obtaining the expected behaviour also differs, with authors using neural networks, parametric models or electrical models. This work uses the comparison of measured and modelled DC current at string-pair level. This comparison metric is used because the string-pair current measured by each Sunny String-Monitor is the lowest level measurement available in the dataset. Detecting faults at string-pair level is valuable since the cause of system abnormalities can be located by operations and maintenance teams with less effort. The electrical model for calculating expected current as mentioned in Chapter 2 is used in the proposed fault detection algorithm. Authors, [18] and [19], also use a single diode model for modelling expected power loss or voltage and current characteristics. Most of the fault detection and diagnosis methods in literature that were experimentally studied, was implemented on PV systems with less than 150 kWp output power. The fault detection in this project is applied to an utility-scale PV power plant with a 75 MWp installed capacity. The following sections provide implementation detail of the fault detection algorithm.

4.3.1 Calculating Expected Current at String-pair Level

Chapter 2 covers the electrical single diode theory in detail. The use of PVLIB for modelling PV system output, given measured plane-of-array irradiance and module temperature, is also covered. Using these methods, the expected current for each string-pair in the power plant is modelled. The meteorological data from the weather station closest to the inverter to which the string-pair is connected is used. Chapter 3.1.1 covers the distribution of weather stations throughout the power plant. Since there are only five weather stations, five different modelled string-pair currents can be calculated. All string-pairs connected to inverters that share the closest weather station will have an identical modelled current. During the fault detection process, the modelled string-pair current is only calculated for each inverter which saves on computational cost and time. The expected current is then compared to the measured current of each string-pair connected to the inverter. In order to compare the measured and modelled currents, an indication of similarity is needed.

4.3.2 Time Series Dissimilarity Measures

A time series, or time sequence, is an ordered list of elements, where each element consists of a value and a corresponding timestamp. When two time

series are compared, the terms distance, similarity and dissimilarity are often used. All of these terms, in essence, refers to the degree of resemblance or similarity the two sequences share. Two common time series distance metrics are considered in this work. Even though the number of approaches for dealing with time series similarity is vast, Euclidean distance and dynamic time warping distance is well researched and often used in literature [42].

4.3.2.1 Euclidean Distance

The Euclidean distance is considered to be the simplest way to estimate the dissimilarity between two time series [42]. This metric is known as a lock-step measure, where samples at exactly the same temporal location is compared [42]. Given two ordered sequences x and y, the Euclidean distance is calculated as shown in Equation 4.1.

$$d(x, y) = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2} \,, \tag{4.1}$$

where M is the length of the time series and x_i and y_i are the i-th element in the series x and y respectively. The Euclidean distance is often favoured for its computational simplicity. A consideration when using the Euclidean distance is that both time series have to have the same number of samples. A simple Python implementation for calculating Euclidean distance is shown in Segment 4.1.

Segment 4.1: Simple Python implementation for calculating Euclidean distance.

```
In [1]: import numpy as np

        # Euclidean distance:
        def Euclidean_Distance(x, y):
            return np.sqrt(sum((x - y)**2))
```

4.3.2.2 Dynamic Time Warping Distance

Dynamic Time Warping (DTW) is another classic approach for calculating the dissimilarity between two time series. DTW is known as an elastic dissimilarity measure, where time series are aligned in the temporal domain so that the accumulated cost of the alignment is minimised [42]. This approach is resistant to minor time-shift differences and is able to compare similar patterns in the sequences [43]. In order to calculate the DTW distance, a warping matrix is constructed. Consider $x = [1, ..., n]$ and $y = [1, ..., m]$, two time series being compared. The warping matrix, $D_{n,m}$ is initialised with $D(i, j) = \infty$ for

$i = 0, ...n$ and $j = 0, ..., m$. The first entry is set to $D(0, 0) = 0$. The rest of the entries are calculated by solving

$$D(i, j) = \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} + d(i, j) \ , \tag{4.2}$$

where $d(i, j) = (x_i - y_j)^2$ for $i = 1, ..., n$ and $j = 1, ..., m$. The DTW distance is calculated by taking the square root of the accumulated cost,

$$d_{DTW}(x, y) = \sqrt{D(n, m)} \tag{4.3}$$

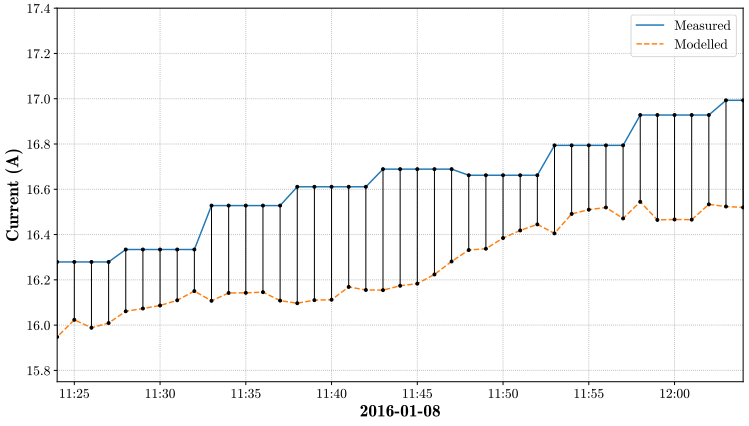
The time complexity for calculating the DTW distance is an order of N × M and can quickly become expensive for long time series. Therefore a locality constraint window $w \in [0, m]$ is often included which decreases the computational time of the algorithm. The constraint limits $j = \max\{1, i-w\}, ..., \min\{m, i+w\}$ in Equation 4.2. The parameter w in essence restricts the temporal search space at each entry of the warping matrix which can lead to a decrease in the computation time. Note that for $w = 0$, the DTW distance corresponds to the Euclidean distance and $w = m$ results in the unconstrained DTW. Figure 4.1 shows the warping matrix for a sample case. The area shaded with blue corresponds to the entries that are calculated if a warping window of $w = 3$ is applied. A Python implementation for calculating the DTW distance is added in Appendix A.3.

			y							
			1.88	**2.78**	**1.22**	**-1.10**	**-1.75**	**-0.10**	**-0.31**	**-1.43**	**-1.18**
		0	*inf*	*inf*	*inf*	*inf*	*inf*	*inf*	*inf*	*inf*	*inf*
	-0.06	*inf*	**3.76**	11.83	13.47	14.55	17.41	17.41	17.47	19.35	20.60
	0.46	*inf*	5.78	**9.14**	**9.72**	12.15	17.03	17.34	17.93	21.04	22.04
	-0.64	*inf*	12.13	17.48	12.60	**9.93**	11.16	11.45	11.56	12.18	12.47
	-2.23	*inf*	29.02	37.23	24.50	11.21	**10.16**	14.70	15.14	12.20	13.28
x	**0.09**	*inf*	32.22	36.26	25.78	12.63	13.55	**10.20**	10.36	12.67	13.81
	0.04	*inf*	35.61	39.73	27.17	13.93	15.83	**10.22**	10.32	12.48	13.97
	-0.30	*inf*	40.36	45.10	29.48	14.57	16.03	10.26	**10.22**	11.50	12.27
	0.90	*inf*	41.32	43.89	29.58	18.57	21.59	11.26	11.68	**15.65**	15.83
	1.74	*inf*	41.34	42.40	29.85	26.64	30.75	14.65	15.46	21.73	**24.18**

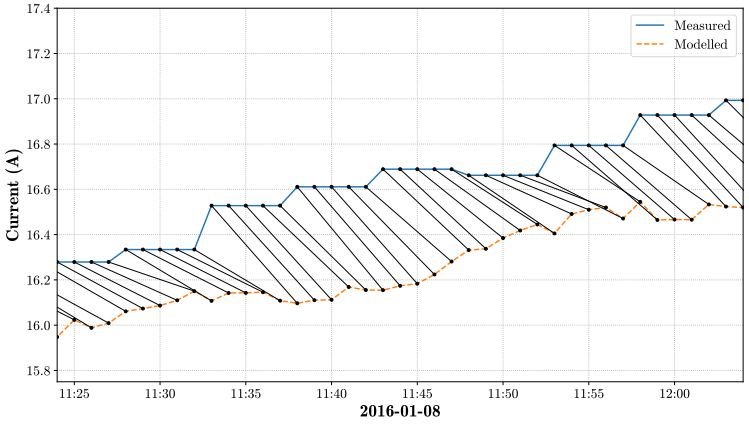
Figure 4.1: The (constrained) warping matrix of x and y [44].

4.3.3 Comparison of Dissimilarity Measures

The two time series distance metrics are compared on sample measured and modelled string-pair current data. Figure 4.2 shows Euclidean distance compared to the DTW distance with a warping window of $w = 6$. The black lines between the measured and modelled curves indicate the temporal indices that are compared. The Euclidean distance method in Figure 4.2a compares samples in the same temporal location, whereas the DTW approach in Figure 4.2b allows warping. Note from Figure 4.2 that the measured string-pair current is recorded using a sample-and-hold method at five minute intervals. Therefore, high frequency changes in the modelled current will not be present in the measured current. The use of either Euclidean distance or DTW distance largely depends on the application. Therefore, a few test cases are used in order to establish which distance metric results in a more accurate comparison of measured and modelled currents.

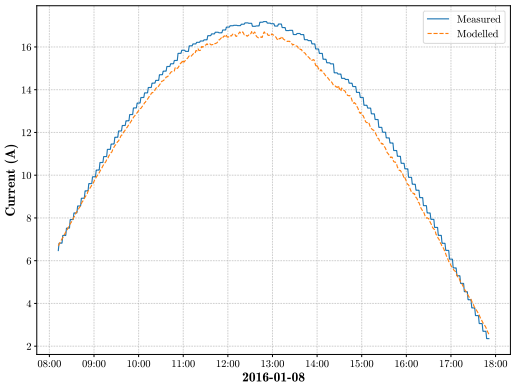


(a) Euclidean distance

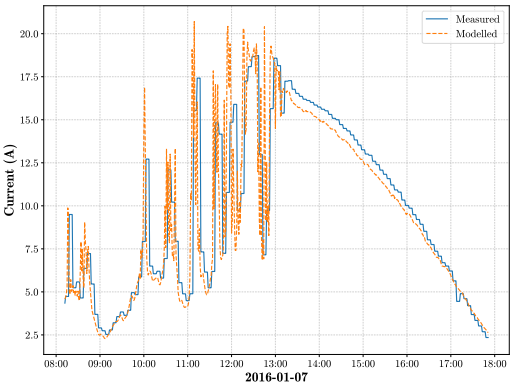


(b) Dynamic time warping distance

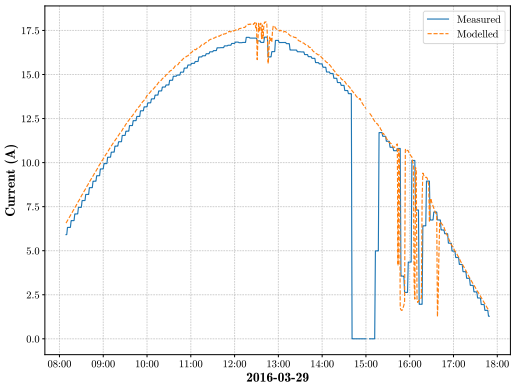
Figure 4.2: Comparison of Euclidean and DTW distance in terms of sample alignment.



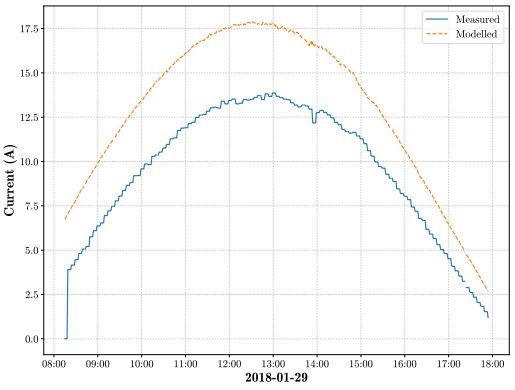
(a) $d_{Euclidean} = 12.03$, $d_{DTW} = 6.67$



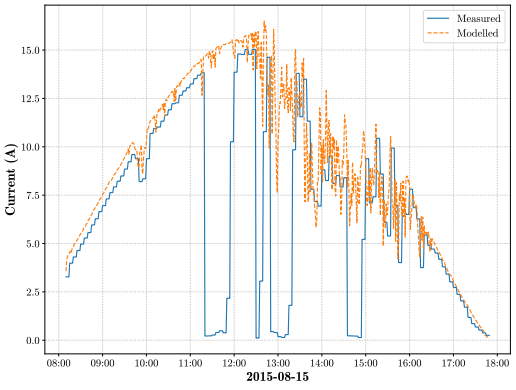
(b) $d_{Euclidean} = 61.03$, $d_{DTW} = 28.20$



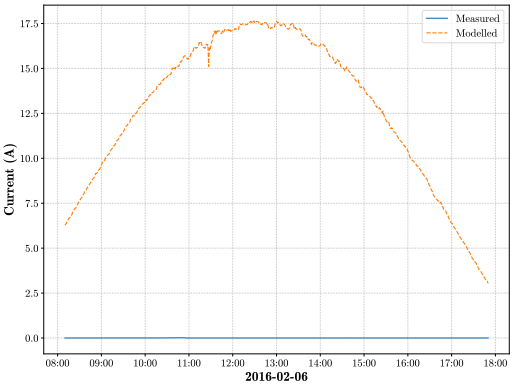
(c) $d_{Euclidean} = 76.35$, $d_{DTW} = 67.45$



(d) $d_{Euclidean} = 83.93$, $d_{DTW} = 76.78$



(e) $d_{Euclidean} = 122.05$, $d_{DTW} = 103.23$



(f) $d_{Euclidean} = 323.01$, $d_{DTW} = 323.01$

Figure 4.3: Comparison of Euclidean and DTW distances for a few test cases.

Figure 4.3a and Figure 4.3b show plots corresponding to normal behaviour. Figures 4.3c to 4.3f corresponds to possible fault cases where the measured current deviates from the expected current. The Euclidean and DTW distances

follow a similar trend, where the distance between measured and modelled is larger for unexpected behaviour. During the fault detection procedure, the distance will be used in order to flag events where the measured string-pair current deviates significantly from the modelled. Therefore, the distance metric which results in the largest separation between normal and fault events will lead to the highest detection accuracy. Note that in Figure 4.3f the Euclidean and DTW distances are equal since no warping that results in a smaller DTW distance is found.

Table 4.1 shows execution times for calculating Euclidean distance, DTW distance with a warping window of $w = 10$ and DTW distance with no locality constraint for a sample data set with x and y each 600 samples long. The Python library '*timeit*' is used to determine the execution times for each distance metric. Since the distance will be calculated for each of the 6511 string-pairs, using the DTW distance without constraint is infeasible. Even with a locality constraint of 10, the time cost of calculating the DTW distance is regarded too high for this application. The distance calculation alone will take up to three minutes for each day being processed. The Euclidean distance is chosen since the separation for normal and faulty operating conditions, although smaller than DTW distance, is acceptable. The Euclidean distance comes at much less computational cost which allows for possible future real-time implementation of the fault detection algorithm. The comparison of measured and expected string-pair currents is efficiently achieved by calculating the Euclidean distance between the two time series. Events where the two time series differ significantly can thus be classified as possible system faults. This concept is the basis for the fault detection procedure.

Table 4.1: Table comparing execution times of calculating Euclidean distance and DTW distance for 6511 repetitions.

Distance metric	**Execution time (s)**
Euclidean	0.82
DTW ($w = 10$)	169.38
DTW	4309.09

4.4 Implementation Considerations

Initial analysis into the data for the string-pair currents uncovers a few considerations that need to be taken into account. String-pair currents have many instances where the data is missing. Figure 4.4 for example shows a day where

the majority of the string-pair current is unavailable. Since no data is available for this period, no comparison to the modelled current can be made. The detection algorithm is therefore dependent on the availability of data.

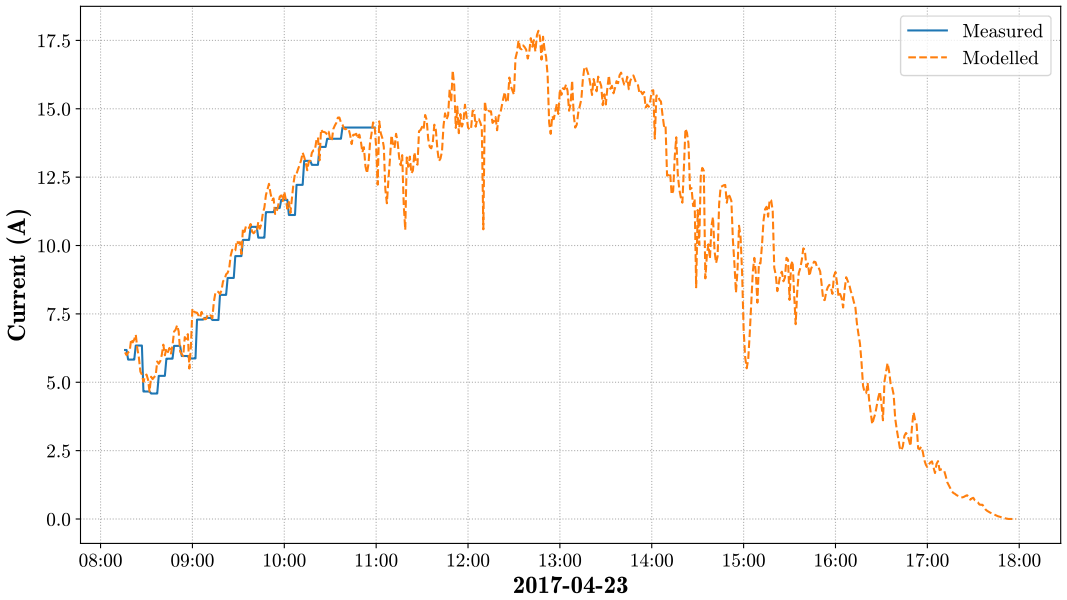


Figure 4.4: String-pair current with missing data.

The current data for string-pair currents are recorded only in certain operating hours. The SCADA system thus records zero current before the logging begins, although current is being produced. These hours seem to differ slightly between different days of the same inverter and also between different inverters. Inverter five consistently starts logging later than the rest of the inverters. Figure 4.5 shows the different logged hours for inverter one and five. In order to address this problem, the average recording hours for each inverter is determined by averaging the first and last recorded times for each day. The average operating hours for inverter one is between '08:12:00' and '17:51:00', while the average operating hours for inverter five is between '08:59:00' and '18:08:00'. During the fault detection procedure, only the values between the average recording times are compared.

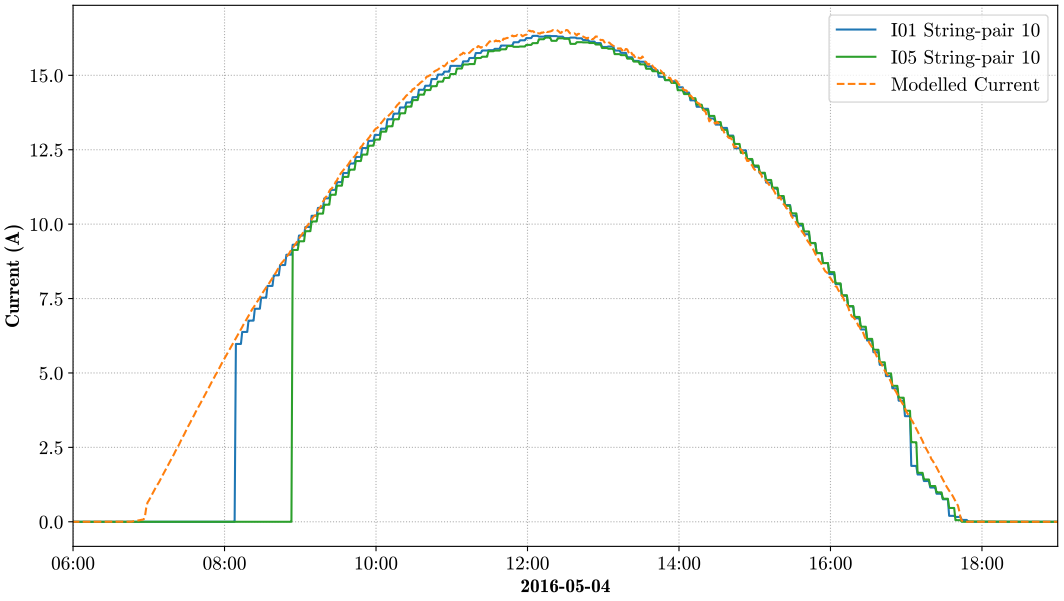


Figure 4.5: Comparison of recording hours between string-pairs.

Any constant deviation between the measured and modelled string-pair current leads to a greater Euclidean distance. Figure 4.6 shows the difference between measured currents for string-pairs connected to the same inverter. It is clear that string-pair 42 will have a greater Euclidean distance due to constant deviation from the modelled current. The string-pairs connected to an inverter are driven at the same power point determined by the MPPT of the inverter. This may cause the variation between string-pair currents, since not all string-pairs necessarily experience the same irradiance and temperature. The threshold added in Section 4.7 aims to address this problem. Constant deviation between measured and modelled string-pair current is handled as false positives during the fault detection procedure. Nevertheless, this knowledge has inspired another research direction comparing the difference of measured values between string-pairs which is reported in Chapter 5.

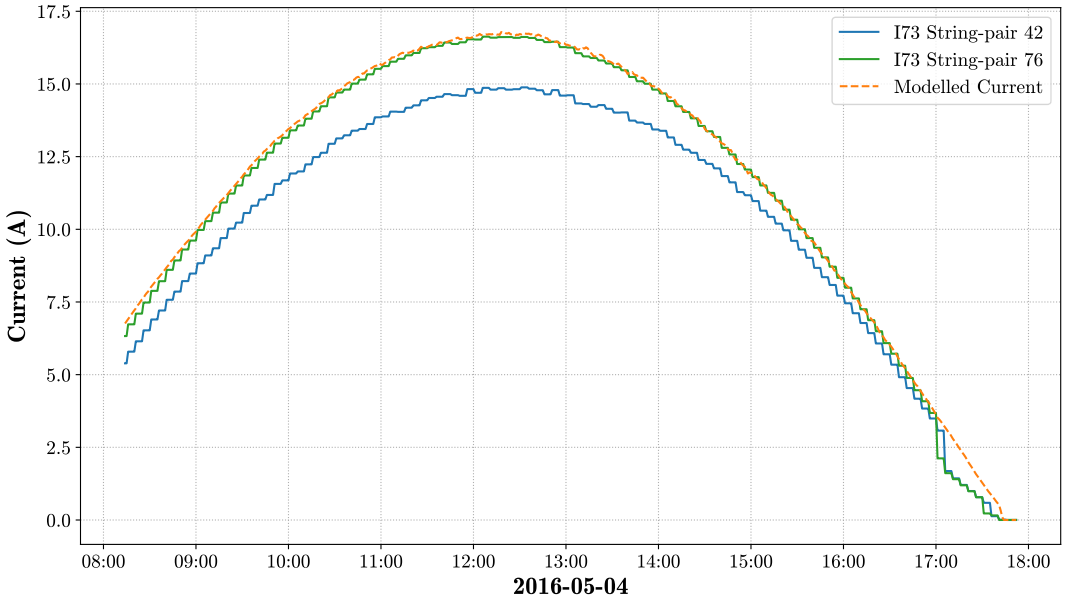


Figure 4.6: Constant deviation between modelled and measured string-pair currents.

These considerations are taken into account when designing the fault detection algorithm proposed in the following section.

4.5 Identifying Events of Unexpected Behaviour

The fault detection procedure involves comparing measured and modelled currents of every string-pair for each day in the available data. Events are flagged if the distance between the measured and modelled values is larger than expected. The usable time period for the dataset ranges from '2015-05-01' to '2018-03-31' resulting in 1066 days. Since each string-pair is compared with the modelled data for an entire day, faults can be detected at the end of each day. The fault detection rate, or window, is therefore considered one day. The comparison window can be narrowed to obtain faster detection rates, but at an increased cost of accuracy and computation. The daily detection rate is deemed acceptable for this project, given the experimental nature of the study. Faster detection rates or real-time implementation is proposed in future work.

The fault detection algorithm handles each day separately and relies on two main parts. Firstly, the Euclidean distances between measured and modelled current for all string-pairs are calculated. Outliers in the list of all string-pair distances are then identified and flagged as possible fault events. Equation 4.4

shows the permissible range of values that are considered normal operating behaviour.

$$\bar{D} - (k\sigma) \leq D_i \leq \bar{D} + (k\sigma) \quad , \tag{4.4}$$

where $\bar{D}$ is the mean of all distances, σ is the standard deviation of the distances and k is a multiplier which determines how wide the permissible range is. A small value for k will result in a more aggressive detection algorithm, whereas a large value for k might miss some fault events. The value for k was empirically solved by adjusting the value until an acceptable detection rate was obtained. The multiplier $k = 5$ is chosen as it minimises false positives while maximising detection rate. Values that lie outside the permissible range in Equation 4.4 are considered faulty. Note that the comparison of all Euclidean distances is used to identifying outliers. Therefore, if the entire plant is in a fault state, the detection algorithm could produce false negative detections. This behaviour has the advantage that instances where the plant is operating below expected, but not in a fault state, false positives are not detected. An example of one such instance is during curtailment. The two parts are combined to form the fault detection procedure. Algorithm 4.1 shows the simplified pseudocode for fault detection. Note that Section 4.7 shows a possible improvement to the fault detection algorithm by implementing a threshold which is also added in the pseudocode.

Algorithm 4.1 String-Pair Fault Detection Pseudocode

```
get plant layout: inverters, monitors and strings
get all days between start date and end date
FOR each day in all days DO
    FOR each inverter in all inverters DO
        get irradiance and temperature from closest weather station
        get expected current with irradiance and temperature
        FOR each monitor in monitors connected to inverter DO
            FOR each string in strings connected to monitor DO
                get measured current for string
                get Euclidean distance between measured current and
                    expected current
                add distance and corresponding string to
                    list of distances for day
            END FOR
        END FOR
    END FOR
    get mean and deviation for list of distances
    FOR each string and distance in list of distances DO
        IF distance lies outside five times deviation from mean THEN
            IF distance greater than threshold THEN
                add fault for string on day to list of faults
            END IF
        END IF
    END FOR
END FOR
```

4.6 Fault Events Detected with the Algorithm

The fault detection algorithm is implemented on the first dataset where 1066 days are evaluated. Figure 4.7 shows the distribution of events that are identified as abnormal. The x-axis in Figure 4.7 shows the Euclidean distance, while the y-axis reflects the number of events flagged for the given distance.

A total of 45 040 events are identified as possible faults. This number is significant and emphasises the need for efficient fault detection and correction. Given the size of the dataset, not all flagged events can be investigated. In order to establish the detection accuracy, a sub-sample of 1200 random events are investigated. The term 'fault' can be interpreted in various ways, but any behaviour that indicates clear unexpected contrast to the modelled current is considered a fault. Faults and false positives in the sub-sample are visually verified and labelled accordingly. Figures 4.8 and 4.9 shows some of the events that are flagged as outliers. Figure 4.8a shows a small Euclidean distance, but

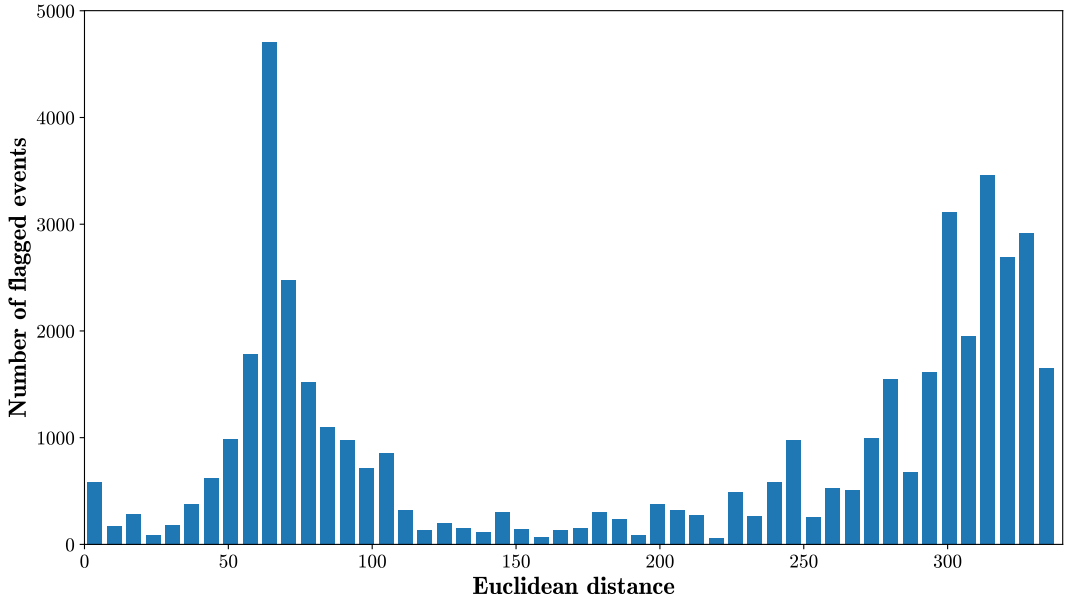
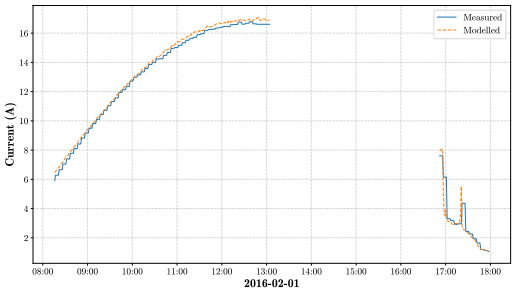
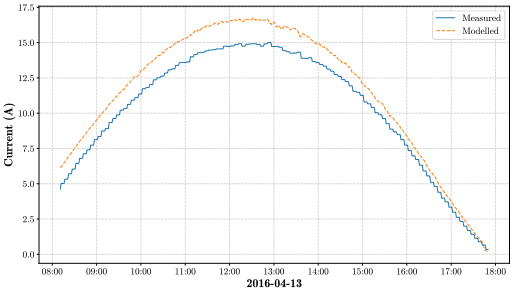


Figure 4.7: Distribution of distances for events flagged by the detection algorithm.

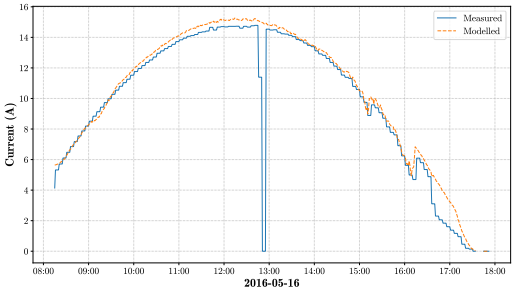
this string-pair was still considered far from the mean distance for the particular day. Missing data could contribute to this false positive identification. Figures 4.8b and 4.8f shows string-pairs that are flagged by the detection procedure due to constant deviation from modelled current. These cases, however, are regarded as normal operation and are therefore labelled as false positives. Figure 4.8e shows another string-pair flagged as an outlier. The difference between the average start time and actual start time in this case leads to a higher Euclidean distance and subsequently the string-pair is flagged. It can be argued that these "false" detections are still valuable since the measured current differs from what is expected. However, for this application, these cases are considered within the permissible normal operating range. The rest of the figures in 4.8 and 4.9 shows string-pairs that were successfully flagged as faulty. A clear trend can be seen where the Euclidean distance is greater for events where measured current deviates more from the modelled current.



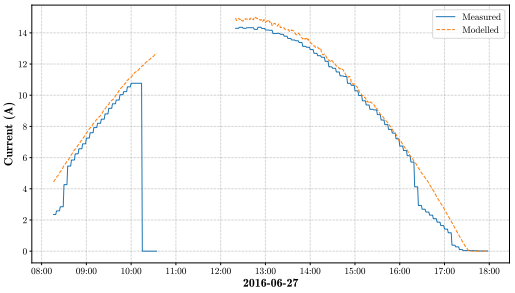
(a) False positive,
$d_{Euclidean} = 8.22$



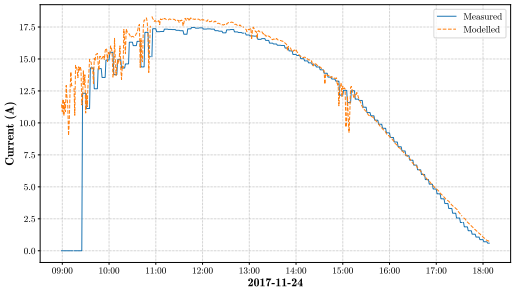
(b) False positive,
$d_{Euclidean} = 32.17$



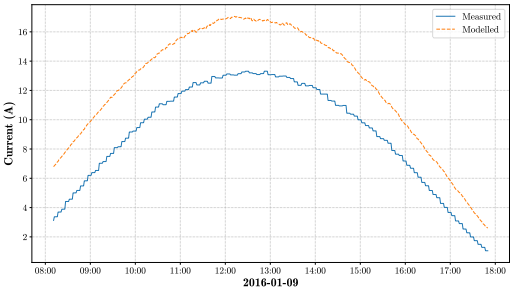
(c) True positive,
$d_{Euclidean} = 37.13$



(d) True positive,
$d_{Euclidean} = 57.65$

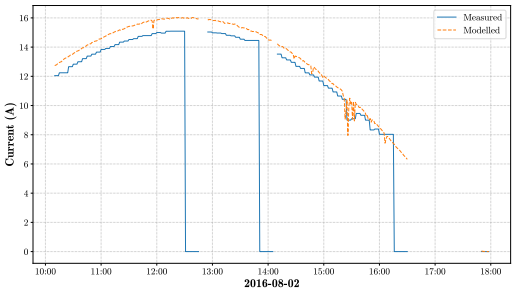


(e) False positive,
$d_{Euclidean} = 66.41$

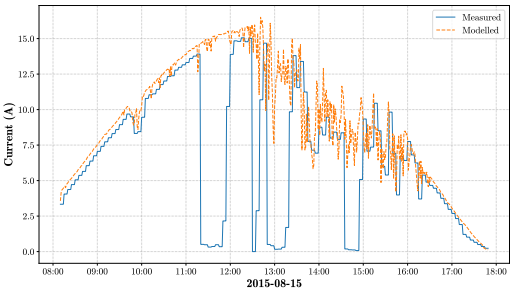


(f) False positive,
$d_{Euclidean} = 80.86$

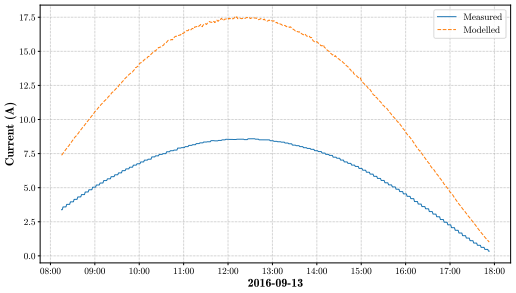
Figure 4.8: Some of the events, with corresponding labels, flagged by the fault detection algorithm.



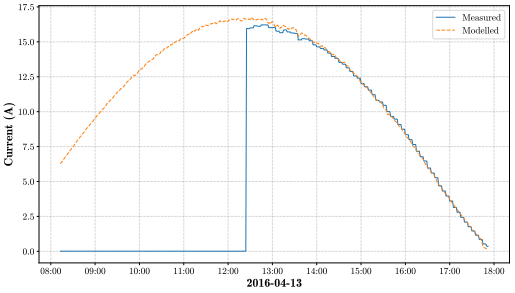
(a) True positive, $d_{Euclidean} = 89.06$



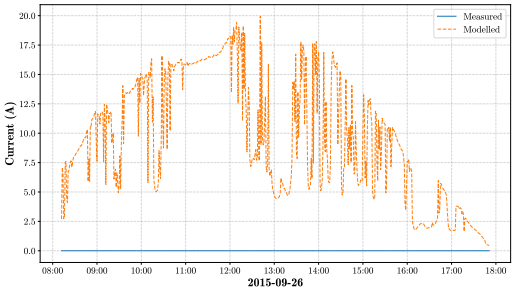
(b) True positive, $d_{Euclidean} = 125.59$



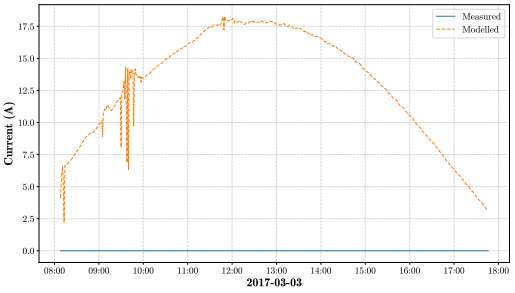
(c) True positive, $d_{Euclidean} = 163.20$



(d) True positive, $d_{Euclidean} = 212.57$



(e) True positive, $d_{Euclidean} = 262.68$



(f) True positive, $d_{Euclidean} = 330.22$

Figure 4.9: Some of the events, with corresponding labels, flagged by the fault detection algorithm (continued).

4.7 Improving the Detection Algorithm

1200 events were labelled and the resulting accuracy and false positive detection rate for the sub-sample is 88.5% and 11.5% respectively. The false positive rate is considered too high and in order to improve the algorithm, the sub-sample is further analysed. Figure 4.10 shows the normal distribution for true positives and false positives in the sub-sample of flagged events. Notice that the mean Euclidean distance of false positives is much lower than the mean of true positives. Since false positives are more likely to have a smaller

Euclidean distance, a threshold can be used to limit the flagged events to distances greater than a given value. The value for the threshold is determined by calculating the effect on accuracy each threshold value has. Values between 0 and the intersection point of the the two distribution curves (roughly 76) are tested. The threshold value of 44 is determined to maximise the accuracy (true positives and true negatives) while minimising false positives and false negatives.

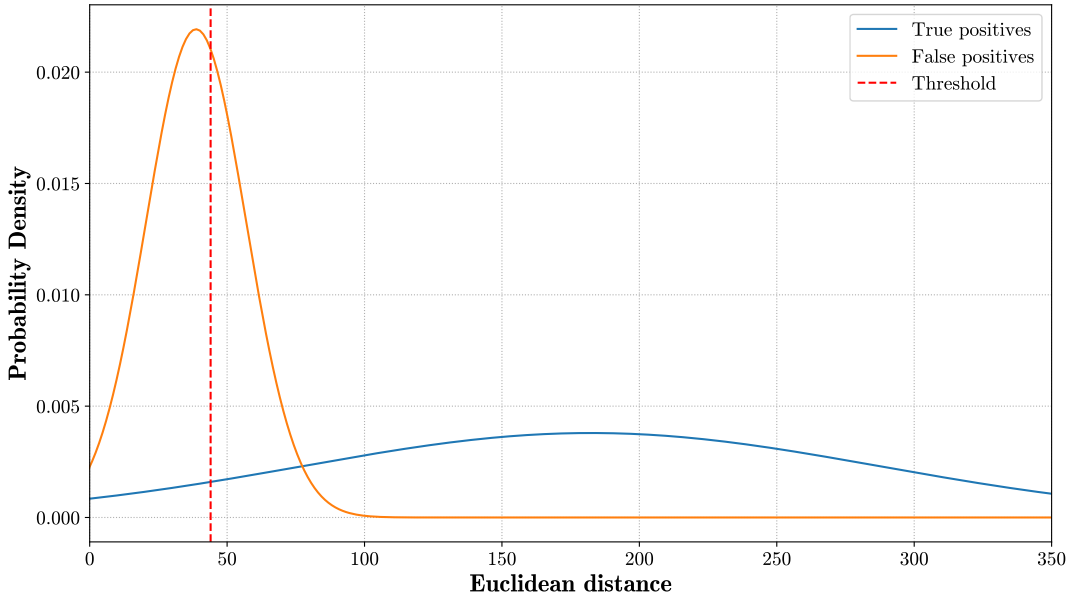


Figure 4.10: Normal distribution of true positives and false positives in the sub-sample of flagged events.

Implementing the threshold on the sub-sample of flagged events results in an improved fault detection accuracy of 94.67%. The false positive rate is reduced to 4.25% and the algorithm achieves a false negative rate of 1.08%. False negatives are caused by the threshold which excludes true fault events with Euclidean distances smaller than 44. Since faults with smaller Euclidean distances can be considered less urgent, the false negative rate of 1.08% is acceptable. Considering each string-pair for each day in the dataset, a total of 6 940 726 operating events are investigated during the fault detection procedure. The improved fault detection algorithm flags 43 139 events as possible faults. Given the estimated accuracy of 94.67%, abnormal operating conditions make up 0.59% of the total operation for the period investigated. It is noted that the sample set only represents 2.66% of all flagged events. A larger subset could be analysed, in order to improve the reliability of the detection rate estimate. The performance of the fault detection algorithm on the subset of flagged events is, however, deemed successful.

4.8 Summary

This chapter covered the review and implementation of fault detection for PV systems. Common faults in a PV system include hot-spots, shading, open- or closed circuit faults, ground faults, arc faults and degradation. Faults left unresolved can increase the risk of fire and even introduce safety hazards. The need for effective fault detection and diagnosis is clear, especially in large-scale PV systems. Methods for fault detection mentioned in literature includes visual inspection, signal analysis, statistical approaches and comparative models. The need for investigation into fault detection at utility-scale was identified. A fault detection algorithm is proposed that incorporates and expands on some of the concepts from previous methods. The detection procedure compares modelled and measured values of string-pair current in order to identify unexpected behaviour in the power plant. The Euclidean distance is used as the similarity metric when comparing the measured and expected current. The fault detection procedure is implemented on historic operational data and a sample detection accuracy of 88.5% is achieved. An improvement to the algorithm introduces a threshold to discard events with a small Euclidean distance. The improved sample detection rate is 94.67%. Future improvements could include narrowing the fault detection window to allow faster detection rates and possible real-time implementation. A larger sub-sample could also be analysed to obtain a precise detection accuracy. Investigation into detected events should provide insight into common system faults and provide a basis for research into predicting future anomalies. Using the measured data for fault diagnosis as described in [18] is also suggested as a topic for further research.

Analysis of events flagged during the fault detection procedure has inspired investigation into differences between string-pair currents. Performance differences between sub-sections of the power plant is valuable information for O&M teams as well as management. Chapter 5 covers the comparative performance analysis of the PV system.

Chapter 5

Plant Performance

Plant performance analysis is an important part of an utility-scale solar power plant. Power loss due to underperforming systems may lead to decreased profits. An effective monitoring system and performance reporting software enables operation teams to supervise the production of the power plant. Commercially available solutions like SMA's Sunny Portal, provides an interface for the monitoring and visualisation of systems and system data [45]. Sunny Portal includes an inverter comparison feature, where the user is notified when the yield of an inverter is outside the average yield of all the inverters. The comparison of subsystems is beneficial since underperforming subsystems is easily identifiable. The efficient location of subsystems showing abnormal behaviour in a large solar power plant saves time and effort for maintenance personnel having to perform corrective measures. The comparison of string-pair data can provide an even more detailed plant performance overview. Underperforming regions can thus be identified at a lower level than inverter level, which will improve response times to fault correction. Less downtime maximises power production and profits.

This chapter considers the use of visualisation to effectively communicate the plant performance and differences between string-pairs. A tool visualising the comparison between string-pairs is developed. The visualisation tool is used to verify some of the flagged events mentioned in Chapter 4. The benefits of using the tool for real-time plant performance evaluation is also considered. The average performance of string-pairs is compared by visualising the average power from each array over a period of three years. Lastly, the change in performance is investigated by calculating the degradation rate of each string-pair and visualising the results.

5.1 Visualisation

A visualisation tool is created to provide a comparative overview of string-pairs in the power plant. The plant layout is used in creating the visual representation of the entire power plant at string-pair level. The visualisation tool is built with Tkinter, a graphical user interface (GUI) package for Python [46]. Figure 5.1 shows a snapshot of the completed visualisation tool.

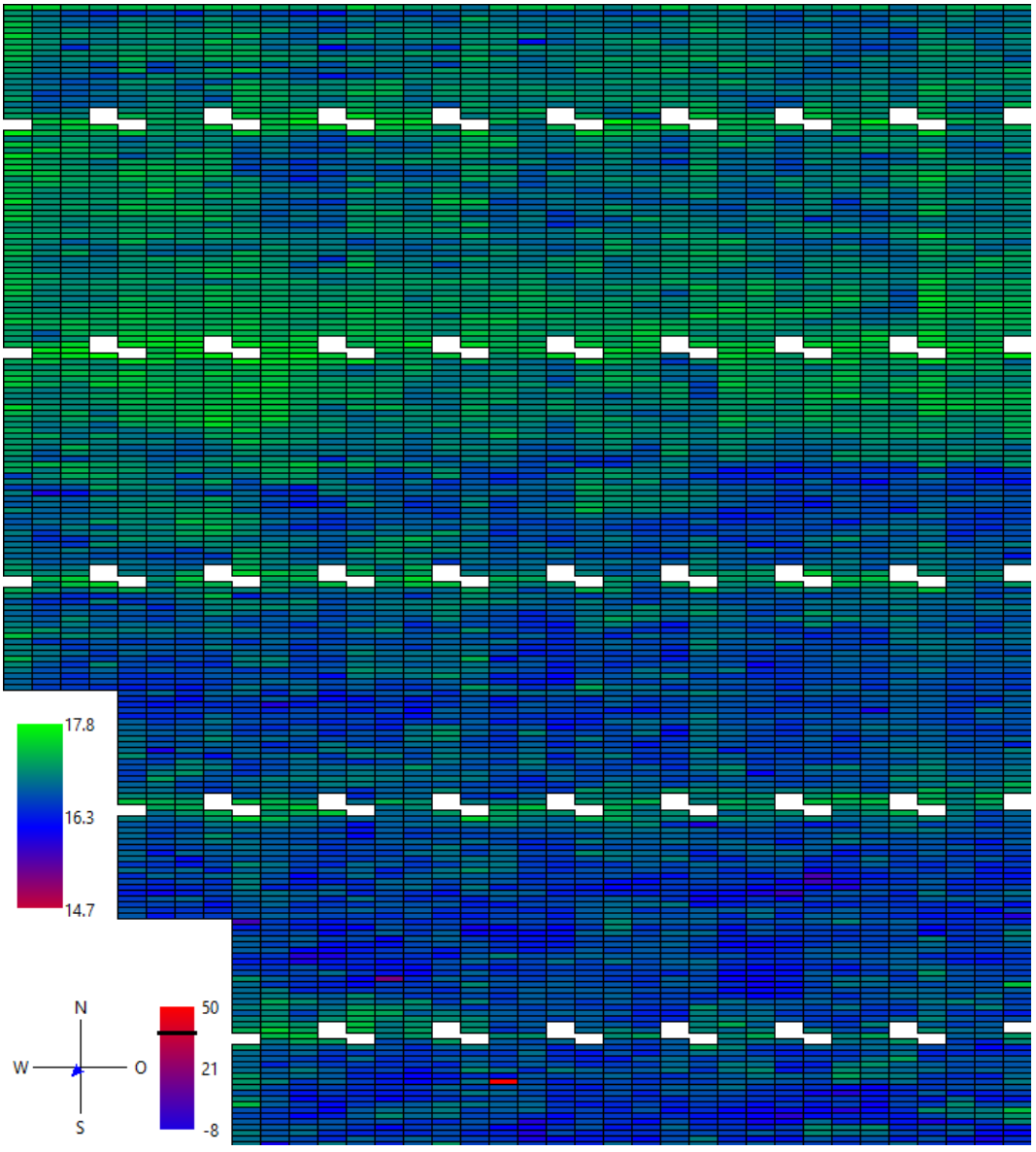


Figure 5.1: Snapshot of the string-pair performance visualisation tool.

Each coloured rectangle in the visualisation represents a string-pair. The string-pairs are arranged based on the relative location of the device in the actual plant layout. The visualisation is created by comparing the currents of all string-pairs. This process involves populating a list with the measured current for each of the 6511 string-pairs at a specific timestamp. Using the list of all measured currents, the maximum and minimum values are determined. Each string-pair is assigned a colour based on the gradient scale in Figure 5.2. The rectangle representing the string-pair with the highest current will be assigned green, while the string-pair with the lowest current is assigned red. A string-pair with a measured current exactly in between the minimum and maximum is assigned blue. The rest of the string-pairs are assigned shades between these three colours corresponding to the measured current of each. The variation in colours between regions in Figure 5.1 shows promise in using the tool for identifying underperforming sections in the power plant.

Figure 5.2: Colour gradient scale used in the performance visualisation.

The visualisation is updated each time-step which results in a playback of plant performance. Using the one-minute logged data, the visualisation can be updated for each minute in a day. As an illustrative example, an animation showing cloud movement over the power plant is presented in Figure 5.3. The animation is sped up to play four minute-snapshots per second. This example shows how the visualisation could be implemented as a real-time performance evaluation tool. Changes in string-pair behaviour are easily identified and would be beneficial for detecting abnormalities in the power plant. In case the reader is displaying this document in an application or format that does not allow animations, an online version should be available at: `https://github.com/waynedyamond/masters/blob/master/cloud_movement.gif`. The individual frames from the animation can also be viewed in Appendix C.

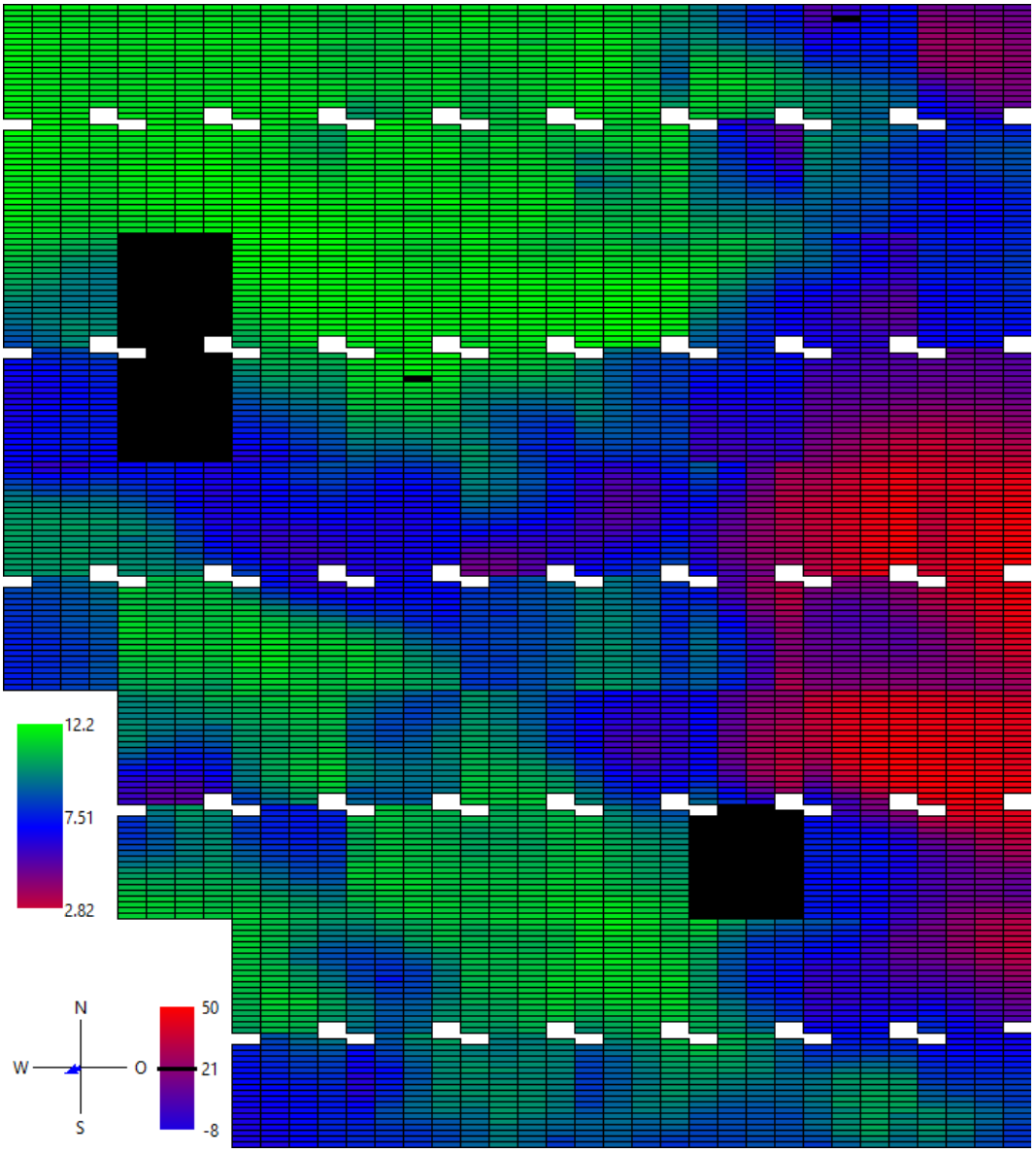


Figure 5.3: Animation of cloud movement over the power plant.

The current, wind and temperature displays, shown in Figure 5.4, provide the string-pair current scale and weather data recorded for the specific time. The ability to identify the string-pairs showing lower performance is an important requirement of the visualisation tool. During playback, if the user clicks on a rectangle, the name of the corresponding string-pair is printed. Right clicking on a rectangle will display the measured and modelled string-pair current for the entire day. Figure 5.5 shows a snapshot of this functionality. The developed visualisation tool is used to verify some of the faults detected in Chapter 4.

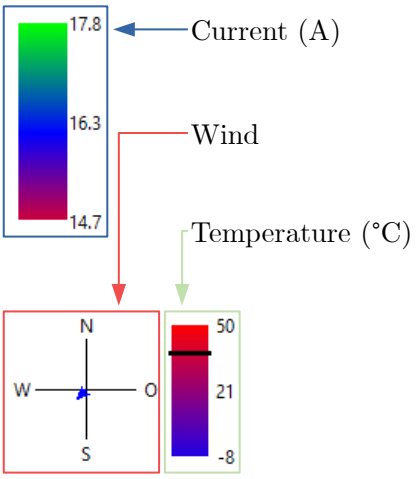


Figure 5.4: Current, wind and temperature displays.

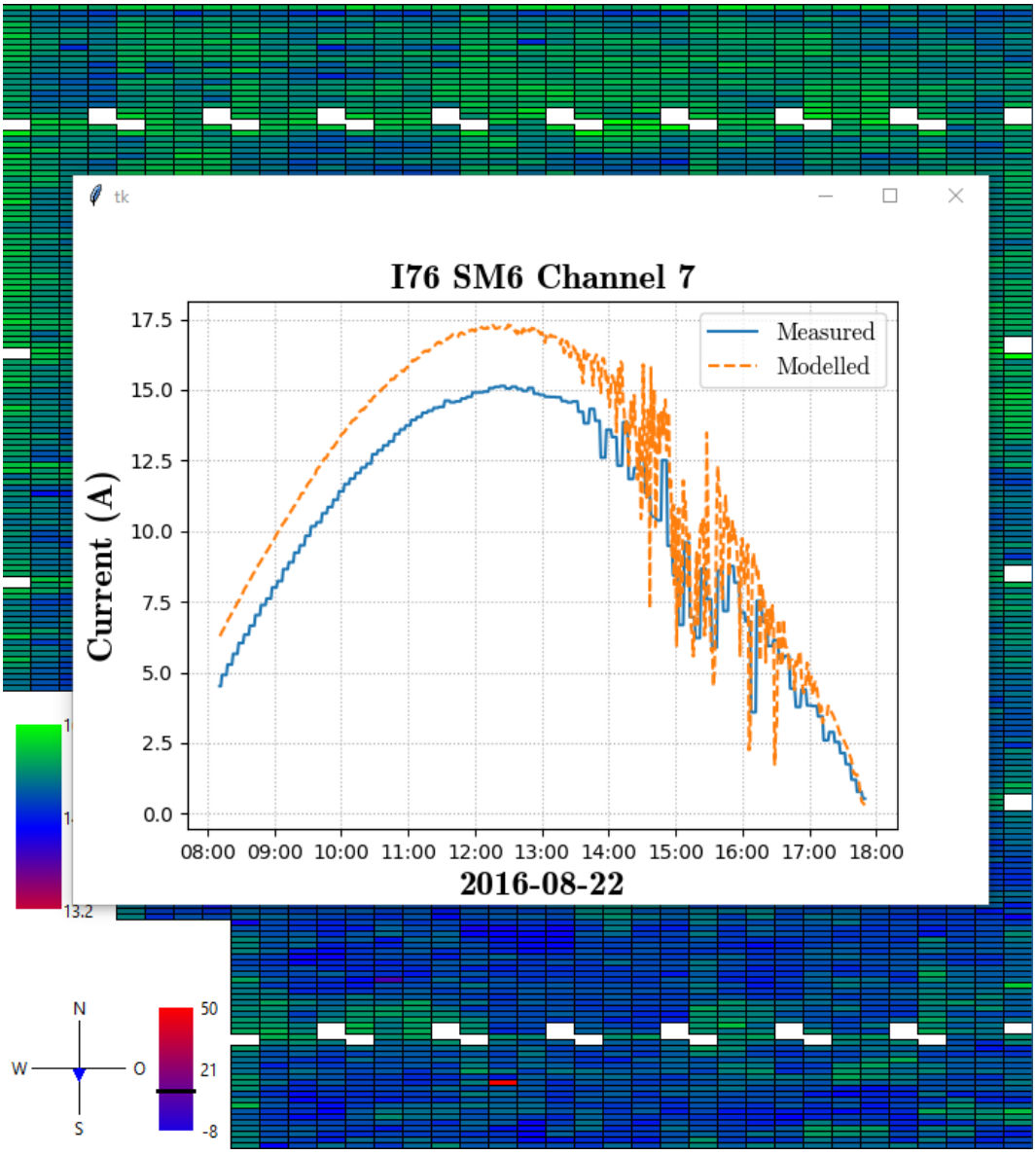
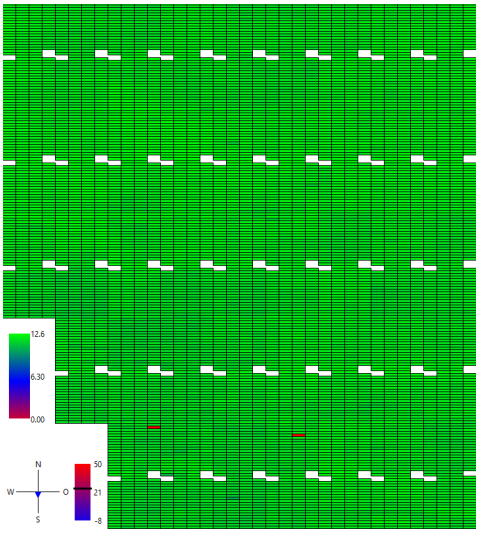


Figure 5.5: Right-click displays a pop-up of the string-pair current for the day.

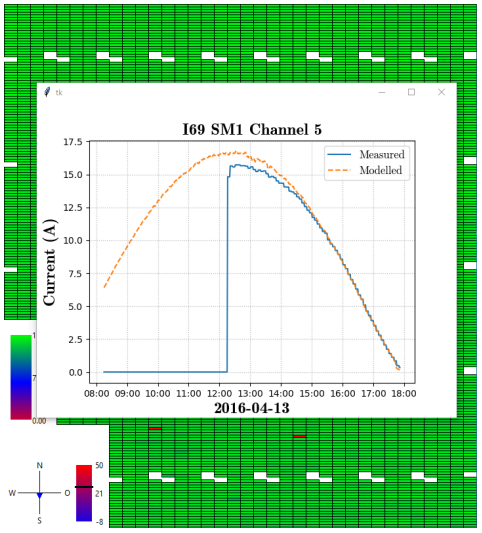
5.1.1 Identifying Fault Locations

Faults detected at string-pair level can be verified by using the performance visualisation tool. Since a string-pair in a fault state will typically have a current that differs significantly from the rest of the plant, the faulty string-pair will be visually identifiable. Figure 5.6 shows the visualisation and corresponding string-pair for a few test cases. The faulty string-pairs are clearly distinguishable from the rest of the string-pairs in the visualisation. This observation confirms the success of the fault detection algorithm and also illustrates the possibility to use the visualisation tool during operations and maintenance.

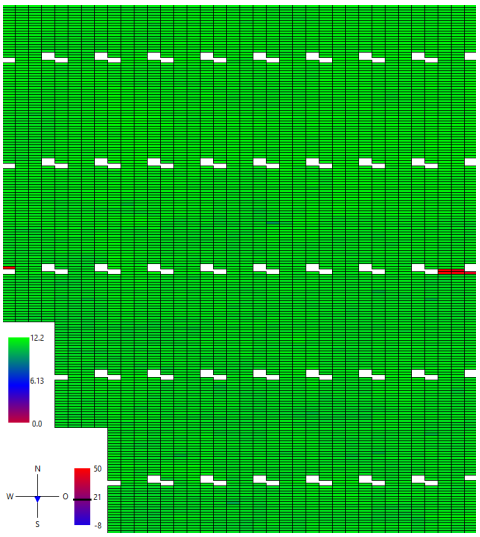
Figure 5.6a shows that individual faulty string-pairs can easily be identified. Multiple string-pairs and even faulty inverters can also be located using the visualisation tool as shown in figures 5.6c and 5.6e. The visualisation tool could be implemented to provide the operations team with a real-time visual representation of the power plant. Efficient fault location will reduce downtime and increase power output. The ability to verify and locate faults using a visual representation of the power plant is shown to be successful. In order to investigate more subtle performance variations, the distribution of colour in the overview can be analysed.



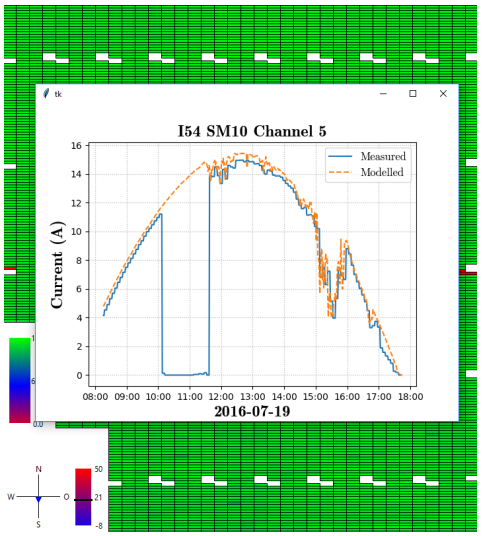
(a) Visualisation overview for a single time-step on 2016-04-13



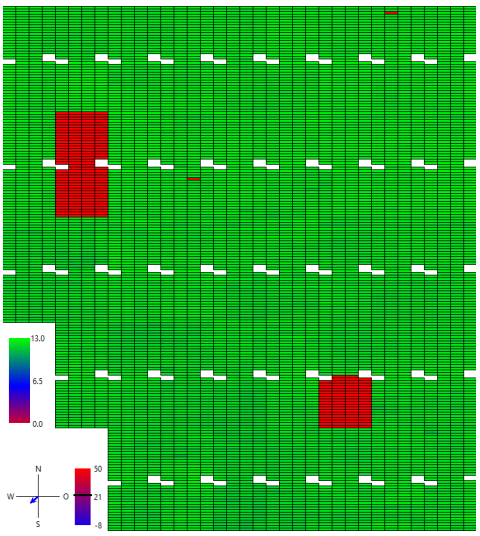
(b) One of the two corresponding faults for 2016-04-13



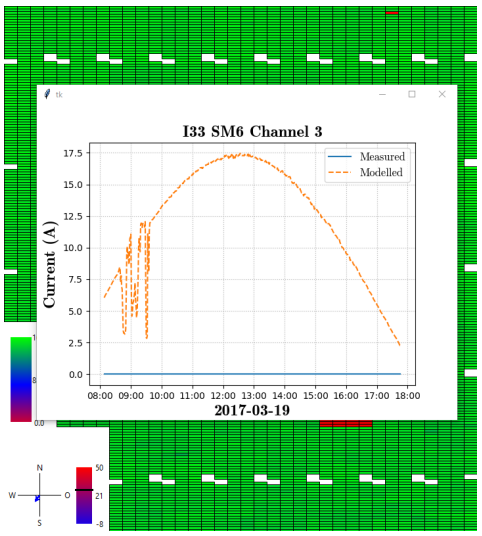
(c) Visualisation overview for a single time-step on 2016-07-19



(d) One of the corresponding faults for 2016-07-19



(e) Visualisation overview for a single time-step on 2017-03-19



(f) One of the corresponding faults for 2017-03-19

Figure 5.6: Visualisation and corresponding fault for a few test cases.

5.1.2 Visualising Performance Differences Between Regions in the Plant

The performance visualisation is used to identify regions of lower performance. Regions consistently producing less power will appear a shade of red, while better performing regions will appear green. In order to minimise fluctuations due to weather, the power is averaged over a period of time. The string-pair power can be used to indicate the performance of individual arrays; however, for the present work, only measured current is available at string-pair level. In order to calculate string-pair power, the measured DC input voltage at inverter level is used. Since string-pairs are connected in parallel, the voltage for string-pairs connected to the same inverter is equal. The string-pair power is determined by multiplying the string-pair current and inverter DC voltage. It is relevant to note that the inverter DC input voltage will be affected by string-pair faults, but since the average performance is considered, faults are assumed to have negligible effect on the final visualisation.

The period chosen for the visualisation corresponds to the span of times in the first dataset. Similar to the visualisation playback in Section 5.1.1 the power for each string-pair is extracted and compared to the rest. The scale is slightly altered for this analysis. The minimum and maximum value colour scale used in the previous visualisations is beneficial for identifying faults, but for subtle performance differences, outliers will affect the entire visualisation. In the modified colour gradient scale, string-pairs with power below three standard deviations from the mean are coloured red. String-pairs with an average power three standard deviations above the mean is assigned green. String-pairs with values between these bounds are assigned a colour based on the scale in Figure 5.7. The measured string-pair power between operating hours for each day is extracted and averaged for the time period. Each string-pair is assigned a colour corresponding to the gradient scale. Figure 5.8 shows the visual representation of average string-pair power for the entire plant.

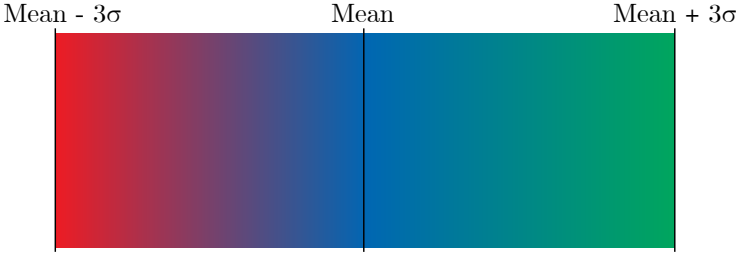


Figure 5.7: Modified colour gradient scale.

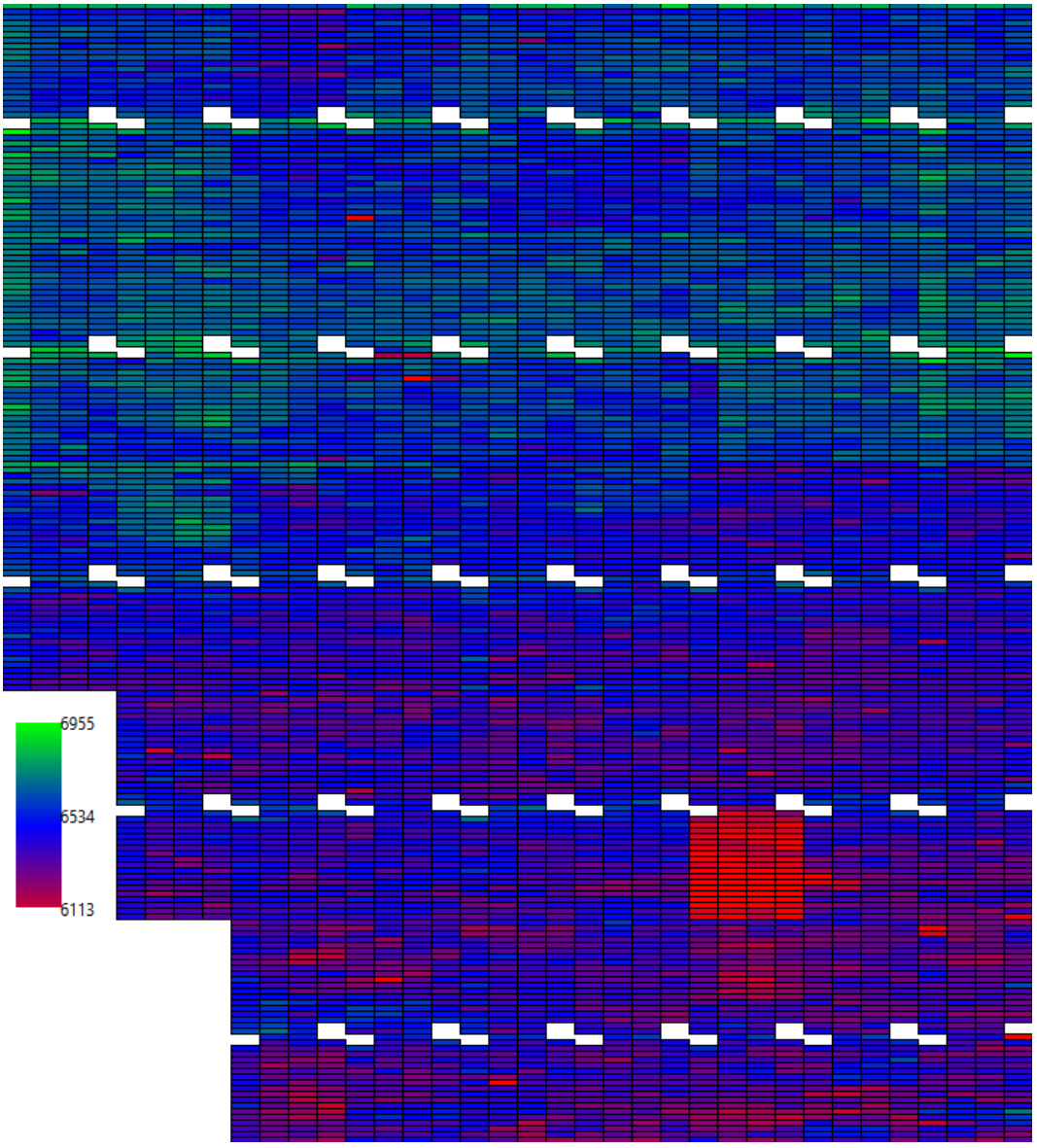


Figure 5.8: Visualisation of average string-pair power during operating hours.

It is clear from Figure 5.8 that some regions perform better on average than others. The highest density of green (better performing) string-pairs is found in the top half of the plant, while the bottom half has a higher density of red (lower performing) string-pairs. Inverter 60 contains almost entirely low performing string-pairs. This is caused by missing string-pair current data (see Figure 3.7) which influences the average. The importance of data availability is emphasised once again. Weather conditions could affect the power plant causing different regions to perform better or worse than others. The temperature at the different weather stations is compared to give an estimate of the effect temperature will have on different string-pairs. Figure 5.9 shows the

module temperatures measured at each of the four weather stations located in the plant for a clear sky summer day. A maximum difference of 10.1 °C is measured. The average module temperatures for the investigation time period are compared in Table 5.1 for further analysis.

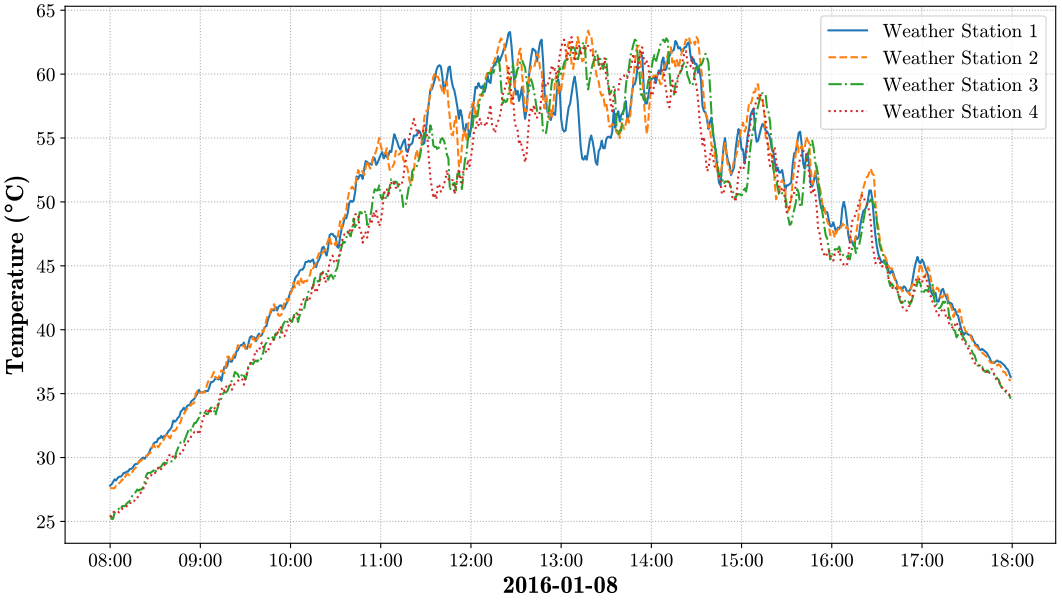


Figure 5.9: Module temperature at the four weather stations for a clear sky day in summer.

Table 5.1: Comparison of average module temperature between weather stations for 2016 and 2017.

Weather Station	**Average Temperature (°C)**
WS 1	37.37
WS 2	36.58
WS 3	35.18
WS 4	37.59

It is important to acknowledge that calibration and measurement inaccuracies could distort the averages. Assuming accurate analysis, the power plant does experience differences in temperature across the layout. Wind is another environmental factor that could influence module temperature and in turn average performance. Figure 5.10 shows the wind rose diagram generated from the measured wind speed and direction for the evaluation period.

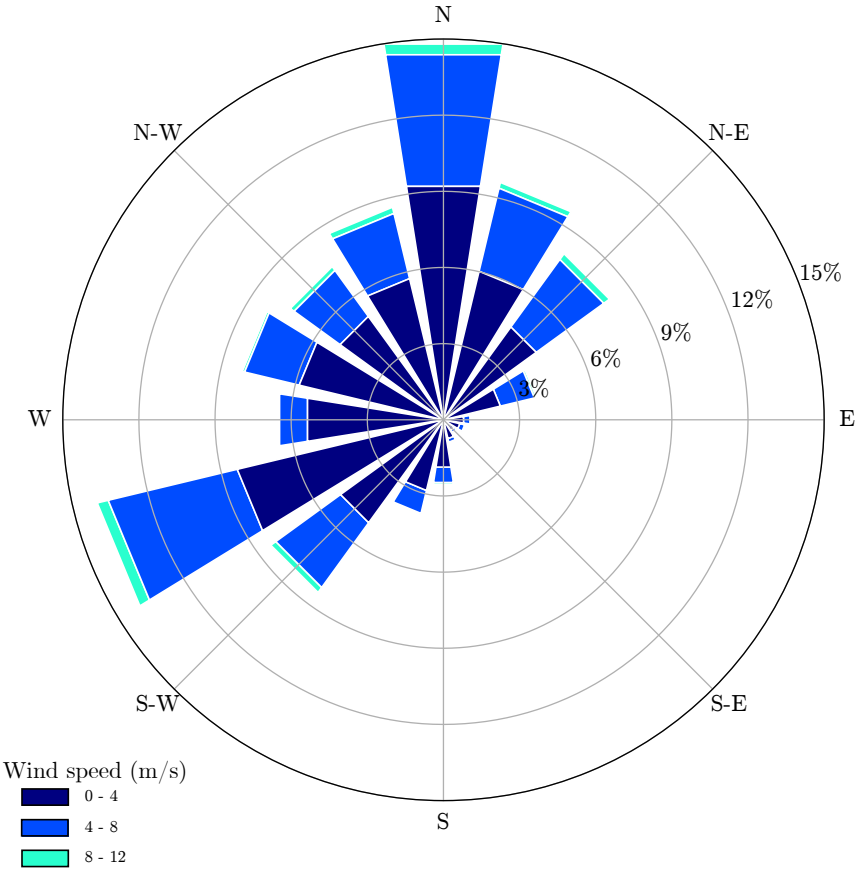


Figure 5.10: Wind rose diagram generated from measured wind speed and direction for the evaluation period.

The wind rose shows that the power plant experiences predominant wind from the north and west-southwest. The arrays in the direct wind flow would be cooled and experience higher performances, while also obstructing wind flow to regions further down the flow path. Differences between environmental conditions for individual string-pairs could explain the distribution of average performance in Figure 5.8. Future research could validate this hypothesis and lead to other possible causes. A study into the topic may produce valuable insights for planning future power plant layout and designs.

The average performance indicates the overall difference in performance among string-pairs; however, considering the change in performance over time is another relevant investigation.

5.2 Degradation

The decline of power produced by a PV system over time is known as degradation. The quantification of this phenomenon is termed as degradation rate, where higher degradation rates translate directly into less power produced [47]. The degradation of the PV devices in the plant at string-pair level can be visualised in a similar fashion to the work proposed above. Knowing which devices degrade at a faster rate, might also provide O&M with valuable insight.

5.2.1 Causes of Degradation

Degradation can occur at different subsystems in the entire PV system. The DC-side degradation might be affected by module- and conductive degradation as well as power point tracking control deterioration. Conductive losses can increase over time due to corrosion and failing connections. Maximum power point control can deteriorate due to hardware- or software failure. Module degradation is a well-researched topic. Causes of module degradation include soiling, optical degradation, cell degradation, mismatched cells and temperature-induced degradation [48]. Soiling degradation refers to the accumulation of dust on the surface of a module and can be alleviated by following washing schedules. Optical degradation is caused by the discolouration of the module surface due to extended ultraviolet (UV) exposure [48]. Discoloured modules can be visually identified and replaced as necessary. Temperature-induced degradation is caused by the elevated temperatures experienced by a PV module operating outdoors. Modules operating at temperatures higher than 25 °C leads to decreases in the cells' saturation current, which results in an overall reduction in the cell efficiency [48]. Cell degradation is mainly caused by increased series resistance or decreased shunt resistance. The series resistance can be increased by resistance arising in ageing solder bonds. A decrease in shunt resistance, often caused by crystal damage, leads to increased shunt currents and lower overall module performance. Mismatched cells are caused by surface soiling, shading, cell damage and manufacturing defects [48]. Mismatched cells can lead to hot-spot faults and degrade the performance of the device. The effects of cell degradation and mismatch are typically amplified as the module ages.

Independent power producers require accurate predictions of reduced power production over time for predicting return on investment [47]. After extensive investigation of literature relating to degradation rates by [47], the authors conclude that the median degradation rate for modules is 0.5%/year and the average value is 0.8%/year. This value corresponds to most manufactures warranty of about 1%/year. The module installed in the power plant investigated in this work, has a warranty relating to 10% degradation in 10 years. The total string-pair degradation is expected to be in this range, or slightly higher

than the typical module degradation rate, since conductive losses and MMPT-error also influences the degradation value. The visualisation tool is used to investigate the performance degradation of individual string-pairs. Knowledge of degradation at string-pair level could be used for better prediction of power production over time.

5.2.2 Calculating Degradation

Authors of [49] published a review of different methodologies for calculating photovoltaic degradation. Four major statistical analysis methods for calculating degradation rates are identified. These methods include: Linear Regression (LR), Classical Seasonal Decomposition (CSD), Auto Regressive Integrated Moving Average (ARIMA) and LOcally wEighted Scatterplot Smoothing (LOESS) [49]. Linear Regression is found to be the most popular statistical analysis method, although the method results in a larger variation and uncertainty than the ARIMA and LOESS counterparts. The statistical analysis methods can be applied to I-V curve characteristics corrected to STC or normalised ratings such as Performance Ratio (PR) to obtain the degradation rate. In the case of I-V characterisation, I-V curves from a few sample modules are recorded periodically and the degradation rate is calculated as the percentage error between consecutive recordings [49]. Using the performance ratio is also popular, since direct comparisons between different PV technologies, PV system capacities and geographical locations can be made. Since I-V curve characteristics are unavailable, the performance ratio is used as the performance metric in the presented work. Linear regression is used during the calculation of the degradation rate as it is widely used in literature.

5.2.2.1 Performance Ratio

The performance ratio of a PV system refers to the ratio of actual measured power yield to the theoretical reference power yield according to the nameplate rating of the system. The performance ratio is defined in the IEC 61724 standard [50] and calculated as shown in Equation 5.1 [47].

$$PR = \frac{P}{\left[P_{STC}\left(\frac{G_{POA\_i}}{G_{STC}}\right)\right]} \tag{5.1}$$

where:

PR = performance ratio (unitless)
P = measured power generation (kW)
P_{STC} = total DC power rating of system at standard test conditions (kW)
G_{POA} = measured plane-of-array (POA) irradiance (W/m^2)
G_{STC} = irradiance at standard test conditions (1000 W/m^2)

The performance ratio is susceptible to changes in weather since module temperature has an effect on performance. If the time period over which the PR is calculated is long enough (typically a year), the weather effect is eliminated. Since the performance ratio is calculated daily for the analysis in this work, the temperature corrected performance ratio is preferred. Correcting the PR for temperature is a common countermeasure to seasonal variations in performance ratio. The PR can be corrected to the STC temperature of 25°C which should reduce seasonal changes. Equation 5.2 shows the weather-corrected PR.

$$PR_{corr} = \frac{P}{\left[P_{STC}\left(\frac{G_{POA\_i}}{G_{STC}}\right)\left(1 - \frac{\delta}{100}(T_{STC} - T_{module\_i})\right)\right]} \tag{5.2}$$

where:

PR_{corr} = corrected performance ratio (unitless)
T_{module} = measured module temperature (°C)
T_{STC} = module temperature at testing conditions (25 °C)
δ = temperature coefficient for power (%/°C) of installed modules.

The performance ratio in Equation 5.1 and 5.2 is based on the ratio of measured and rated power. The string-pair power is used to visualise the degradation differences between regions in the power plant. The maximum operating string-pair power calculated as $P_{STC} = (2 \times 24)(240)$ W, since the string-pair consists of two strings (24 modules) in parallel. The STC rated power of a single module is 240 W and the temperature coefficient of power is $-0.47\%/°C$, as found in the module datasheet (Appendix B). The average irradiance and temperature from all four weather stations are used in order to ensure consistency when calculating the PR. The Python code segment for calculating the weather-corrected performance ratio is shown in Segment 5.1.

Segment 5.1: Python implementation for calculating performance ratio of a string-pair.

```
In [1]: def weather_corrected_pr(power, irradiance, temperature):
            pr = power/((2*24*240)*(irradiance/1000)*(1 - (0.0047*(25 - temperature))))
            return pr
```

The PR of each day in 2016 is calculated for a sample string-pair and the resulting scatter-plot is shown in Figure 5.11. The points coloured red are regarded as outliers and subsequently removed since these values will skew the degradation rate. Similar to the procedure in the fault detection algorithm, values are removed if the fall outside of the permissible range shown in Equation 4.4. The k value is chosen as $k = 2$ which discards values two standard

deviations from the mean. It is noteworthy, however, that performance ratios far from the trend might also be used to identify days in which the string-pair operation was faulty.

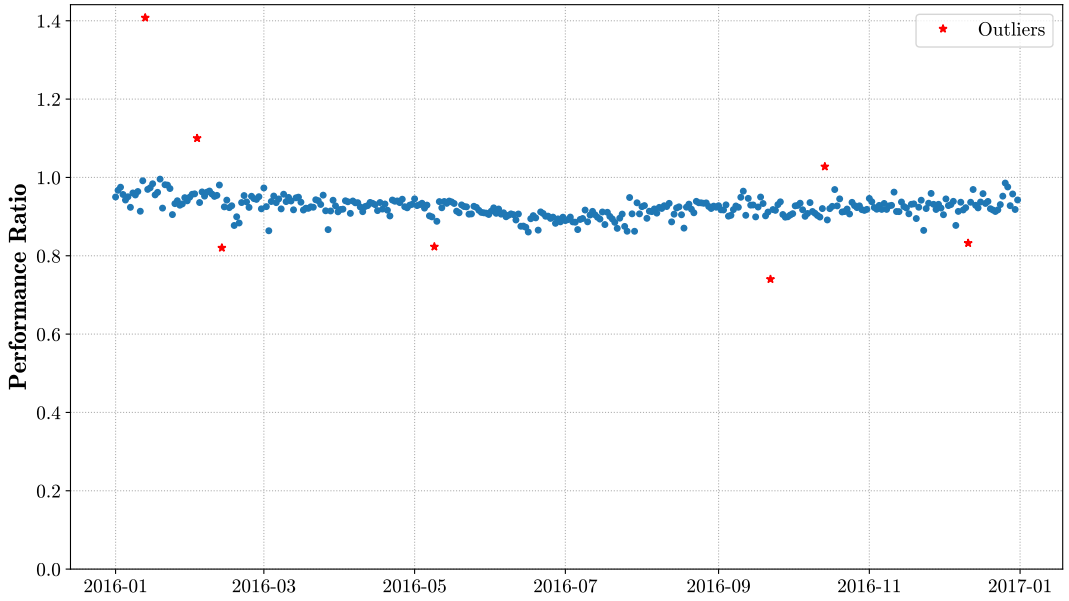
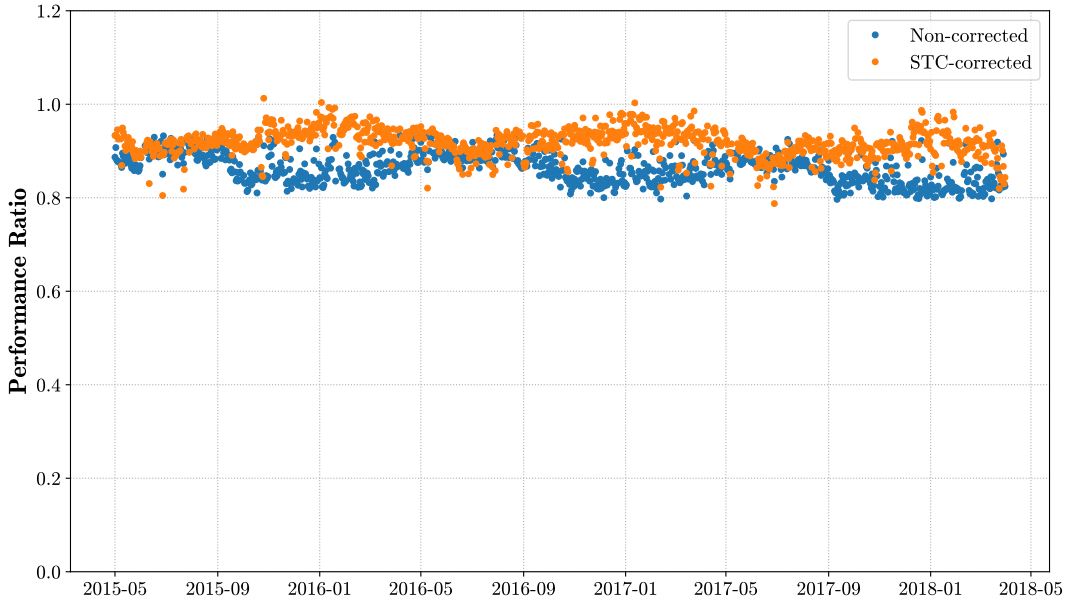
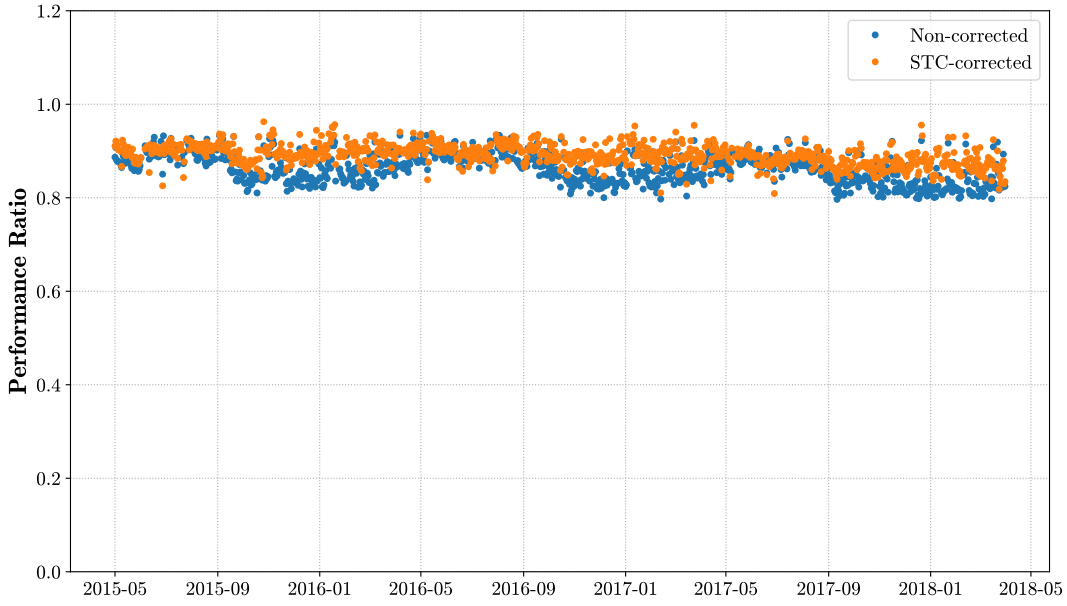


Figure 5.11: Scatter plot of the performance ratio calculated each day for a year.

Interestingly, using the temperature corrected performance ratio does not seem to suppress seasonal variation. Figure 5.12a shows the standard IEC 61724 PR versus the PR corrected to STC conditions. It is evident that the temperature correction still produces seasonal artefacts. The temperature coefficient of power on the datasheet might be inaccurate. In order to investigate this possibility, the temperature coefficient for power is slightly altered during the PR calculation. A value of $-0.25\%/°C$ seems to suppress the most noticeable seasonal variations. Since there is such a large discrepancy, an unknown factor could be affecting the temperature corrected performance ratio. Figure 5.12b shows the comparison of the standard performance ratio and the PR calculated with the reduced temperature coefficient. Another possibility is that the difference in solar spectrum between winter and summer months, which is not accounted for in PR calculation, has a larger influence on the result. Since the linear trend is considered for degradation, the seasonal variations in PR is tolerable. Further research should be conducted to investigate the seasonal artefacts in the temperature corrected performance ratio. The trend in performance can be determined using linear regression.



(a) Temperature coefficient for power from datasheet (−0.47%/°C).



(b) Altered temperature coefficient for power (−0.25%/°C).

Figure 5.12: Comparison of standard performance ratio and performance ratio corrected to STC conditions.

5.2.2.2 Linear Regression

The main concept of linear regression is to find a straight line, $y = mx + c$, which provides the best fit through a set of data points. The gradient of the straight line that fits the PR points calculated for each day is taken as the DC degradation rate. The Scikit-learn module for Python provides an

implementation of linear regression which is used to calculate the equation of the degradation line [51] (see Appendix A). Figure 5.13 shows the degradation line fit to the performance ratio values. The gradient of the degradation line shown in Figure 5.13 is -0.0189, which can be expressed as a DC degradation rate of 0.65%/year. The degradation rates for all string-pairs are calculated over the entire time period and used to visualise differences in performance decline throughout the power plant.

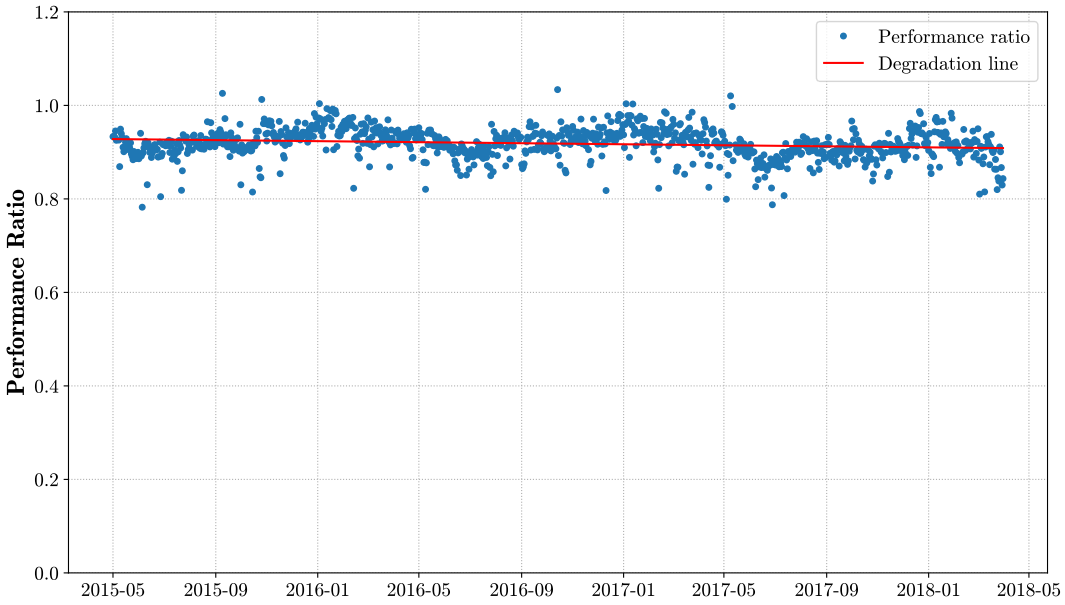


Figure 5.13: The degradation line fit against performance ratio points using linear regression.

5.2.3 Visualising Degradation

In order to visualise the DC degradation at string-pair level, the degradation rates for all string-pairs are calculated. The investigation time period is between '2015-05-01' and '2018-03-31'. Algorithm 5.1 shows the pseudocode in which the list of all degradation rates is obtained. The degradation rates are then visualised using the same method for producing Figure 5.8.

Algorithm 5.1 Pseudocode for generating list of degradation rates of all string-pairs.

```
get plant layout: inverters, monitors and strings
FOR each inverter in all inverters DO
    FOR each monitor in monitors connected to inverter DO
        FOR each string in strings connected to monitor DO
            get all days between start date and end date
            FOR each day in all days DO
                get performance ratio for string
                add performance ratio to list of ratios for string
            END FOR
            remove outliers from list of ratios
            apply linear regression to find degradation rate
            add degradation rate and corresponding string to
                list of degradation rates
        END FOR
    END FOR
END FOR
```

Figure 5.14 shows the distribution of DC degradation for the power plant. Note that the sign (+/-) is included in the gradient scale, since some string-pairs show an increase in performance over time. A positive value corresponds to a performance increase, while a negative value means that the string-pair performance has decreased over time. The unit for the degradation is in %/year where a value of -0.6 resembles a string-pair performance decrease rate of 0.6%/year. Figure 5.14 indicates that different regions in the power plant have different corresponding degradation rates. This knowledge can be valuable to operations personnel since modules in regions of faster DC degradation might need to be replaced more often to ensure optimal performance.

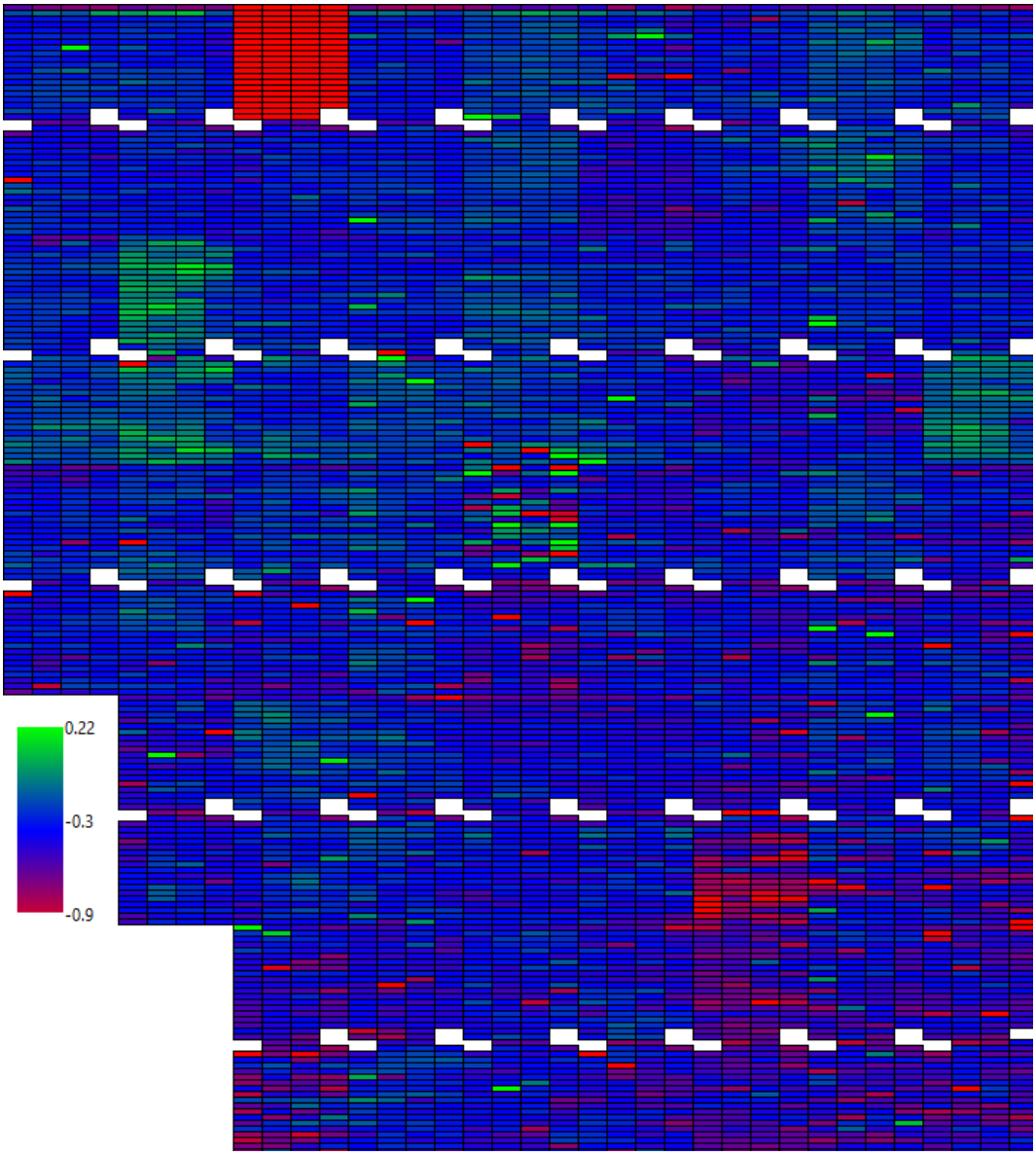


Figure 5.14: Visualisation of degradation rates for the entire power plant for '2015-05-01' to '2018-03-31'.

Inverter blocks 5, 33, 34 and 60 stand out from the rest in the degradation visualisation. String-pairs connected to inverter 5 and 60 show high degradation rates, while inverter 33 and 34 have string-pairs with low degradation rates. Due to missing data, the performance ratios for inverter 60 are unreliable (Figure 5.16d). Inverter five shows a high degradation rate which is caused by inconsistency in the operating hours for the inverter (Figure 5.16f). This problem is mentioned in Chapter 4.4. Inverter 33 and 34 show string-pairs with increased performance since both inverters were decommissioned for three months due to damage caused by lighting (Figure 5.16b). The degra-

dation line is thus influenced by the missing data. The installation of new inverters could also cause a slight increase in PR for the connected string-pairs due to increased efficiency and MPPT accuracy. Figure 5.15 shows the same degradation visualisation with these inverter blocks removed. The comparison between the rest of the string-pairs is more pronounced when the outliers are removed.

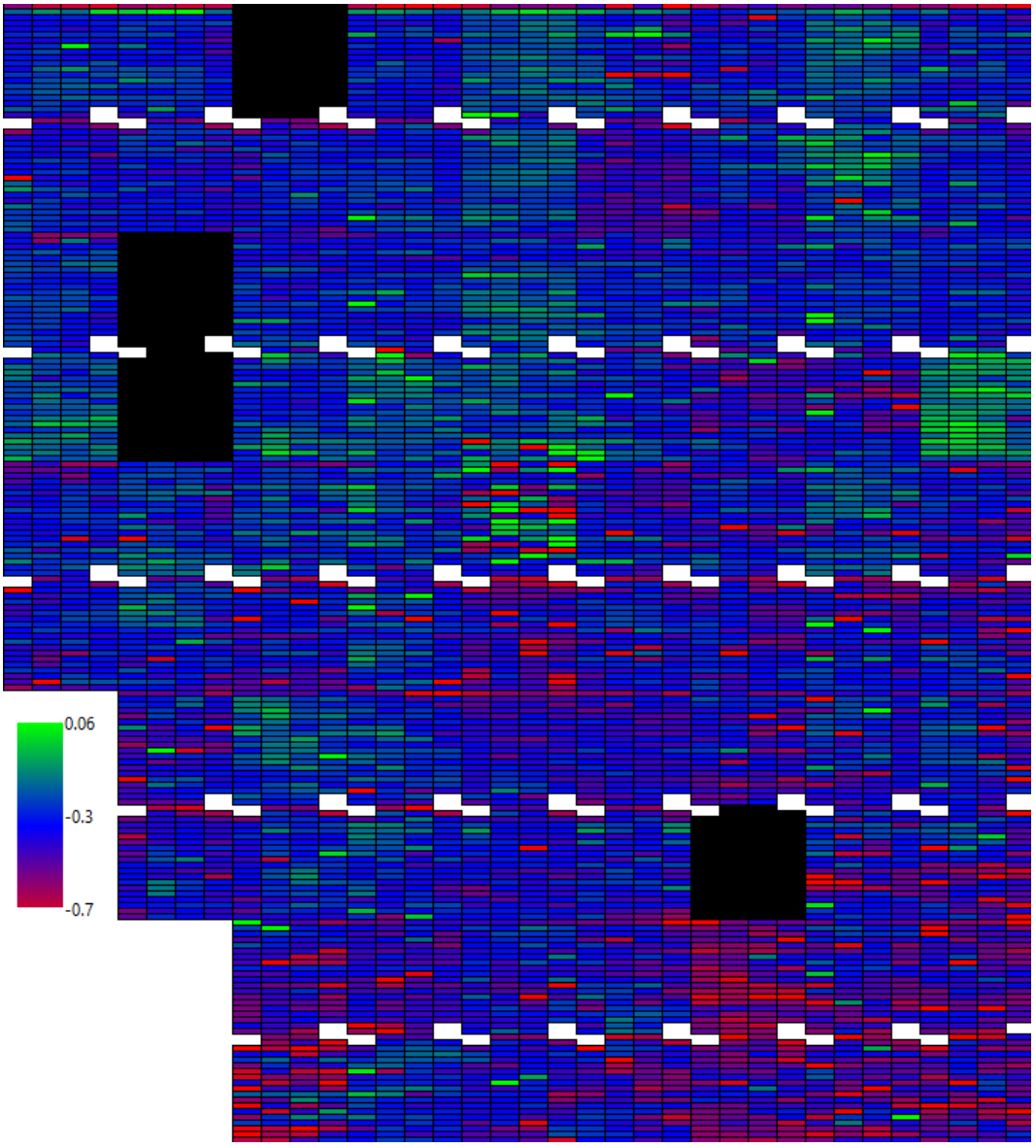
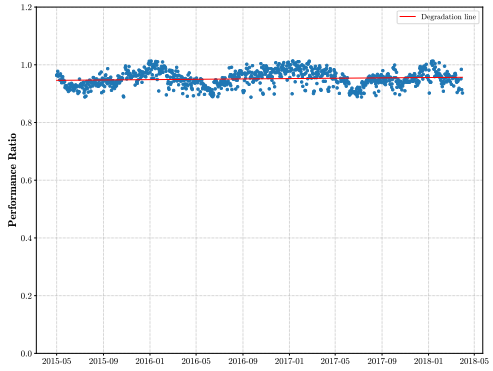
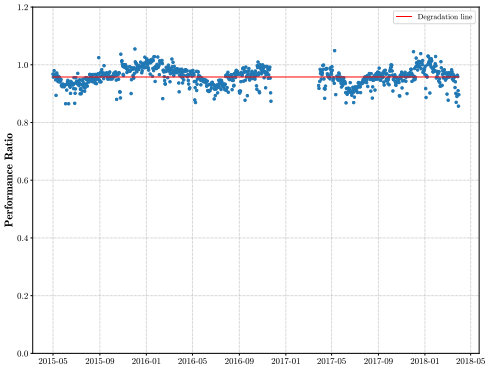


Figure 5.15: Visualisation of degradation rates with outlier inverter blocks removed.

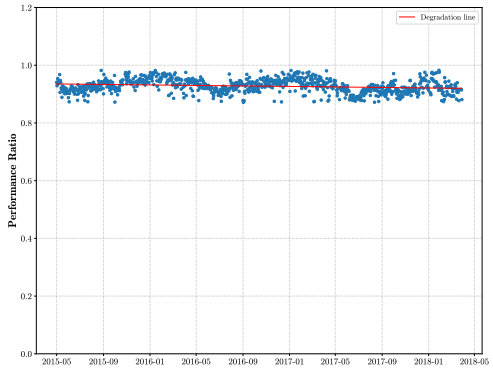
The average degradation rate for all string-pairs is 0.38%/year (-0.38 on the visualisation). There are, however, string-pairs that show significantly faster degradation rates and other string-pairs displaying only minimal degradation, or even a performance increase over time. Figure 5.16 shows a number of sample string-pairs and the corresponding degradation values of each.



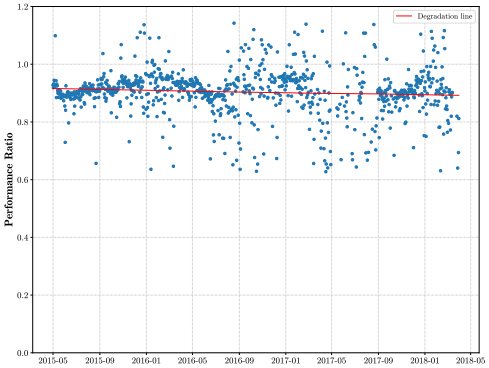
(a) Performance increase rate: 0.362%/year



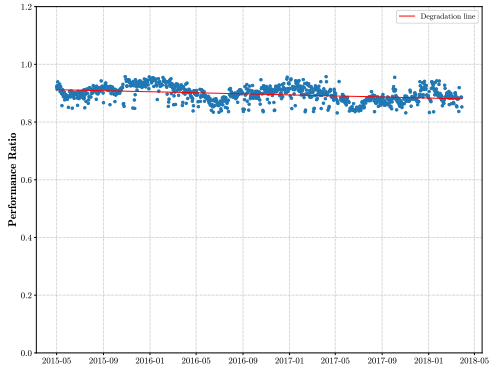
(b) Performance increase rate: 0.005%/year



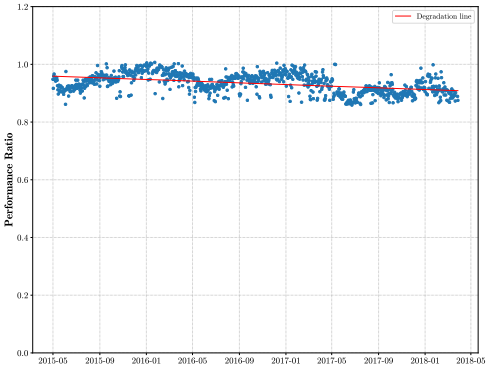
(c) Degradation rate: 0.520%/year



(d) Degradation rate: 0.809%/year



(e) Degradation rate: 1.081%/year



(f) Degradation rate: 1.722%/year

Figure 5.16: Some string-pair degradation curves, with corresponding rates.

Considering the median value for degradation found in [47], the average degradation rate is not improbable; however a value this low is surprising. The influence of missing data and seasonal variation on the degradation, should be taken into account. The average DC degradation could be compared to the degradation of the entire power plant, which might confirm or contradict the result. The main causes of degradation are mentioned in Section 5.2. Since the visualisation shows that different regions in the power plant experience different degradation rates, these causes might affect some regions more than others. Assuming the degradation analysis is correct and the data integrity can be trusted, investigating temperature distribution, performance differences and maximum power point tracking errors could lead to answers regarding the degradation distribution. Another consideration is the possibility that different module batches used during construction could have slightly different degradation rates. Modules are often batched by output tolerances and production dates, which could lead to slight differences in module characteristics and ultimately, different degradation rates. The string-pair degradation analysis is considered valuable. Degradation differences throughout the power plant can be used by O&M as well as power plant managers to assess the performance of specific regions and string-pairs in the power plant. String-pairs showing higher degradation rates could be tested more frequently for module defects. A proactive approach might lead to fault prevention rather than fault correction.

5.3 Summary

This chapter introduces a comparative visualisation of string-pair performance. The tool is shown to be useful in the detection and location of underperforming arrays. A few faults identified in the previous chapter are verified. The difference in average power between string-pairs is illustrated, where some regions are shown to perform above or below average. A visual representation of degradation throughout the power plant is also produced and the resulting average degradation rate is $0.38\%/year$. The visualisation tool is useful since an entire plant overview is reported, while also preserving lower-level string-pair details. The use of string-pair current data shows the advantage of recording measurements at lower-level devices in the plant. The effective analysis and visualisation of data generated by devices in a photovoltaic power plant is valuable and could spur future research into the topic.

Chapter 6

Conclusions and Recommendations

6.1 Introduction

The need for alternative energy sources to fossil fuels is made clear by the rising annual greenhouse gas emissions and limited availability of resources. Solar energy is posed as a good renewable alternative, since photovoltaic technology has seen improved efficiency and declining costs in the past decade. The variability of the resource is, however, one of the main drawbacks. Improving the reliability of solar power production is a necessary condition for large-scale implementation of PV power. This project investigated the use of measured data, generated by a utility-scale PV power plant, for fault detection and performance visualisation. The thesis is structured into four content chapters, each covering one of the objectives set out for the project. These objectives include:

- processing the raw data into a clean workable dataset,
- modelling string-pair power using the measured data,
- development of a fault detection procedure tested on measured data from a utility-scale power plant and
- creating a visualisation tool for comparing performance differences in the system.

These objectives have been met as reported in chapters 2 - 5.

6.2 Chapter 2

Chapter 2 covers the basic building blocks of a PV system and the process of modelling devices in the system. The single diode model proves to be an

accurate model for estimating the current and voltage of a PV device. The current, modelled using PVLIB, is compared to a baseline parametric model [16]. The single diode model is the preferred modelling method, since calculating the coefficients for the parametric model is dependent on site-specific historic measured data. Reference parameters for the single diode equation are obtained in the database of CEC and Sandia modules which PVLIB provides. However, a non-iterative method, demonstrated in Appendix A.1, can also be used to determine these reference parameters. An important consideration is that the model is highly dependent on measured irradiance and module temperature. Since incorrect measured weather data will lead to inaccurate modelling, regular calibration and testing of measurement equipment is advised.

6.3 Chapter 3

The data processing procedure is presented in Chapter 3. Raw sensor data is received as multiple CSV files with a total size of 560 GB. A database is created on a network-connected server in order to combine and centralise the dataset. The size of the dataset is drastically reduced to 145 GB by removing redundant columns and replacing far-out values with NULL. The decision is made not to impute missing values; however, for other applications, imputation techniques could be used to reduce the impact of missing data on the results. A second batch of raw data was received before an effective data pipeline was established. Had the proposed data pipeline been incorporated on the first dataset, the second dataset could have been integrated immediately, resulting in a larger testing dataset. As a result, only the first dataset is used in the rest of the project. The processed dataset is found to contain almost two years of missing data, which could have been lost after storage or due to communication error. The data for the first two years is disregarded during fault detection and performance visualisation. The amount of remaining available data is still considered acceptable to test the proposed fault detection algorithm.

6.4 Chapter 4

A fault detection procedure based on the comparison of measured and modelled string-pair current is developed in Chapter 4. String-pair current data is used for comparison, which allows fault detection at the lowest level. The modelling method described in Chapter 2 is used to determine the expected current. Euclidean distance is chosen as the comparison metric between measured and modelled values, since the calculation time for dynamic time warping distance is considered too high. The Euclidean distance also provides an acceptable separation between true positive and false positive faults. Missing

data, string-monitor recording hours and inaccuracies in the modelled current are considered during the detection procedure. These aspects have an impact on the accuracy of the fault detection algorithm. The proposed algorithm is applied to data generated by a substantially larger PV system than in previous research. 45 040 events are identified as having possible fault conditions. The large amount of events identified during the fault detection procedure necessitates the use of a sample set to determine accuracy. A subset of 1200 events is randomly selected, but future improvements should analyse a larger subset in order to obtain a more reliable estimate of the detection accuracy. 11.5% of detected events in the sample set are labelled as false positives. Since most false positives correspond to an event with a small Euclidean distance, a threshold of $d_{Euclidean} > 44$ is implemented to improve the detection accuracy. The improved detection algorithm has a sample accuracy of 94.67%.

6.5 Chapter 5

Chapter 5 includes research into plant performance visualisation. A tool is developed which visualises the comparison of string-pair currents for the entire power plant. It is shown that the visualisation tool can be used to identify underperforming string-pairs and inverters. Some of the faults detected in Chapter 4 are verified with the graphical interface. Next, the inverter DC input voltage is used to calculate string-pair power. It is noted that string-pair faults could influence the inverter input voltage, since string-pairs are connected in parallel. The average power for each string-pair over a period of three years is visualised. Regions with average performance above or below the mean are clearly visible. A similar visualisation is generated for representing the change in performance over time. The temperature corrected performance ratio and linear regression are used to calculate the degradation rate for each string-pair. An interesting observation is made with regard to using the datasheet value for the temperature coefficient during PR calculation. The temperature correction was expected to suppress seasonal variation; however, these artefacts were still present. A comparative visualisation is created to illustrate the differences in degradation between string-pairs. The average degradation rate is $0.38\%/year$, with some string-pairs showing clear deviation from the mean. Missing data for inverter 33, 34 and 60 caused inaccuracies in the average performance and degradation visualisations, which illustrates the dependency on complete and accurate measured data. The visualisation tool shows promise in effectively representing the overall plant performance while also maintaining string-pair features.

6.6 Project Conclusions

This project focused on the use of measured data, mainly string-pair currents, for fault detection and performance analysis. The importance of data pre-processing and quality assessment is clear. This process is time consuming, but knowledge about the dataset is necessary for further analysis. The impact of missing data is evident in the results of the fault detection procedure and performance visualisations. This project used a data driven approach to fault detection and performance visualisation. A main concern with this method is that the analysis is often only speculative. Justifying causes of common faults, average string-pair power distributions and degradation patterns using only measured data is difficult. Missing data and measurement inaccuracies will influence the outcome of these procedures. Validation of the results using physical tests, error logs and extended examination is required.

6.7 Recommendations and Further work

Fascinating results from fault detection and string-pair visualisation could stimulate further research. The comparison window used in the detection algorithm corresponds to the times between operating hours in a single day. In future implementations, the comparison period could be shortened to allow a faster detection rate. Constraining the comparison period could decrease detection accuracy and is therefore a consideration to investigate. The possibility of real-time fault detection is another research topic which could be investigated. This project implemented fault detection, but the need for accurate fault diagnosis clearly arises in literature. Most approaches in the reviewed literature used voltage and current characteristics to diagnose a pre-defined set of faults. The faults detected in this project could be used in future research to identify common system faults and possibly predict system abnormalities. A larger subset of flagged events could also be analysed in order to establish greater confidence in the accuracy of the detection procedure.

Implementation of the visualisation tool could provide operations and maintenance with valuable real-time insight and improve response time to system faults. An investigation into environmental differences between regions in the power plant could explain the variation in average power and degradation between string-pairs. Installing more weather stations would provide greater detail of temperature and irradiance differences within the plant. Further investigation into temperature corrected performance ratio is also recommended. Exploring the effect of solar spectrum differences between winter and summer months could explain why seasonal variation is still present after correcting for temperature. The average degradation rate is also considered surprisingly low and could be confirmed with further analysis of degradation in the en-

tire plant. Visualising performance differences between different devices in the system is shown to be an effective reporting tool. Further research could investigate other visualisation techniques and the comparison of devices other than string-pairs. Measurements at higher-level systems could be analysed in future research. Finding valuable trends in the data, common system faults and effective visualisation techniques could all prove to be important considerations. Knowledge gained by future research into these topics could provide increased stability and reliability of photovoltaic systems.

Appendices

Appendix A

Code

A.1 Estimating Reference Parameters

Batzelis et al. [21] implements a non-iterative method for estimating the five parameters of the single diode model. This approach can also be used to estimate reference parameters from information given in the module datasheet. The reference parameters are used in modelling the expected power as described in Section 2.2.2. The non-iterative method described in [21], introduces a coefficient δ_0 at STC which is calculated using the temperature coefficients $\alpha_{I_{sc}}$ and $\beta_{V_{oc}}$. The δ_0 coefficients is then used in deriving the auxiliary parameters δ and ω in equations A.1 to A.3.

$$\delta_0 = \frac{1 - \beta_{V_{oc}} T_0}{50.1 - \alpha_{I_{sc}} T_0} \tag{A.1}$$

$$\delta = \delta_0 \frac{V_{oc0}}{V_{oc}} \frac{T}{T_0} \tag{A.2}$$

$$\omega = W_0\{e^{\frac{1}{\delta}+1}\} \tag{A.3}$$

Note that $\delta = \delta_0$ for calculating the reference parameters. Equations A.4 to A.8 show the how the coefficients determined by in the previous equations are used to calculate the five parameters of the single diode model. Segment A.1 shows the reference parameter estimation for the 'BYD 240P6-30' module from datasheet specifications found in Appendix B.

$$a = \delta V_{oc} \tag{A.4}$$

$$R_s = \frac{a(\omega - 1) - V_{mp}}{I_{mp}} \tag{A.5}$$

$$R_{sh} = \frac{a(\omega - 1)}{I_{sc}(1 - \frac{1}{\omega}) - I_{mp}} \tag{A.6}$$

$$I_{ph} = \left(1 + \frac{R_s}{R_{sh}}\right) I_{sc} \tag{A.7}$$

$$I_s = I_{ph} e^{-\frac{1}{\delta}} \tag{A.8}$$

Segment A.1: Implementation of non-iterative parameter estimation for obtaining module reference parameters from datasheet specifications.

In [1]:
```
import numpy as np
from scipy.special import lambertw

# Datasheet information:
# ********************** #
Voc = 37.54
Vmp = 29.55
Isc = 8.9
Imp = 8.12
B_Voc = -0.34/100
a_Isc = 0.045/100
# ********************** #

# Values at STC
T_0 = 25 + 273.15
Voc_0 = Voc
T = T_0

# Calculate coefficients as described by Batzelis et. al
d_0 = (1 - B_Voc*T_0)/(50.1 - a_Isc*T_0)
d = d_0*(Voc_0/Voc)*(T/T_0)
w = np.real(lambertw((np.exp((1/d) + 1))))

# Calculate reference parameters
a_ref = d*Voc
R_s = (a_ref*(w-1)-Vmp)/Imp
R_sh_ref = (a_ref*(w-1))/(Isc*(1-(1/w))-Imp)
I_L_ref = (1+(R_s/R_sh_ref))*Isc
I_o_ref = I_L_ref*np.exp(-1/d)

reference_parameters = {'a_ref': a_ref, 'R_s': R_s, 'R_sh_ref': R_sh_ref,
                        'I_L_ref': I_L_ref, 'I_o_ref': I_o_ref}
reference_parameters
```

Out[1]:
```
{'a_ref': 1.512927326888829,
 'R_s': 0.40230270162060483,
 'R_sh_ref': 84.62893108422166,
 'I_L_ref': 8.942308156307211,
 'I_o_ref': 1.4975348602169721e-10}
```

A.2 Parametric Model Curve Fitting

A Python implementation of the parametric model from [16] is used during baseline comparison for PV modelling. Equation A.9 shows the parametric model for output power (I) given irradiance (G) and module temperature (T_m). The the coefficients a_1 through a_4 are unknown, but can be determined using historic irradiance, temperature and current values. The curve fitting technique known as non-linear least squares is used to solve the coefficients. The irradiance, temperature and power data for a string-pair in 2014 is used with the Python function '*scipy.optimize.curve_fit*' to solve the four unknown coefficients. The sample Python code in Segment A.2 shows the implementation and resulting coefficients.

$$I = G(a_1 + a_2 G + a_3 \log(G))(1 + a_4(T_m - 25)) \tag{A.9}$$

Segment A.2: Sample implementation of curve fitting to solve the coefficients in the parametric model.

In [2]:
```
import numpy as np
from scipy.optimize import curve_fit

def func(X, a1, a2, a3, a4):
    return X[0] * (a1 + a2*X[0] + a3*np.log(X[0])) * (1 + a4*(X[1] - 25))

# in-plane irradiance, module temperature and string-pair current for 2014
xdata = np.array((irradiance, temperature))
ydata = np.array(current)

popt, pcov = curve_fit(func, xdata, ydata)
print("Coefficients (a1, a2, a3, a4):")
print(popt)
```

```
Coefficients (a1, a2, a3, a4):
[-2.66349127e-02 -1.56998387e-05  8.13731326e-03  5.39239968e-03]
```

A.3 Dynamic Time Warping Distance

Segment A.3: Python implementation for calculating DTW distance.

In [1]:
```
import numpy as np

# Dynamic Time Warping:
def DTW_Distance(x, y, w=None):
    # fill empty matrix with np.inf
    dtw_matrix = np.ones((len(x)+1, len(y)+1)) * np.inf

    # initial conditions
    dtw_matrix[0][0] = 0
    for i in range(1, len(x)+1):
        dtw_matrix[i][0] = np.inf
    for j in range(1, len(y)+1):
        dtw_matrix[0][j] = np.inf

    # no warping window specified
    if w is None:
        # calculate warping matrix
        for i in range(1, len(x)+1):
            for j in range(1, len(y)+1):
                cost = (x[i-1] - y[j-1])**2
                dtw_matrix[i][j] = cost + min(dtw_matrix[i][j-1],
                                              dtw_matrix[i-1][j],
                                              dtw_matrix[i-1][j-1])
    else:
        # locality constraint w
        w = max(w, abs(len(x)-len(y)))
        # calculate warping matrix
        for i in range(1, len(x)+1):
            for j in range(max(1, i-w), min(len(y)+1, i+w)):
                cost = (x[i-1] - y[j-1])**2
                dtw_matrix[i][j] = cost + min(dtw_matrix[i][j-1],
                                              dtw_matrix[i-1][j],
                                              dtw_matrix[i-1][j-1])

    # calculate DTW distnace
    dtw_distance = np.sqrt(dtw_matrix[-1][-1])
    return dtw_distance
```

A.4 Calculating the Degradation Line Using Linear Regression

Segment A.4: Python implementation for calculating the degradation line using linear regression.

In [7]:
```
performance.head()
```

Out[7]:

	day	ratio
0	2015-05-01	0.933188
1	2015-05-02	0.933973
2	2015-05-03	0.932281
3	2015-05-04	0.945284
4	2015-05-05	0.925365

In [8]:
```
from sklearn import linear_model
import numpy as np

# fit degradation curve
X = np.arange(0, stop = len(performance['ratio']))
y = np.array(performance['ratio'])
X = np.array(X).reshape(-1, 1)
y = np.array(y).reshape(-1, 1)

# Create linear regression object
regr = linear_model.LinearRegression()

# Linear fit
regr.fit(X, y)
y_pred = regr.predict(X)
last_day = performance['day'].iloc[X[-1]].iloc[0]
first_day = performance['day'].iloc[X[0]].iloc[0]
num_days = first_day - last_day
num_years = num_days / np.timedelta64(1, 'Y')
degradation = (y_pred[-1] - y_pred[0])/num_years * 100
print(f"Degradation rate: {degradation[0]}")
```

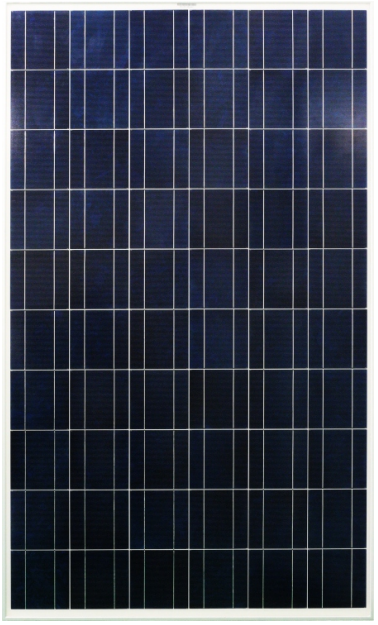
```
Degradation rate: 0.6492203199682552
```

Appendix B

Module Datasheet

Datasheet for the PV modules installed in the power plant [52].

BYD P6-30 Series



BYD, a fastest-growing green technology manufacturer, provides advanced PV products ranging from Wafer to PV Module. Based on its advanced technology, BYD Solar aims at Grid Parity, which can drive mass market-adoption of renewable energy.

Features

- High efficiency-BYD has achieved 17.4% efficiency
- Excellent optical performance
- Easy to be installed
- Strong frame module, passing mechanical load test of 5400Pa

Warranty

- 10 years for product
- 10 years on 90% for performance
- 25 years on 80% for performance

Recommended Applications

- Residential roof top systems
- On-grid commercial systems
- On-grid utility systems
- Off-grid commercial systems
- Off-grid utility systems

Certificates

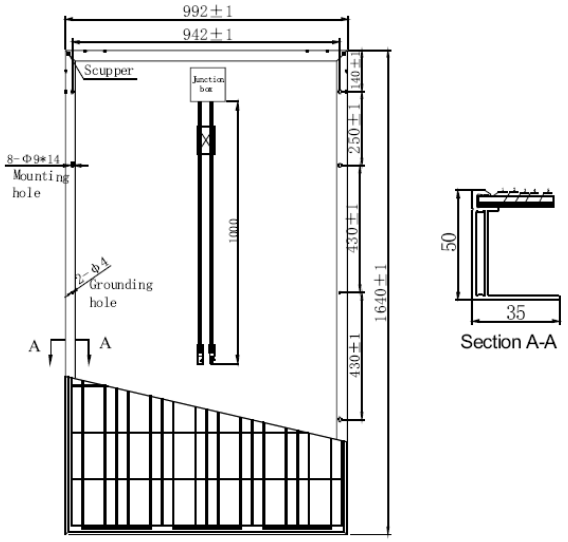
UL1703, CE, TUV, IEC 61215, IEC 61730, PV Cycle, MCS and CEC.
ISO9001:2008, ISO14001:2004

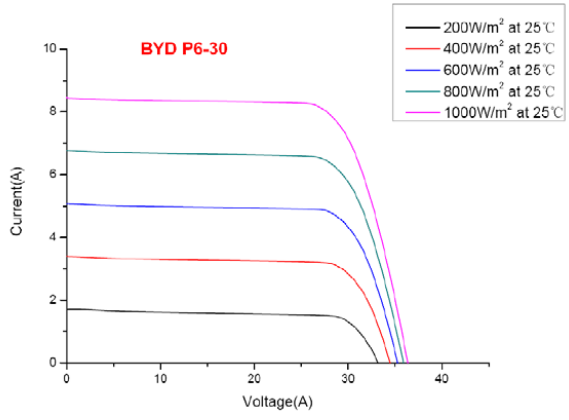
Model	BYD 220P6-30	BYD 225P6-30	BYD 230P6-30	BYD 235P6-30	BYD 240P6-30	BYD 245P6-30	BYD 250P6-30
Open Circuit Voltage (Voc)	36.18 V	36.36 V	36.75 V	37.07 V	37.54 V	37.80 V	38.00 V
Maximum Operating Voltage (Vmp)	28.29 V	28.49 V	28.67 V	29.06 V	29.55 V	30.06 V	30.40 V
Short Circuit Current (Isc)	8.40 A	8.44 A	8.50 A	8.69 A	8.90 A	8.94 A	8.98 A
Maximum Operating Current (Imp)	7.80 A	7.91 A	8.02 A	8.09 A	8.12 A	8.15 A	8.22 A
Maximum Power in STC (Pmax)	220 Wp	225 Wp	230 Wp	235 Wp	240 Wp	245 Wp	250 Wp
Module Efficiency	13.52%	13.83%	14.14%	14.44%	14.75%	15.06%	15.37%
Operating Temperature	-40 ℃ ~ + 85 ℃						
Maximum System Voltage	1000 VDC(IEC) / 600 VDC (UL)						

■ STC: IRRADIANCE 1000W/㎡, Module Temperature 25℃, AM=1.5

BYD (Shanghai) Industrial Co.. Ltd Address: No.999 Xiangjing Road, Songjiang, Shanghai, 201611, P. R. China
TEL: +86-21-5777 8888 FAX: +86-21-5777 5086 Website: www.byd.com.cn, http://www.bydenergy.com

BYD COMPANY LIMITED





Temperature Coefficients	
NOCT	45 ℃ ± 2 ℃
Short-circuit current temperature coefficient	0.045%/ ℃
Open-circuit voltage temperature coefficient	-0.34%/ ℃
Peak power temperature coefficient	-0.47%/ ℃
Power tolerance	0~ 3%

NOCT: Nominal Operating Cell Temperature. The above data is only for reference.

Specifications	
Cell	Polycrystalline Silicon solar cells 156mm * 156 mm / 6 inch
No. of cells	60 (6 *10) pcs
Dimension of module	1640 mm * 992 mm * 50 mm / 64.6 inch * 39.1 inch *2.0 inch
Weight	19.6 kg / 43.21 lbs
Front Glass	3.2 mm (0.13 inch) tempered glass
Frame	Anodized aluminum alloy
Junction Box Protection Degree	IP65 rated
Plug connector protection degree	IP65 / IP67(MC 4)
Bypass-Diodes	6 pcs. (IEC) / 3 pcs. (UL)
Max. Fuse Current Rating	15 A
Type of Connector	MC4,MC4 compatible,MC3 compatible, 0-1394462-4/6-1394461-2

Output Cables	
Cable Section Area	4 mm² / 0.0062 Sq in
Cable Length	2 * 1000 mm / 2 * 39.4 inch

Package Information	
Package	40' HC
Pcs/pallet	20
Pallet/container	28
Pcs/container	560

Appendix C

Animation Frames

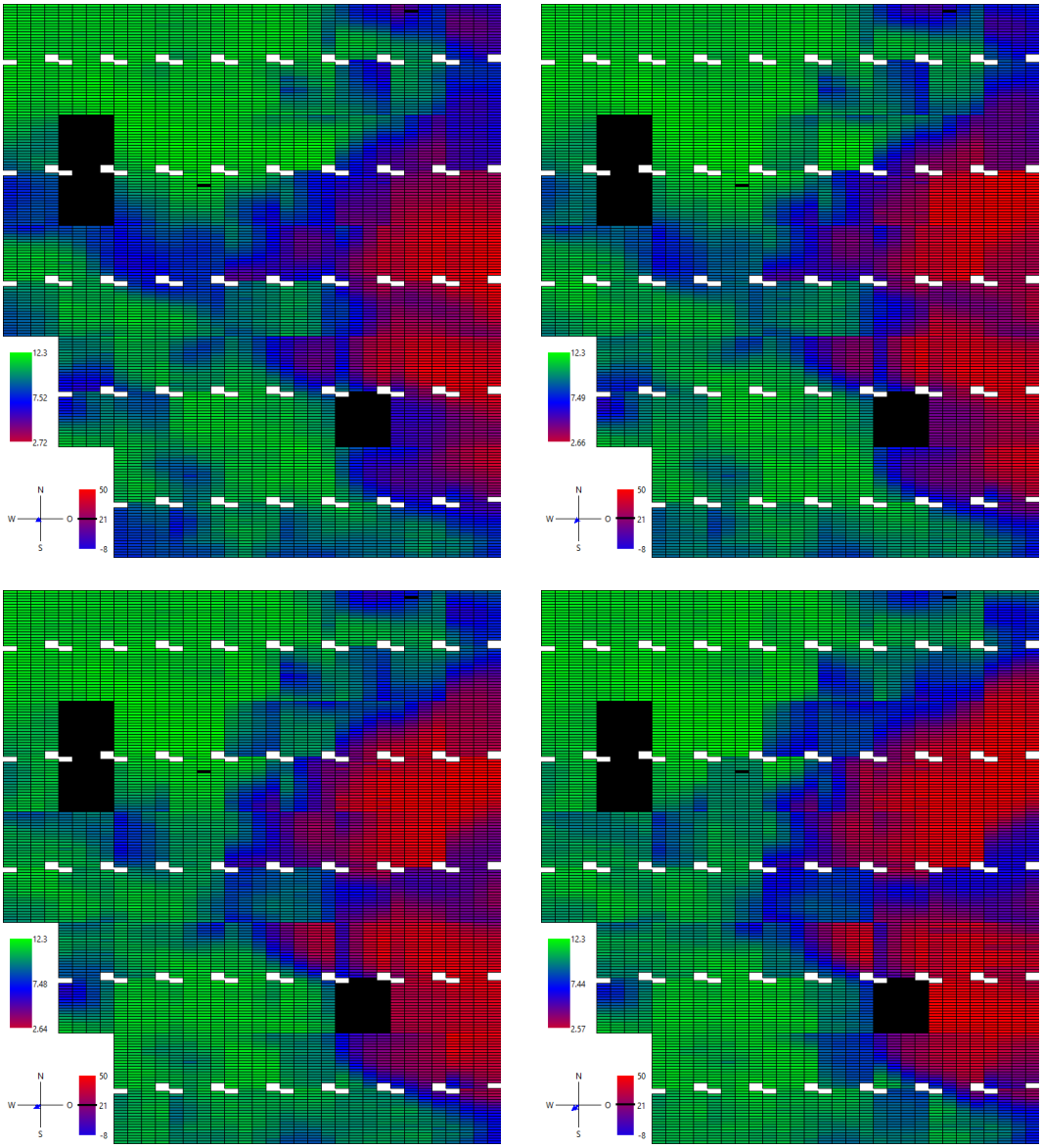


Figure C.1: Individual frames from the cloud movement animation in chapter 5.1.

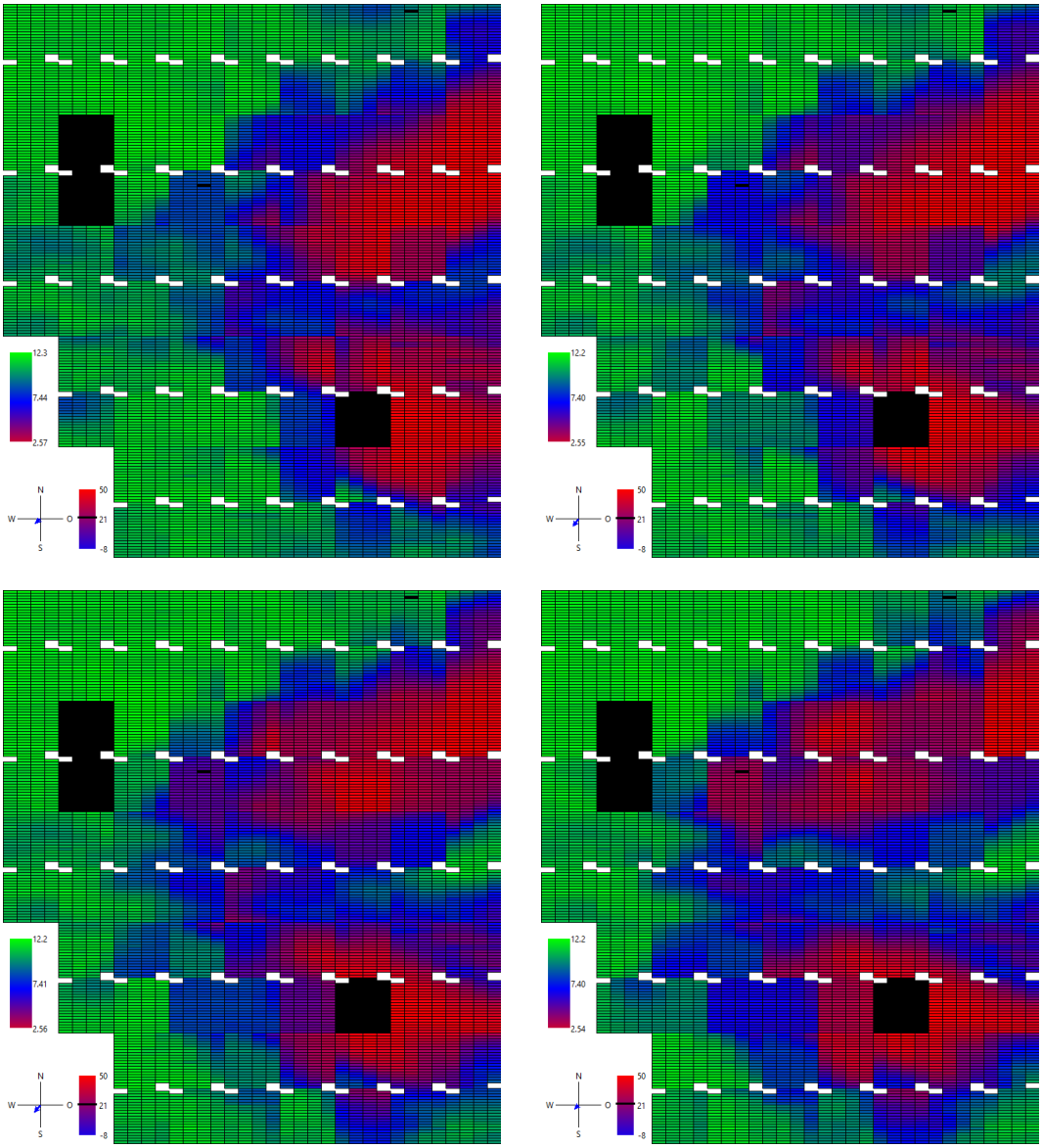


Figure C.2: Individual frames from the cloud movement animation in chapter 5.1 (continued.)

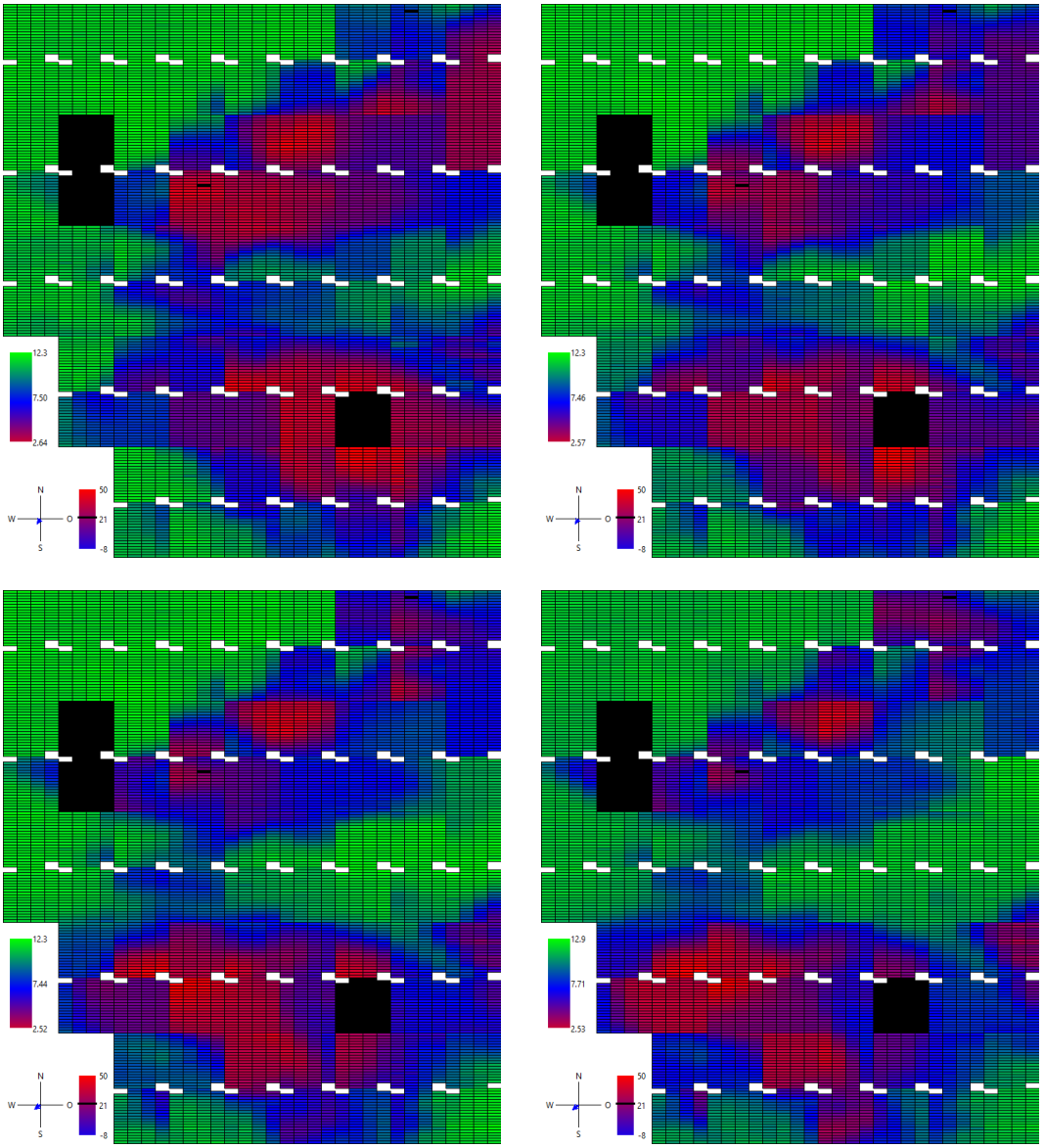


Figure C.3: Individual frames from the cloud movement animation in chapter 5.1 (continued..)

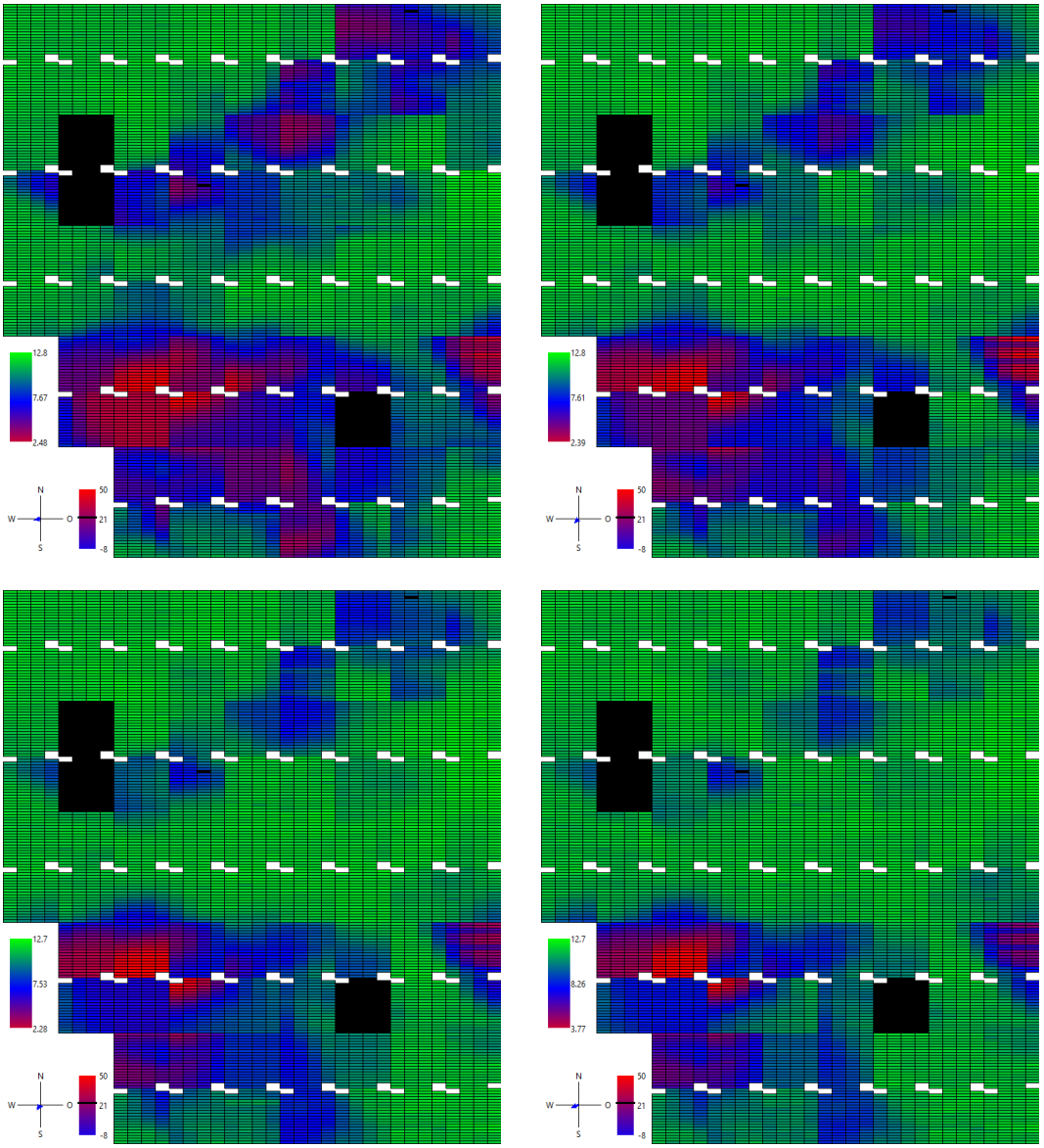


Figure C.4: Individual frames from the cloud movement animation in chapter 5.1 (continued...)

List of References

[1] Kannan, N. and Vakeesan, D.: Solar energy for future world:-a review. *Renewable and Sustainable Energy Reviews*, vol. 62, pp. 1092–1105, 2016.

[2] BP: Bp statistical review of world energy 2019. *BP Statistical Review of World Energy 2019.*, vol. 68, 2019.

[3] Timilsina, G.R., Kurdgelashvili, L. and Narbel, P.A.: *A review of solar energy: markets, economics and policies*. The World Bank, 2011.

[4] Renewables 2018. Oct 2018.
Available at: `https://www.iea.org/renewables2018/power/`

[5] National Renewable Energy Laboratory: *U.S. Solar Photovoltaic System Cost Benchmark: Q1 2018*. Oct 2018.
Available at: `https://www.nrel.gov/docs/fy19osti/72133.pdf`

[6] Electricity information 2019 overview. Jul 2019.
Available at: `https://www.iea.org/statistics/electricity/`

[7] of Energy, S.A.D.: *South African Energy Sector Report*. 2018.
Available at: `http://www.energy.gov.za/files/media/explained/2018-South-African-Energy-Sector-Report.pdf`

[8] Eberhard, A. and Naude, R.: The south african renewable energy independent power producer procurement programme: A review and lessons learned. *Journal of Energy in Southern Africa*, vol. 27, no. 4, pp. 1–14, 2016.

[9] Mellit, A., Tina, G.M. and Kalogirou, S.A.: Fault detection and diagnosis methods for photovoltaic systems: A review. *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 1–17, 2018.

[10] Fraunhofer, I.: Photovoltaics report. *Fraunhofer ISE, Freiburg*, 2019.

[11] DAO, S.: How PV Solar Plants Work? A Beginners Guide. September 2017.
Available at: `https://medium.com/@solar.dao/how-pv-solar-plants-work-a-beginners-guide-79f085b8ee88`

[12] Madeti, S.R. and Singh, S.: A comprehensive study on different types of faults and detection techniques for solar photovoltaic system. *Solar Energy*, vol. 158, pp. 161–185, 2017.

[13] Khatri, P., Hazrat, S.F., Butt, M.A. and Zaman, M.: Review of SCADA system for photovoltaic power plants. *International Journal of Creative Research Thoughts*, vol. 6, pp. 1565–1572, 2018.

[14] De Soto, W., Klein, S. and Beckman, W.: Improvement and validation of a model for photovoltaic array performance. *Solar energy*, vol. 80, no. 1, pp. 78–88, 2006.

[15] Drews, A., De Keizer, A., Beyer, H.G., Lorenz, E., Betcke, J., Van Sark, W., Heydenreich, W., Wiemken, E., Stettler, S., Toggweiler, P. *et al.*: Monitoring and remote failure detection of grid-connected pv systems based on satellite observations. *Solar energy*, vol. 81, no. 4, pp. 548–564, 2007.

[16] Platon, R., Martel, J., Woodruff, N. and Chau, T.Y.: Online fault detection in pv systems. *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1200–1207, 2015.

[17] Mekki, H., Mellit, A. and Salhi, H.: Artificial neural network-based modelling and fault detection of partial shaded photovoltaic modules. *Simulation Modelling Practice and Theory*, vol. 67, pp. 1–13, 2016.

[18] Chouder, A. and Silvestre, S.: Automatic supervision and fault detection of pv systems based on power losses analysis. *Energy conversion and Management*, vol. 51, no. 10, pp. 1929–1937, 2010.

[19] Pei, T. and Hao, X.: A fault detection method for photovoltaic systems based on voltage and current observation and evaluation. *Energies*, vol. 12, no. 9, p. 1712, 2019.

[20] Villalva, M.G., Gazoli, J.R. and Ruppert Filho, E.: Comprehensive approach to modeling and simulation of photovoltaic arrays. *IEEE Transactions on power electronics*, vol. 24, no. 5, pp. 1198–1208, 2009.

[21] Batzelis, E.: Non-iterative methods for the extraction of the single-diode model parameters of photovoltaic modules: A review and comparative assessment. *Energies*, vol. 12, no. 3, p. 358, 2019.

[22] Hansen, C.: Parameter estimation for single diode models of photovoltaic modules. *Sandia National Laboratories, Albuquerque, NM, Forthcoming*, 2015.

[23] Dobos, A.P.: An improved coefficient calculator for the california energy commission 6 parameter photovoltaic module model. *Journal of solar energy engineering*, vol. 134, no. 2, p. 021011, 2012.

[24] Reis, L., Camacho, J. and Novacki, D.: The newton raphson method in the extraction of parameters of pv modules. In: *Proceedings of the International Conference on Renewable Energies and Power Quality (ICREPQ'17), Malaga, Spain*, pp. 4–6. 2017.

[25] Holmgren, W.F., Andrews, R.W., Lorenzo, A.T. and Stein, J.S.: Pvlib python 2015. In: *2015 ieee 42nd photovoltaic specialist conference (pvsc)*, pp. 1–5. IEEE, 2015.

[26] Stein, J.S., Holmgren, W.F., Forbess, J. and Hansen, C.W.: Pvlib: Open source photovoltaic performance modeling functions for matlab and python. In: *2016 ieee 43rd photovoltaic specialists conference (pvsc)*, pp. 3425–3430. IEEE, 2016.

[27] Gurupira, T. and Rix, A.: Pv simulation software comparisons: Pvsyst nrel sam and pvlib. In: *SAUPEC 2017*, pp. 1–6. 2017.

[28] SMA Solar: *SUNNY CENTRAL 800CP XT / 850CP XT / 900CP XT*. SMA Solar Technology AG, .
Available at: `http://files.sma.de/dl/18859/SC800CP-900CP-DEN1751-V23web.pdf`

[29] Modbus, I.: Modbus application protocol specification v1. 1a. *North Grafton, Massachusetts (www. modbus. org/specs. php)*, 2004.

[30] SMA Solar: *SUNNY STRING-MONITOR SSM16-11/SSM24-11*. SMA Solar Technology AG, .
Available at: `https://files.sma.de/dl/9751/SSM16-24-IA-IEN105120.pdf`

[31] Firth, S.K., Lomas, K.J. and Rees, S.J.: A simple model of pv system performance and its use in fault detection. *Solar Energy*, vol. 84, no. 4, pp. 624–635, 2010.

[32] Breitenstein, O., Bauer, J., Kwapil, W., Lausch, D., Rau, U., Schmidt, J., Schneemann, M., Schubert, M., Wagner, J.-M. and Warta, W.: Understanding junction breakdown in multicrystalline solar cells. 2010.

[33] SMA Solar: *SUNNY CENTRAL 500CP XT/630CP XT/720CP XT/760CP XT/800CP XT/850CP XT/900CP XT/1000CP XT*. SMA Solar Technology AG, .
Available at: `https://files.sma.de/dl/18857/SCCPXT-E7-BA-en-58.pdf`

[34] Tsanakas, J.A., Vannier, G., Plissonnier, A., Ha, D.L. and Barruel, F.: Fault diagnosis and classification of large-scale photovoltaic plants through aerial orthophoto thermal mapping. In: *Proceedings of the 31st European Photovoltaic Solar Energy Conference and Exhibition 2015*, pp. 1783–1788. 2015.

[35] Tsanakas, J.A., Ha, L. and Buerhop, C.: Faults and infrared thermographic diagnosis in operating c-si photovoltaic modules: A review of research and future challenges. *Renewable and sustainable energy reviews*, vol. 62, pp. 695–709, 2016.

[36] Takashima, T., Yamaguchi, J., Otani, K., Oozeki, T., Kato, K. and Ishida, M.: Experimental studies of fault location in pv module strings. *Solar Energy Materials and Solar Cells*, vol. 93, no. 6-7, pp. 1079–1082, 2009.

[37] Vergura, S., Acciani, G., Amoruso, V. and Patrono, G.: Inferential statistics for monitoring and fault forecasting of pv plants. In: *2008 IEEE International Symposium on Industrial Electronics*, pp. 2414–2419. IEEE, 2008.

[38] Lorenz, E., Betcke, J., Drews, A., Heinemann, D., Toggweiler, P., Stettler, S., van Sark, W., Heilscher, G., Wiemken, E., Heydenreich, W. and Beyer, H.G.: Pvsat-2: Intelligent performance check of pv system operation based on satellite data. 06 2004.

[39] Pelland, S., Galanis, G. and Kallos, G.: Solar and photovoltaic forecasting through post-processing of the global environmental multiscale numerical weather prediction model. *Progress in Photovoltaics: Research and Applications*, vol. 21, no. 3, pp. 284–296, 2013.

[40] Ducange, P., Fazzolari, M., Lazzerini, B. and Marcelloni, F.: An intelligent system for detecting faults in photovoltaic fields. In: *2011 11th International Conference on Intelligent Systems Design and Applications*, pp. 1341–1346. IEEE, 2011.

[41] Chouay, Y. and Ouassaid, M.: An intelligent method for fault diagnosis in photovoltaic systems. In: *2017 International Conference on Electrical and Information Technologies (ICEIT)*, pp. 1–5. IEEE, 2017.

[42] Serra, J. and Arcos, J.L.: An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems*, vol. 67, pp. 305–314, 2014.

[43] Sakoe, H., Chiba, S., Waibel, A. and Lee, K.: Dynamic programming algorithm optimization for spoken word recognition. *Readings in speech recognition*, vol. 159, p. 224, 1990.

[44] Shou, Y., Mamoulis, N. and Cheung, D.W.: Fast and exact warping of time series using adaptive segmental approximations. *Machine Learning*, vol. 58, no. 2-3, pp. 231–267, 2005.

[45] SMA Solar: *Webconnect Systems in SUNNY PORTAL*. SMA Solar Technology AG, 2018.
Available at: `https://files.sma.de/dl/18915/SPortalWebcon-BA-en-14.pdf`

[46] Shipman, J.W.: Tkinter 8.4 reference: a gui for python. *New Mexico Tech Computer Center*, 2013.

[47] Jordan, D.C. and Kurtz, S.R.: Photovoltaic degradation rates - an analytical review. *Progress in photovoltaics: Research and Applications*, vol. 21, no. 1, pp. 12–29, 2013.

[48] Meyer, E.L. and Van Dyk, E.E.: Assessing the reliability and degradation of photovoltaic module performance parameters. *IEEE Transactions on reliability*, vol. 53, no. 1, pp. 83–92, 2004.

[49] Phinikarides, A., Kindyni, N., Makrides, G. and Georghiou, G.E.: Review of photovoltaic degradation rate methodologies. *Renewable and Sustainable Energy Reviews*, vol. 40, pp. 143–152, 2014.

[50] Standard, B. *et al.*: Photovoltaic system performance monitoring-guidelines for measurement, data exchange and analysis. *BS EN*, vol. 61724, 1998.

[51] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. *et al.*: Scikit-learn: Machine learning in python. *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[52] BYD Company Limited: *BYD P6-30 Series*. BYD Company Limited. Available at: `https://www.zonnepanelen.net/nl/pdf/panels/BYD_220-250P.pdf`