PV System Monitoring and Fault Detection using Peer Systems

Maitheli Makarand Nikam

PV System Monitoring and Fault Detection using Peer Systems

by

Maitheli Makarand Nikam

in partial fulfilment of the requirements for the degree of Master of Science
in Sustainable Energy Technology at the Delft University of Technology,
to be defended publicly on Friday, 20th of August, 2021 at 14:00 hour.

Student number:	5108853		
Project duration:	Dec 14, 2020 – Aug 20, 2021		
Thesis committee:	dr. ir. H. Ziar,	TU Delft	Supervisor
	ir. A. Alcañiz,	TU Delft	Daily supervisor
	ir. Y. Snow,	Solar Monkey	Daily supervisor
	dr. ir. I. Gordon,	TU Delft	Internal committee member
	dr. ir. J. L. Cremer,	TU Delft	External committee member

Delft University of Technology

Preface

This graduation report is the result of my thesis research project that was fulfilled as a requirement to obtain the Master of Sustainable Energy Technology degree at the Technical University of Delft. The research project was carried out in collaboration with Solar Monkey from December 2020 - August 2021. With this project, I got an opportunity to apply the knowledge I obtained during the master program, resulting in a Machine Learning model that monitors any residential PV system using data from similar systems.

This result would not be possible without the support and guidance of several people. Therefore, I would like to thank everyone who assisted me during my research project. First of all, my supervisors at TU Delft, dr. ir. Hesan Ziar and ir. Alba Alcañiz. Thank you for your constructive criticism and feedback. It helped me push the boundaries of my knowledge and grow. I appreciate how generously they made time for this project. Furthermore, I would like to thank the members of my graduation committee, dr. I. Gordon and dr. J.L. Cremer as well as the rest of the PVMD group for their feedback during the presentations.

Special thanks to my external supervisor at Solar Monkey ir. Yitzi Snow. He gave me the opportunity to get to know the company and other projects of Solar Monkey. We frequently discussed the contents of my project and without his constant support this project would not have been possible. Yitzi and Alba had the patience to support me and answer all my questions, which greatly increased the value of the results and my thesis experience.

Within Solar Monkey, I want to thank the rest of the R&D team, Jaap Donker, Rohi Perlsteyn and Amr Elsherbini. Their feedback and out-of-the-box insights during our weekly meetings were greatly appreciated and helped progress the work in the right direction. Special thanks to Eric van den Broek for helping me out with the fault detection part of the thesis. I would also like to thank everyone else from the Solar Monkey team for making my time and experience at the company very enjoyable. I will gladly carry the company values with me forward.

I would like to express my deepest gratitude to my friends in the Netherlands for their support in the past two years. The discussions we had about machine learning, renewable energy, sustainability, and veganism truly enriched my experience. Also, incredibly thankful to my friends in India, especially 'aaple lok', for always encouraging me through thick and thin despite the long distance and a pandemic.

Last but not least, I want to thank my family, especially my parents and my sister, for their enormous emotional and financial support throughout my studies. Thank you for making everything possible.

Maitheli Makarand Nikam
Delft, August 2021

Abstract

Solar energy is an abundant, scalable, and clean source of energy. With an exponential drop in prices of PV modules, more and more rooftop photovoltaic (PV) systems are being installed worldwide. Since these small-scale PV systems do not use expensive sensors, it is difficult to detect malfunctions for these systems. This could lead to lower energy generation along with financial losses for the owners. Thus, a method for PV yield monitoring is developed for early and remote fault detection. This method does not use the conventional analytical approach as it depends on inaccurately extrapolated weather data. Instead, the proposed method uses data from similar or neighboring or peer PV systems for estimating the expected energy generation. By comparing the expected energy generation with actual energy generation, a faulty system can be flagged. In this project, information from about 12000 PV systems is used, which includes system design information such as location, number of panels, panel orientation, etc. along with the historical daily energy generation for periods ranging from two months to up to seven years per system. This data is pre-processed to improve its quality and then split into training data and testing data.

In this thesis, a machine learning model was developed for predicting energy yields, which uses Genetic Algorithm (GA) for optimization. This model splits the available data into system design, system location and system yield data. Thus, the model uses these as criteria for finding PV systems similar to the monitored system. Once good peer systems are located, system yield data of those systems is used for estimating the expected energy yields of the monitored system. The three criteria used by the model do not have equal influence on finding good peers, thus, the model had to be trained or optimization was done using the training data. Post optimization using the genetic algorithm, the relative influence of system design:system yield:system location was found to be 0.125:0.875:0 with on average 16 good peers needed for accurate predictions. The proposed model has a mean normalized RMSE of 0.057 and about 95% of the systems tested had an R^2 score higher than 0.85. The existing commercial software at Solar Monkey has a mean normalized RMSE of 0.082 and about 83% of the systems tested had an R^2 score higher than 0.85. It was found via experiments that for the best results a minimum of 1700 years of data is needed for training. This 1700 years of data could be in the form of 850 PV systems each with two years of data or any other combination. However, a minimum of 6-9 months of data per PV system is needed.

The predicted energy generation calculated by the proposed model is compared with the actual energy generation to detect any malfunctions that may have occurred in the monitored system. Using the predictions from the proposed model and some other models previously developed by Solar Monkey, 120 randomly chosen PV systems were analysed for faults. Based on this, a semi-automatic categorization framework was created with the proposed model as one of the criteria to detect common faults in the system such as missing data, under-performance, over-performance and false positives. Using the categorization framework, certain PV systems were found as interesting examples for under-performance with broken panels or string, over-performance with system size change and false positives. The model is especially useful for separating system design mismatch from actual system malfunctions. With the framework, it was shown how the proposed peer-to-peer model can be used for fault detection along with certain other models.

Contents

List of Figures

List of Tables

Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Networks
$ARIMA$	Auto-regressive Integrated Moving Average
$ARMA$	Auto-regressive Moving Average
AZ	Arizona
CCS	Carbon Capture and Storage
$COBYLA$	Constrained Optimization BY Linear Approximation
CUF	Capacity Utilization Factor
CUR	Capacity Utilization Ratio
DHI	Diffuse Horizontal Irradiance
DNI	Direct Normal Irradiance
EA	Evolutionary Algorithms
FPS	Fitness Proportionate Selection
GA	Genetic Algorithm
GHG	Greenhouse Gases
GHI	Global Horrizontal Irradiance
INF	Infinity
$k-NN$	k-Nearest Neighbours
$KNMI$	Royal Netherlands Meteorological Institute
MAD	Median Absolute Deviation
MAE	Mean Absolute Error
$MAPE$	Mean Absolute Percentage Error
ML	Machine Learning
MSE	Mean Squared Error
NAN	Not-A-Number
$NARX$	Non-linear Auto-regressive Exogenous
$NMAE$	Normalized Mean Absolute Error
$NRMSE$	Normalized Root Mean Squared Error
$P2P$	Performance to Peers
POA	Plane of Array

PR	Performance Ratio
PSO	Particle Swarm Optimization
PV	Photovoltaic
$PVGIS$	Photovoltaic Geographical Information System
$PVMD$	Photovoltaic Materials and Devices
RD	Research and Development
RF	Random Forrests
$RMSE$	Root Mean Squared Error
$SAPM$	Sandia Array Performance Model
$SLSQP$	Sequential Least Squares Programming
SPA	System Performance Analysis
SVM	Support Vector Machines
WP	Watt Peak

1 Introduction

Over the last 170 years, the concentration of carbon dioxide in the atmosphere has raised by 48% above pre-industrial revolution levels due to human activities [1]. The rise in levels, as seen in Figure 1.1, are higher compared to the natural increase over a twenty thousand year period. The industrial revolution of the 18^{th} and 19^{th} century has changed the world technologically, socioeconomically and culturally [2]. It also changed how humans access and use energy with the use of new energy sources such as fossil fuels and internal combustion engines.

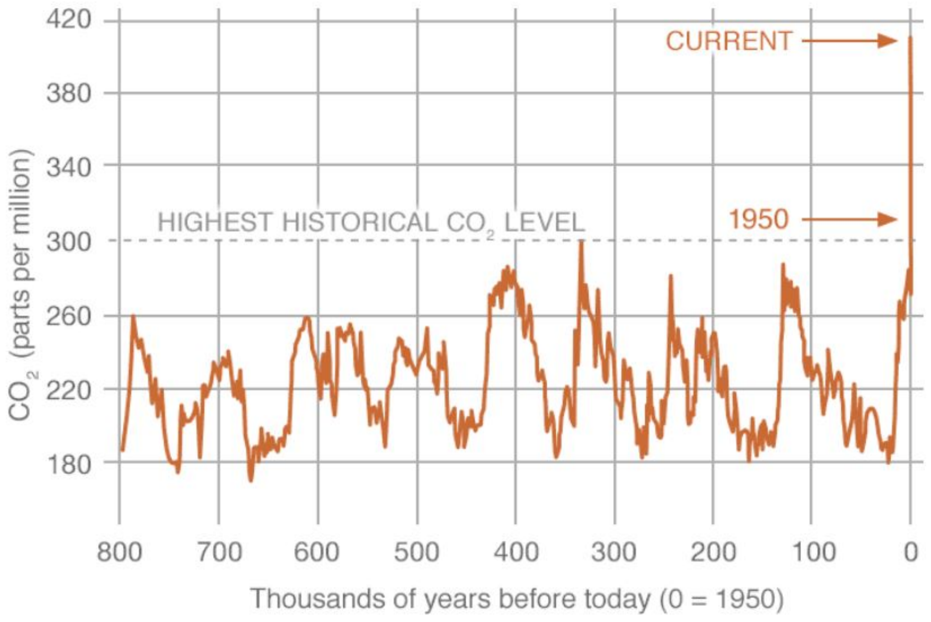


Figure 1.1: Historical concentrations of carbon dioxide in the atmosphere [1]

Along with the post-industrial revolution prosperity and access to cheap abundant energy, came the emissions of Greenhouse Gases (GHGs) including carbon dioxide released due to deforestation and burning fossil fuels for transportation and industry [2]. The increase in population with better health care and higher life expectancy led to an increase in demand for energy from these fossil fuels and thus, more deforestation. This spectacular development has led to climate change with an increase in global temperatures, rising sea levels and loss of biodiversity. Thanks to the efforts of many climate activists, in 2015, an agreement was signed in Paris by 176 countries to keep the global temperature rise below 2^{o}C [3][4]. While each country is free to choose the path towards this goal, the building blocks are the same with promising new technologies such as photovoltaic (PV) technology, wind turbines, batteries, carbon capture and storage (CCS), etc. [3].

1.1. Photovoltaic Technology

The sun is one of the most consistent sources of energy. Solar energy is an abundant, increasingly affordable, scalable, and clean source of energy [5]. It is possible to generate electricity from solar energy using photovoltaic panels with no direct emissions, i.e., it does not have any GHG emissions during use but only due to the manufacturing and installation processes. With non-renewable energy sources set to decline and an exponential drop in prices of PV modules, it has the potential to supply electricity that is environmentally as well as economically attractive [6]. One other advantage of solar energy is that it is decentralized, meaning it is possible to have rooftop solar energy systems close to the electricity load or solar micro-grids that can power entire communities. Thus, the market for residential rooftop photovoltaic systems has been growing each year in the Netherlands and the rest of Europe [6]. To tap into the large potential of solar energy, a need has arisen for better designing of PV systems and monitoring them. This is where Solar Monkey steps in in the market of residential PV systems.

Solar Monkey is a scale-up company, with a goal to accelerate the energy transition to renewable energies (here solar PV) by offering a software to PV installers that allows them to accurately design a PV system within a few minutes [7]. It is particularly developed to design PV systems for residential home owners. The software places the panels on roof images and utilises aerial and height-imagery of the installation site to account for the sky view factor and shading losses. In this way, they ensure optimal accuracy of the results.

In 2019, Solar Monkey successfully moved on from the startup incubator YES! Delft to The Hague Tech in Den Haag, Netherlands. The company is growing rapidly and as of today, Solar Monkey is operating in three countries (the Netherlands, Belgium, and Spain) and monitors about 15000 systems daily [7].

1.2. PV Monitoring & Fault Detection

With the growth of the market for PV panels, came the need to monitor them. Monitoring of PV panels includes activities like observing, checking energy yields as well as cleaning, repairing the broken panels, etc. Under monitoring, in this thesis, the energy generated by the panels is observed and compared with the expected energy generation in order to detect any fault in the PV panels. Monitoring of solar PV systems is necessary to maximize the energy yield of the system. The conventional approach used in monitoring of entire PV systems, is to use design information along with accurate weather data. This is suitable for large or commercial solar farms where expensive sensors can be installed on site. These sensors are generally used for measuring irradiance, temperature, wind speed, etc. However, for residential PV systems, extrapolated weather data has to be used. This rarely represents the local weather as the weather stations are often far away from the actual site of PV system installation. Thus, local phenomena like a cloud passing over the area cannot be accounted for in the calculation of expected energy generation. This affects the accuracy of monitoring.

Again, Solar Monkey comes to the rescue as they provide "Zonnegarant" (Yield Guarantee). This is essentially a monitoring service which guarantees that the performance of the residential PV systems will not fall below a set expected limit. This is to ensure that the system generates the expected energy and to report system failures to residential owners in order to enable immediate action. Thus, monitoring and immediate detection of fault in the system is vital to maximize the energy generation of that system. Currently, Solar Monkey uses the conventional method based on extrapolated weather data for its predictions and monitoring. This thesis project aims to enhance and expand the monitoring services of the company.

To overcome the problem of weather data, a new approach was proposed for this project where peer systems, i.e. "neighbouring" PV systems will be used to monitor each other. A peer or "neighbouring" PV system is essentially a system similar to the monitored PV system. It could be physically nearby or have similar design parameters. As an example, if a cloud passes over one PV system, it will also pass over a geographically close peer system and thus, by comparing their yields, a faulty system can be flagged.

Another benefit of peer monitoring, is that it allows for more accurate monitoring in cases where, for example, the shading calculation is not accurate. Here, the peers may be better at predictions than the conventional approach. Peer monitoring could also be useful to separate the actual faults, i.e., malfunctions in the PV system from the design mismatch between the installed system and design in the Solar Monkey app. This is possible because peer monitoring depends on its own historical energy yields along with the peer predictions and not entirely on the system design parameters.

Inline with the motivation for this topic, a literature study was done of the research available on PV monitoring and fault detection as presented in next section.

1.3. Literature Study

A lot of research has been done previously on PV monitoring and fault detection. This research was helpful to elucidate the objective and scope of the project undertaken. This section provides a brief description of possible approaches found in the literature while working on this thesis. In general, most of the PV monitoring research has been done for large commercial PV farms that have higher monetary incentives than small residential PV systems. Nevertheless, abundant research about residential PV monitoring was found with few papers specifically tackling the problem using neighbouring or peer systems. Accordingly, first, a general overview of methods used for PV monitoring is discussed. This research provides an insight into different models developed already and what features are important when monitoring PV systems. Next, the literature specifically using neighboring systems is studied. The last part focuses on studies conducted for detecting anomalies in PV systems. The following review follows research conducted in multiple countries with varied climates and terrain. It is by no means exhaustive, but it does provide a good starting point for the task undertaken during this thesis.

Monitoring Approaches

Latest review studies conducted by Sobri et al., Kumar et al., and Antonanzas et al. provide a general overview of possible approaches to PV monitoring [8]–[10]. These approaches can be broadly classified into physical methods, time-series statistical methods, Artificial Intelligence (AI) methods, and ensemble methods. Some of the physical models follow an analytical approach similar to ones described in Smets et al. [5]. It is a combination of multiple models implemented together to describe the relationship between physical state of the system in the atmosphere and its power production. This model may consist of sub-models such as irradiance decomposition models, clear sky models, satellite imaging models, numerical weather prediction models, etc. Some of the sub-models need high frequency temporal data and/or finer spatial data. However, for residential PV installations per minute or per second data is not feasible without expensive measuring devices. This makes it difficult to use analytical models without compromising on the accuracy of predictions.

The second possible approach is the time-series based statistical method. A time-series is a set of observations of a parameter measured successively in time. These methods reconstruct a relationship between hourly irradiance and past meteorological parameters [8]. This approach includes models such as Regression, Auto-regressive Moving Average (ARMA), Auto-regressive Integrated Moving Average (ARIMA), etc. These methods measure and define the correlations between a dependent parameter and independent (predictor) parameters. The advantage of using an analytical method over a statistical one is that no historical data is needed, thus, it is possible to calculate the power output of a system prior to installation [10]. On the other hand, statistical models do not need any internal system information. It is a purely data-driven approach which is able to define correlations between past data to predict the future behavior of the system.

The third approach highly suggested by Antonanzas et al., is the AI based approach using Artificial Neural Networks (ANN), Support Vector Machines (SVM), k-Nearest Neighbours (k-NN) and Random Forrests (RF) [10]. According to them, about 24% of the studies use ANN techniques. Most recent papers about PV monitoring also recommend and use one or the other machine learning techniques [11]. A research was conducted by Daniel Grzebyk at Photovoltaic Materials and Devices (PVMD) group in Delft University of Technology and Solar Monkey in 2020 that followed this approach where the author examined six different models, namely, Solar Monkey analytical model, Polynomial Regression, Random Forest, ElasticNet, a simple persistence model and Extreme Gradient Boosting; out of which Extreme Gradient Boosting was selected as the best [12].

The final approach according to Sobri et al. is an ensemble or hybrid method which combines analytical models with machine learning or statistical methods. Bouzerdoum et al. and Ramsami et al. are good examples of combining either two or more statistical models or combining a statistical model with the analytical model [13], [14]. These models usually rely on analytical models for pre-processing of data and employ statistical or AI methods for yield forecasting.

Neighbouring PV Systems
According to Antonanzas et al., none of the studies on commercial PV plants that used data from nearby plants provide satisfactory results for short lead times (less than 15 minutes) for forecasting [10]. This was due to the low spatio-temporal correlation with other PV plants. The research by Berdugo et al. considered 11 PV plants which were up to 70 km apart to generate patterns in energy yields [15]. They claimed that if and when more neighbours are available, it was possible for the collaborative method to outperform the non-collaborative approach. On commercial PV plant level, two other studies conducted by Yang et al. and Vaz et al. corroborated this claim [16], [17]. Yang et al. used data from 5 nearby plants as far as 200 km apart to predict solar iradiation. They further used an analytical model for energy generation [16]. Vaz et al. also studied 5 nearby plants but only 7 km apart in the Netherlands. They used ANN along with a Non-linear Auto-regressive Exogenous (NARX) model for solar power forecasting [17]. Both studies found that for horizons of 1 hour and 30 minutes respectively, the model using neighbouring plants outperformed the baseline models. One more interesting research was by Mallor et al., where they developed a model to automatically detect outliers or faults in energy production of identical (not only similar) sets of arrays in one PV farm [18]. In the first unsupervised step, by using outlier detection, 'in control' or clean energy production data is created. In the second step, by using both parametric and three non-parametric methods, control charts are created that can be used for fault detection.

For this project, it is more interesting to look at research done for residential small-scale PV systems. Lonij et al. reported a new method for forecasting power output under cloudy skies using ground-based irradiance sensors in Tucson, AZ region [19]. They considered 80 rooftop PV systems distributed over 2500 km^2 area. Their results outperformed the persistence model for forecast horizons from 30 to 90 minutes. Their model was based on an analytical approach where they introduced a new framework for spatio-temporal correlation of clouds. Another paper Elsinga et al. studied the correlation for 25 rooftop PV systems in the Netherlands within 100 km^2 for little over a year to account for seasonal variances [20]. They created an exponential semi-variogram that can assess the increase in similarity between two random variables as the distance between these variables decreases [21]. They also concluded that for their exponential model, the decorrelation distance, beyond which power fluctuation variability becomes independent of distance, ranges between 100 m and 15 km. This research was further continued by the same group, where they developed a peer-to-peer forecasting method to provide local short-term GHI forecasts to be used for forecasting the power generated by a network of PV systems with forecast horizons between 5 and 8 minutes [22]. All the above papers in one way or the other track the cloud motion in order to forecast the ground irradiance variations which are then used to forecast energy production using analytical models.

On the other hand, Golnas et al., Tsafarakis et al. and Popovic et al. approached the problem differently. Golnas et al. developed a heuristic method that can predict the output of a single PV system from a regional fleet by using data from other systems in that fleet [23]. They correlated the historical performance and the inter-system distance in the fleet by building a NxN matrix that described the correlation of energy generated amongst N different systems. The coefficients of the matrix were calculated via model training, which were then used for estimating expected generation. Tsafarakis et al. developed a method for cluster analysis which can be applied to any pair of datasets which have a linear relationship [24]. The method can be applied to the monitoring of PV systems under the assumption that power produced by the system tends to be linearly related to plane-of-array (POA) irradiance or even power produced by neighboring PV systems. At times when the relationship becomes non-linear, the method can detect any malfunction that can cause changing energy loss like shading. This approach is more suited as fault detection rather than forecasting. Next, Popovic et al. presents a methodology for inter-system comparison between correlated PV systems to estimate the operation status of individual panel in urban surroundings [25]. It compared the estimated horizontal irradiance obtained analytically or via sensor measurements. The performance metrics calculated and their temporal properties can enable detection of operating conditions such as physical failure, ageing process, shading, soiling, etc.

In 2018, research was conducted in the PVMD group by R.C. Nijman on the topic of automatic identification of malfunctioning PV systems using PV yield data [26]. The author developed an open generic model for detecting PV systems with faults by comparing the yields of the monitored system with the yields of similar neighboring systems. They developed a metric yield ratio for the comparison, which is the ratio of yield of PV system and the maximum possible yield for the same system. The similar neighboring systems were found based on how close they were geographically to the monitored system and by comparison of system characteristics, namely, the orientation of the systems and whether or not the system was inverter limited. They monitored 1300 PV systems for two months, out of which 11% were identified to be malfunctioning.

Performance to Peers
Leloux et al. presented a novel performance indicator called Performance to Peers (P2P) instead of using Performance Ratio (PR) [27], [28]. P2P is calculated by comparing the energy production of several neighbouring PV systems and it can be used for automatic fault detection. The papers showed that P2P is more stable than PR especially in the presence of sub par system data. The procedure was developed on 6000 PV installations located across Europe with energy output data for approximately 7 years and a temporal resolution of 10 minutes. It followed three main steps: determination of which of the neighboring installations are the best peers, calculation of P2P from the comparison of energy generation of the monitored system with the best peers and application of P2P for fault detection. The first two steps have been used as inspiration for the model developed during this thesis. More details can be found in chapter 4.

For the third step of fault detection, the paper refers to the clustering method explained by Tsafarakis et al. [24]. They illustrate the use of P2P for four categories of faults namely zero energy faults, fault clusters, shading and excessive degradation. The model was validated on 30 systems where it was confirmed that for 29 out of 30 systems the faults detected were real.

Fault Diagnosis
For a list of faults encountered in PV systems, paper by Triki-Lahiani et al. was referred [29]. They divided the failure modes into three types as seen in Figure 1.2. The failure modes are also classified into: environmental cause failure such as soiling, shading and degradation due to light or heat; human error failure such as incorrect inclination or orientation, vandalism and improper connections; life cycle failure such as poor electrical isolation, weak structure and improper ventilation; and other causes such as cracks, leakage current and seal damage.

The above-mentioned failure modes are especially important for large-scale PV plants, however, for residential PV systems, not all of them are relevant. As explained, the papers by Mallor et al., Tsafarakis et al., Popovic et al. and Leloux et al. suggest different ways of diagnosing faults in

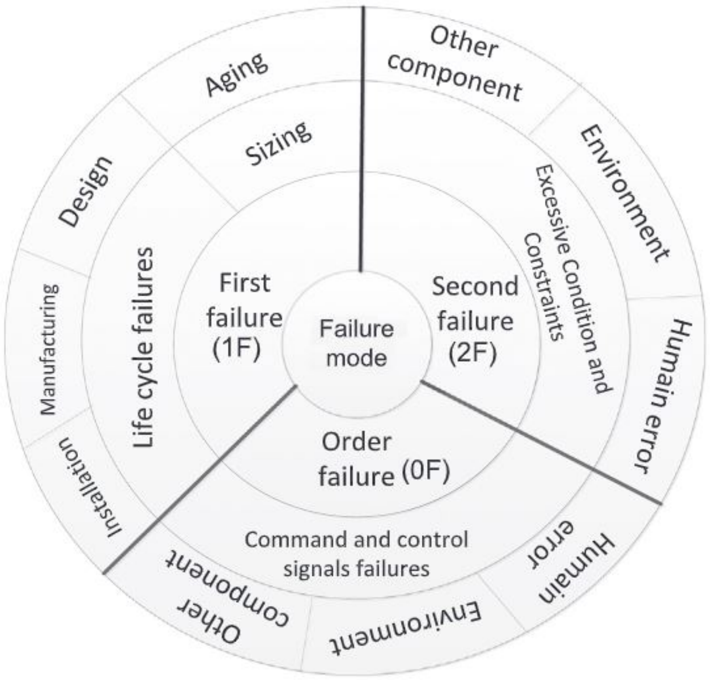


Figure 1.2: Failure modes observed in PV plants [29]

residential PV systems using their respective models [18], [24], [25], [28]. Accordingly, the common faults occurring in the residential sector would be inverter failure, shading, broken strings or modules, soiling, and performance degradation.

1.4. Objective

After a brief overview of the research done, the goals could be set clearly. The objective of the project is to analyse all available PV systems to identify which systems could be classified as peer systems. These systems could be similar in terms of design, total capacity, geographic location, or energy generation. Once the peer systems are chosen, their performance will be used to estimate the expected yields of the monitored PV system, also called the focus system. Furthermore, it will be checked whether the predictions made using peer systems are more accurate than those made using conventional methods or the methods available at Solar Monkey. This project also investigates how the model developed can be used for fault detection along with other models currently used by Solar Monkey. It is not the aim to develop a new automatic fault detection algorithm.

To achieve these objectives, the following research question will be answered in this report:

Can the accuracy of PV systems monitoring be improved by creating a model that can detect malfunctions of one system by comparing the data of this system with that of neighbouring/peer systems?

This research question was used as a guide for decision making in the development process of the model. The following sub-questions complemented the main research question:

- What methods presently exist for PV monitoring and which of them use the data from other similar systems?
- What criteria or what parameters can be used to find similar PV systems?

- How can the chosen parameters be used to predict the yields for a PV system?
- What is the possible way to apply and validate the proposed model?
- How can the model predictions by this approach be used to monitor and detect faults in a PV system?

As mentioned in section 1.3, a similar research has been done in the PVMD group in 2018 and by various other authors. By learning from the aforementioned papers, this project will dive deeper into using peer PV systems for monitoring and fault detection. It expands the monitoring database from two months data of 1300 PV systems to up to seven years data of about 12000 PV systems. This removes the seasonal dependency of the outcome and enables more robust monitoring and fault detection algorithms. In addition, a larger database enables studying features beyond just geographical closeness. For instance, systems which are geographically far apart but behave similarly are also categorized as good peers. In contrast to some research papers [19], [20], [22], [24], [25], some form of weather data is not used for finding similar systems, as this is extrapolated from the data of weather stations further away from the actual site of PV system. Also, only geographical location or only system characteristics or only energy yield data are not used for finding similar systems as done by other authors [23], [26], [28]. Furthermore, in line with the recommendations of Leloux et al. and Nijman, machine learning algorithms are used to increase the accuracy of the model by compensating for the assumed empirical relations used in the latter [26], [28].

The intended outcomes of this thesis are:

- A robust algorithm that identifies the best neighbouring systems for monitored PV system, i.e., to find peer systems
- An analysis on how neighbouring systems can be used to improve the accuracy of performance monitoring, i.e., to use peer systems to predict energy yields for focus system
- A comparison of the predicted energy yields with actual yields of the focus system to detect any faults in the system

To create the algorithm or model and for the analysis of the same, Python programming language was used on Jupyter Notebook platform. This choice was based on the the common use of Jupyter Notebook + Python by Solar Monkey and the knowledge of the author.

1.5. Outline of Thesis

To achieve the goal of this project and answer the research questions, the following report outline was created. The chapter 2 starts with data and data collection in section 2.1, followed by information about data pre-processing for feature selection and for ensuring good data quality in section 2.2. Additionally, some important concepts used for splitting the data for training and testing datasets are explained in section 2.3.

The chapter 3 explains all the theoretical concepts needed for the development and understanding of the model such as the objective function in section 3.1. Many types of optimization algorithms considered for this project are presented in section 3.2, whereas the chosen optimization algorithm i.e., genetic algorithm theory is explained in section 3.3. In section 3.4, the procedure for hyper-parameter tuning of optimization algorithms is explained. Followed by section 3.5, where the equations for the metrics used for evaluating the performance of the model are described.

The chapter 4 introduces the two models used for comparison and the proposed model developed during the course of this thesis. Thus, the Solar Monkey model, the literature model and the proposed model are described in detail in section 4.1, section 4.2 and section 4.3 respectively.

The results of the different models mentioned above are presented in chapter 5. This chapter has four sections. The first section 5.1 shows the results of optimization or model training using genetic algorithm. The second section 5.2 represents how the optimization results can be used for prediction of energy generation of a PV system. The third section 5.3 compares the predicted energy generations of the different models. Finally, section 5.4 has the results of experiments carried out to understand how the proposed model can be used.

In section 6.1 of chapter 6, the current fault detection approach used at Solar Monkey and the approach used in this thesis are explained. This is followed by a categorization framework developed for quick fault detection, explained in section 6.2. At last, in section 6.3 certain PV systems are presented as examples for diagnosis of common faults using the categorization framework.

The chapter 7 concludes this thesis report. The research questions posed at the beginning of the thesis and the intended outcomes are revisited in section 7.1 along with some recommendations for further product development and research in section 7.2.

2 Data

In this chapter, the available information which will be used as input to models is described. The model performance is largely dependent on the data, its quality and amount of data available. Not all information or features provided by Solar Monkey were used in order to limit either computational time or complexity. The data and its collection is further explained in section 2.1. Furthermore, the data pre-processing needed to ensure data quality can be found in the following section 2.2. Data pre-processing also includes feature selection from amongst the data collected. Next, the concept of data splitting used in data science is explained in the final section 2.3.

2.1. Data Collection

This project utilizes daily data resolution of PV systems for energy yield. While finer data resolution like hourly data can provide higher quality results, it is available only for a limited number of PV systems having SolarEdge inverters while the rest can log only daily system yield. Similarly, some PV systems using optimizer can provide energy yields for each module/string while others provide only a single total energy yield value. Thus, to ensure that the proposed model can be used for all PV systems in the database, the total daily energy yields of the entire PV systems are used. Overall, daily energy yields of 12229 active PV systems, for a period ranging from 60 - 2600 days (2 months to 7 years) up to 31^{st} December 2020 were available. As will be explained in chapter 6, for fault detection daily energy yields up to 30^{th} June 2021 were used. All these systems are located within the Netherlands. Figure 2.1 shows a spatial plot to observe the distribution of the 12229 PV systems across the Netherlands.

For each of those 12229 active PV systems, system information like location (latitude and longitude), total watt-peak, number of panels, inclination, orientation, etc., as shown in Table 2.1, are available. It is important to note that inclination and orientation here are the most common inclination and the most common orientation respectively, because for some PV systems, more than one value for inclination and/or orientation is possible. Along with that, yield information like the daily actual energy yield, expected daily energy yield (as per Solar Monkey model in section 4.1) and performance factor are also available for each PV system. This information was used for development of proposed model and comparing its performance with the model used by the company. Finally, some weather data like the irradiance, i.e., Global Horizontal Irradiance (GHI), temperature range, and average wind speed for each system are provided. However, since one of the objectives of this thesis is to overcome the shortcomings of using inaccurate weather data, weather data is not used for the model.

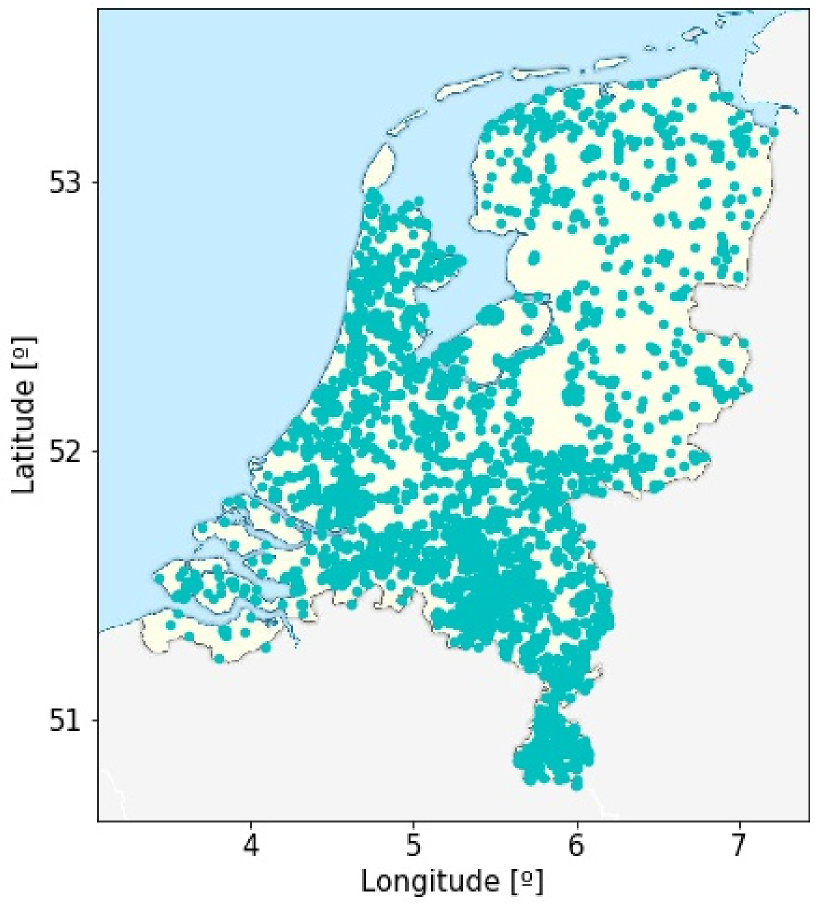


Figure 2.1: Spatial plot of all available PV systems in Netherlands

2.2. Data Pre-processing

2.2.1. Feature Selection

The proposed model built during this thesis project is based on Machine Learning (ML). Certain inputs called features are given to the Machine Learning model. While Table 2.1 lists all the features available, not all of them were used. As mentioned before, weather data is not used since the objective of the thesis is to overcome the problems due to inaccurate weather data. Features about yield data namely Solar Monkey expected yields and performance factor are essentially what the ML model aims to predict or calculate, hence, they are also not the inputs for the model. However, as explained in chapter 3, they are used for model training or optimization. These features are also used later in chapter 5 to compare the predictions of the ML model with those of the Solar Monkey model and to compare the performance.

After dropping the weather data and keeping some yield data aside, the remaining features were considered as input to the model. It is common in machine learning to use all available features as input to the ML algorithm of choice. However, the first step in this project is to find systems similar to the focus system (system to be monitored), i.e., to find peer systems. Thus, one of the parameters used to find peer systems is system data. Amongst the system data, features such as latitude, longitude, total watt-peak, panel count, inclination, orientation, and system age along with actual

System Data per System	**Unit**
Latitude	$^\circ$
Longitude	$^\circ$
Total watt-peak	Wp
Panel count	-
Inclination	$^\circ$
Orientation	$^\circ$
System age	$days$
Panel type	-
Panel efficiency	-
Inverter type	-
Inverter efficiency	-
Module decay per year	-
Yield Data per System per Day	**Unit**
Actual yield	kWh
Solar Monkey expected yield	kWh
Performance factor	-

Table 2.1: All available features/data

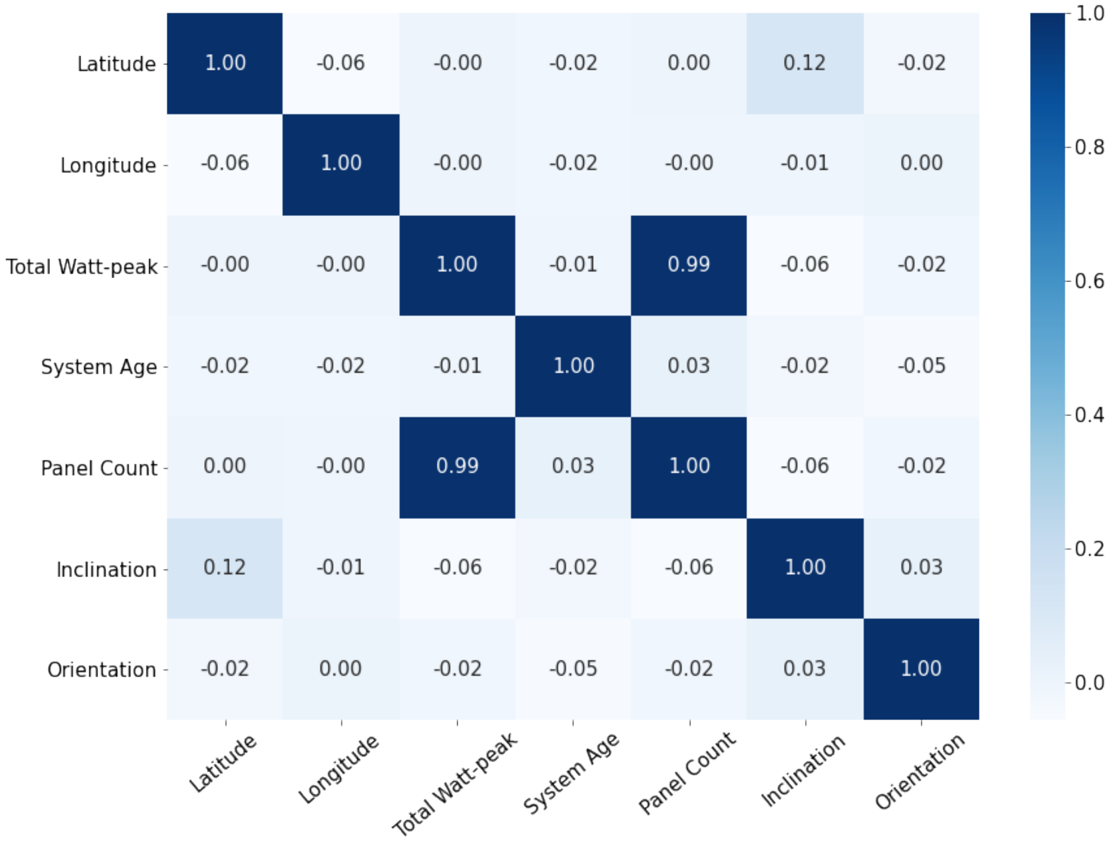


Figure 2.2: Feature correlation heatmap

yield were chosen as inputs. To limit the computational time, panel type and efficiency, inverter type and efficiency, module decay per year, were not considered. These features do affect the performance of a PV system but were considered not as relevant in the search for best peers. The most important aspects of these features are also correlated to the features chosen, for example, in the panel type, the watt-peak of the panel is important, which is correlated to the total watt-peak of the PV system.

After choosing the features, it is vital to check for correlation between them. It is desirable for machine learning models that the features are uncorrelated. The origins of the correlation problem in machine learning go back to needing to make the inverse of a matrix [30]. Thus, when making the inverse of a matrix, the determinant of the matrix needs to be not zero. If the features are linearly correlated, the determinant is zero. There are some machine learning methods which rely on these kind of calculations and perform worse if the features are correlated. Thus, a correlation heatmap was generated using all the chosen features as seen in Figure 2.2. As expected, panel count is highly correlated with the total watt-peak. Apart from that, none of the other features are correlated with each other.

2.2.2. Data Quality

Once the features to be used as input to the model are chosen, the next step is to ensure good model performance by using clean data for the training and testing. This is an important step in the pre-processing of data. First, the most common issues with yield data are found. According to the research conducted by Solar Monkey, they are classified as low outliers, high outliers, missing data, constant yield, and time-shifted data. Some of them are due to fault in the inverters such as reporting very low or no yield due to poor WiFi connection. It was possible to partly overcome this problem by filtering out systems with too much missing data as well as filtering out days with missing data. However, some corrupt data still exists in the dataset, which could affect the overall performance of the model. By filtering out systems with large amount of missing data, the total number of PV systems was reduced from 12229 to 9480 PV systems.

Secondly, there are errors in the system data such as inclination, orientation or panel count that occur due to unverified information provided to the software by customers or installers. Sometimes, a contractor installs the PV system different from the one designed, causing a mismatch in the results. This leads to a serious issue in the data quality. However, it is nearly impossible to detect these errors without actually visiting the site and verifying the design.

The data corruption and system design mismatches could be an issue during the training of the model as will be explained in chapter 5 and chapter 6. In the worst case, it could potentially lead to certain faults being left undetected, as a model trained on corrupt data will not consider that system malfunction as a problem. If a system malfunction goes undetected, it could continue to affect the performance of the system over a longer period of time. The impact of this can be seen in the results of fault detection in chapter 6. Unfortunately, it was not possible to ensure that all problematic data is filtered out.

2.3. Data Splitting

The first step in a machine learning model is to collect relevant data. Usually, the more data you have, the more trained your model will be and will give accurate results or in this case, predictions. The data is essentially the features that could potentially influence the predictions. These features need to be chosen wisely such that they are not correlated with each other. The data also needs to be cleaned before using i.e., before training. Thus, pre-processing is done on the data before it is split into training and testing, as training samples which contain corrupted data affect the performance of machine learning models. After pre-processing for this project, the data can be interpreted as a group of PV systems, each having certain information about the system design like panel count, orientation, etc. as well as daily energy yields for varying time-periods depending on the age of PV system.

Next, an important decision is to split all available data, here all PV systems, into Train data and Test data (refer Figure 2.3). This is to ensure that when evaluating the model, it is tested on new data that is different from the data used for training [31]. The **Training set** is used for model fitting. Thus, in this project, training set is the set of systems over which the relationship between the input features (location, daily yields, system design info, etc.) and the actual energy yield is defined. Once the model is trained, it is used to predict yields for the **Testing set** using only the input features for those systems and no actual energy yield. This set is for model evaluation where the predicted yield is compared with the actual yield to assess the performance of the model or monitor the system once the model is finalised.

Sometimes a **Validation set** is also defined which is used for adjusting the parameters of the model or fine tuning. Validation set can also be important to avoid overfitting of the model [31]. Overfitting is a condition when the trained model not only fits the input-output relationship, but also fits the noise in the output data from the training set. This leads to misleadingly high accuracy on the training dataset but low accuracy on the testing data previously unseen by the model. To avoid overfitting sometimes artificially noise is added to the training dataset. This set is not always necessary and was not part of this thesis. All the tuning was done on the training set itself. To verify that the model is not overfitting, the performance metrics for the training set are compared with the testing set. Ideally they should have similar performance [32].

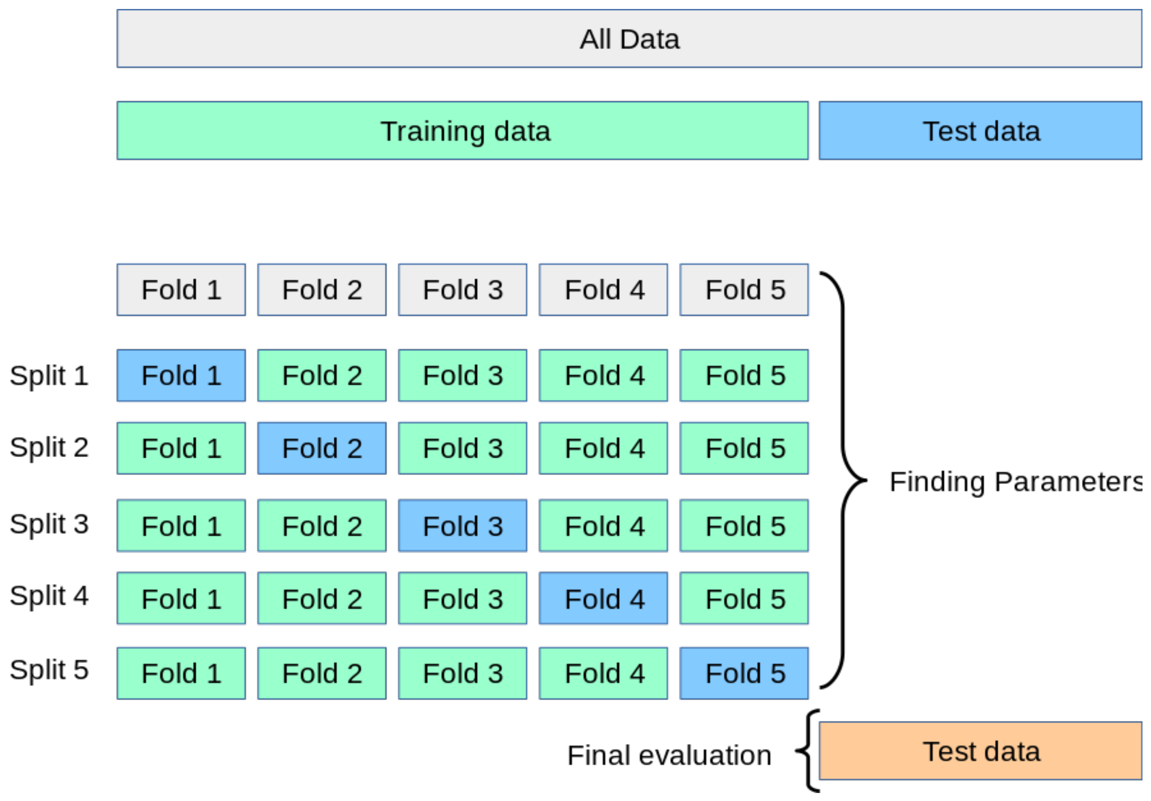


Figure 2.3: Splitting of data into training and testing datasets with cross-validation folds [33].

In many machine learning models, **Cross-validation** is used to evaluate the performance of the model, as seen in Figure 2.3. Cross-validation is where the data is divided into random sets called folds and over every iteration of the model, a different fold is used for testing. Therefore, a 5-fold cross-validation as shown in the figure splits the total dataset into 5 pieces, meaning 20% is used for testing and remainder 80% is used for training. The results of each iteration are used for finalizing the parameters [33]. Usually, the number of folds is optimized iteratively as well. At the end, the final evaluation of the model is performed on the test dataset. Cross-validation ensures that the systems are evenly or randomly distributed and the model performance is not over- or under-estimated due to good or bad systems respectively, present in training or testing data sets [34]. While cross-validation is an important tool, in this project, due to the shear amount of data handled and the computational memory and time it takes for training the model, it would have been very time and memory intensive to perform cross-validation at every step. Thus, all systems are randomly shuffled and then split into training and testing to avoid such bias. Along with this, it is verified that the system distribution in the training dataset and testing dataset are similar to the system distribution of the entire dataset. Thus, it ensures that both datasets actually represent the entire dataset. Although before finalizing the model, a 10-fold cross-validation was performed on the 9480 PV systems dataset. The outcome of cross-validation matches the results presented in chapter 5.

Traditionally, the train-test split ratio is 70:30 or for train-validation-test split it is 60:20:20 [35]. This is reasonable for smaller data sets containing hundred or thousand systems. However, with larger data sets like ten-thousand systems available, it is unreasonable to use 30% of those for testing. Thus, 98:2 for train-test or 98:1:1 for train-validation-test are used for large data sets in order to have as much data as possible for training the model [32]. Two percent of this large set is still sufficient for evaluating the model.

In this thesis, it was decided to start using a 90:10 system split between training and testing set to ensure that only a minimum required number of systems are used for testing. This was essential to manage the computational time needed for training. Once a reliable model was available, experiments were carried out to determine the ideal split between train and test systems (refer section 5.4 for results).

3 Theory

In the following sections, some important concepts used in data science and machine learning are introduced. This information will be useful to understand the model as well as to interpret the results and analysis. These concepts are explained considering readers with no prior experience. First, the key attributes of optimization such as objective function, bounds and constraints are explained in section 3.1. Second, in section 3.2 many optimization algorithms that were tried during the course of this thesis are explained. Next, the optimization algorithm chosen for this project is described in section 3.3. The following section 3.4 explains the process of tuning required for using the optimization algorithms. Finally, the metrics used for evaluating the goodness of model and the results are detailed in section 3.5.

3.1. Optimization Attributes

Optimization is an inherently mathematical concept. It is about minimizing or maximizing a certain mathematical function to arrive at the best possible or rather optimum solution to a problem. It involves selecting the variables that define the problem, determining a mathematical function that describes their relationship, defining which variables are to be optimized, and then finding the best set of values for those variables for an optimum solution [36]. Optimization problems are common in all kinds of domains, engineering or otherwise. However, regardless of the field, any optimization problem has the following features: set of variables, bounds, constraints, and objective function.

3.1.1. Optimization Variables

Optimization variables are decision variables whose values can be changed to find the optimal solution. A solution is a set of values that can be assigned to these optimization variables. A typical optimization problem, here the common Rosenbrock function, could look like Equation 3.1 [37].

$$minimize(f(x,y)) \tag{3.1}$$

where,

$$f(x,y) = (a-x)^2 + b(y-x^2)^2 \tag{3.2}$$

In the Equation 3.1, the optimization problem is to minimize $f(x,y)$ and thus, the optimization variables are x and y. All optimization problems need optimization variables to define them.

3.1.2. Bounds

Some optimization problems allow only a certain range of values for the optimization variables. Accordingly, bounds like $m \le x \le n$ define the range of values allowed for the variables. Bounds limit the set of possible solutions, also called as the search space [36]. Each variable can have a lower bound and an upper bound, however, they are not always necessary for completely defining an optimization problem. If left unspecified, positive infinity (+INF) is considered as the upper bound and/or negative infinity (-INF) is considered as lower bound, in other words, there are no bounds [37].

3.1.3. Constraints

Constraints define which of the solutions in the search space are feasible. It is a condition that the solution of the optimization problem must satisfy [36]. They are like bounds on functions of optimization variables, for example, $g(y) \geq k$. While constraints are also not always necessary to define an optimization problem, in some unconstrained problems, it is possible for the optimization to return absurd results that are not useful. Thus, they are often added to ensure that the solutions only include realistic and reasonable values.

3.1.4. Objective Function

An objective function, also known as the cost function, is the output that is to be maximized or minimized by optimization under certain constraints and bounds. It is used to compare solutions to decide which of them is the best [36]. It can be considered as the goal of the problem. In the above Rosenbrock function, the objective function is Equation 3.2 [37]. The objective function can be linear or non-linear, continuous or discrete, equality or non-equality constrained, single variable or multi-variable, with single minima or multiple local minima, bounded or unbounded, etc. [36].

For this thesis, the objective function is an equation that calculates the error between actual values and predicted values. The goal of model training is to minimize the objective function. It is also possible to maximize the objective function in certain scenarios, in which case it should be defined accordingly. Here, it is a non-linear, discrete, multi-variable function where the coefficients are real numbers rather than integers. It has some equality constraints with multiple local minima. The goal of the function is to minimize the average mean absolute error between the actual and predicted values. More information can be found in chapter 4.

There are several methods available for minimizing the objective function, these are called optimization algorithms. Based on the problem and the objective function, an optimization algorithm can be chosen for the model. These are discussed in the section 3.2.

3.2. Optimization Algorithms

During the development of the model for this project, several optimization algorithms were considered and tried. While there is literature available for qualitative comparison between some of these algorithms, it would have been difficult to judge them without trying them out. This section represents the approach undertaken during this project, thus, leading to a better understanding of the problem at hand.

Before considering any machine learning based algorithms, simple gradient-based methods were tried. Since the model in this thesis is a multi-variate problem i.e., it has multiple variables, optimization algorithms from the Scipy package of Python were tried [37]. Once the results showed that the objective function had many local minima, global optimization techniques from the same package were tried like brute and differential evolution algorithms [37]. Upon success of these algorithms, it was decided to look at some ML-based techniques. The qualitative description of some of these optimization algorithms can be found in the following subsections. Since it was known that the objective function is non-linear, algorithms based on linear regression were not considered.

3.2.1. Local Optimization

One of the most simple, common and often modified algorithms used for optimization is Gradient Descent method [36]. It uses the gradient that is a vector function of the partial first-order derivatives of the cost function with respect to each variable for finding the local minimum. The value of the gradient defines the direction and rate of decrease of the function [36]. Thus, repeated steps are taken in the opposite direction of the gradient towards the steepest descent or ascent (in the case of local maximum) [36]. This process is similar to a drop of water which is sliding down the side of a curved bowl.

An arbitrary initial point (set of values for variables) is chosen and it moves iteratively in the direction of the fastest gradient or decrease of the objective function [36]. Depending on the step size, the algorithm can converge on the local minimum too slow or fast. If it is too large, the algorithm might even diverge from the optimum. Since gradient descent is a local optimization algorithm, it will not reach the global minimum unless appropriate initial point is chosen [36]. Thus this method was found unfit for the model.

Other local optimization methods more complex than Gradient-Descent were also tested. For instance, Nelder-Mead and Powell methods can be used for non-linearity and for multi-dimensional space [37]. However, the objective function for this thesis is discontinuous and hence it is not possible to obtain derivatives, which are needed for both these methods. Other methods tried were Constrained Optimization BY Linear Approximation (COBYLA), Sequential Least Squares Programming (SLSQP), and trust-constr [37]. However, these methods do not accept either bounds and/or constraints, which is inconvenient as it leaves the problem definition incomplete. Detailed information on the working principles of all these optimization methods can be found in the Scipy package documentation [37].

Despite the increased complexity, all these optimizations were stuck in local minimum or gave the initial guess as the final optimized result, even after hyper-parameter tuning. Thus, it was clear that global optimization techniques were necessary for the model.

3.2.2. Global Optimization

Due to the problems seen in the above mentioned optimization techniques, global optimization methods like Brute optimization and Differential Evolution were tried. Both these methods can be used for multi-variate problems defined with certain bounds and constraints. They are also compatible with discontinuous functions. Both these methods are a part of the Scipy package [37].

Brute Optimization

As the name suggests, this method uses brute force to calculate the value of the objective function at every point in a multidimensional grid and returns the global minimum [37]. Thus, a range for each variable needs to be given as input along with a step size to compute the multidimensional grid where each grid-point represents a possible set of values for optimization variables. An appropriate grid spacing needs to be chosen in order to capture the global minimum. Since the function is evaluated everywhere within the given range and the time taken for the optimization strongly depends on the number of grid-points, this approach is inefficient and slow. Even if a coarser grid spacing is chosen, the algorithm can take a long period of time or can run into limited memory problems [37].

In order to overcome this memory or computational time problem, another minimization technique is used along with brute optimization. Accordingly, first brute optimization is used on a coarse grid to find the grid-point closest to the global minimum and then a local optimization method is used to find the global minimum using this grid-point as initial value. The assumption here is that if the grid is just fine enough, the resulting grid-point will be most likely in the neighborhood of the global minimum. Thus, it is possible to use another optimization method to "polish" the grid point into finer spacing, i.e., to find the actual global minimum near the best grid-point using brute optimization [37].

Brute optimization is a guaranteed method to reach close to the global minimum. Using this method verified that more than one local minimum existed in the problem statement and that any local minimization technique would thus fail. However, as mentioned above, it is very time consuming and/or could be an imprecise technique. Hence, brute force was not used for the proposed model.

Differential Evolution
Differential evolution is a stochastic method, good for global optimization, where one or more parameters has randomness [37]. Thus, it can, partially in a random manner, search a large multidimensional space but it might take longer to reach the optimized values compared to gradient-based techniques.

The optimization starts with a random stochastic population, which undergoes mutation by mixing of a member of population with another member to create a new member. This happens over the entire population at every iteration of the optimization. There are different mutation strategies available to chose from with 'best1bin' strategy being the default [37]. In this, two random members are chosen and their difference is used for mutation. Once the new member is accepted, its fitness value, i.e., the value of the objective function for that member is calculated. If this new member is better than the previous or the overall best member, the new member will replace the old member of the population. In this way, eventually the members will converge towards the global minimum after each iteration.

As suggested for differential evolution, hyper-parameter tuning like using larger population size, higher mutation, and lower recombination was done to widen the search radius and reach the global minimum. This method, while slower than gradient method, is faster than brute optimization. It is also independent of grid size. Thus, this technique was a strong contender for the proposed model.

Machine Learning
Global optimization problems can also be solved by making use of machine learning algorithms. In the review papers mentioned in the literature study [8]–[10], ML models were successfully used for forecasting energy yields of PV systems using system information and historical data from the same system as input. ML can identify trends and patterns and deal with large amounts of data. The previous research conducted by R.C.Nijman in PVMD group also recommended the use of ML based algorithms for peer-to-peer monitoring [26].

Machine learning includes several algorithms to learn and to generalize historical data to make predictions on new data. These algorithms approximate a function that maps the input data to the output data, this is in the form of a function optimization. Thus, machine learning defines a parameterized mapping function between input and output, and the optimization function is used for finding the values of these parameters to minimize the error of the function used to map [38]. In a way, every time a machine learning algorithm is fitted on a training dataset, it is solving an optimization problem. Key advantages of using machine learning are no human intervention needed, the ability to handle large and multi-dimensional datasets, easily identify trends and patterns, continuous improvement as more data gets available, and the application to wide variety of fields [38].

ML can be combined with differential evolution method in order to take advantage of both approaches. The global optimization would search in the multidimensional space in an automatic and optimum way. The algorithms that combine both approaches are called evolutionary algorithms and are the choice of this thesis.

3.2.3. Evolutionary Algorithms

Due to their nature, evolutionary algorithms can cover a large candidate space while overcoming the problem of getting stuck in a local minimum. Evolutionary Algorithms (EA) are different from traditional algorithms as they are not static, i.e., they are dynamic and evolve over time. EAs have the following characteristics, namely, they are population-based, fitness oriented and variation driven [39].

Population-based: EAs optimize a process where the current solutions, called the population, generate new solutions. So the old population leads to a new population.
Fitness-oriented: To compare the different solutions and the new population with the old population, fitness values are calculated for each individual solution based on the fitness function (usually the objective function). So a fitness value represents how good of a solution it is.
Variation-driven: EAs generate a new population which is supposed to be better than the old population according to the fitness values of that population. In this process, individual solutions from the old population undergo some variations to generate new solutions.

Amongst the many evolutionary algorithms, Genetic Algorithm (GA) has been popular in industry and the academia mainly due to its intuitiveness, easy implementation, and the ability to efficiently solve non-linear problems with mixed integer optimization [40]. Another algorithm, Particle Swarm Optimization (PSO) is a relatively new heuristic search method inspired by the collaborative behavior of biological populations. Both of these are relatively simple and easy to modify algorithms [40]. PSO and GA are similar in terms of mutation and recombination, however, both of them first convert the numbers into bits; whereas for differential evolution directly numbers are used. Not using direct numbers for mutation and recombination makes these steps rather complicated and counter-intuitive to use or modify.

Moreover, amongst PSO and GA, one performs better in some cases while the other performs better in others. The two algorithms traverse the candidate space rather differently [40]. This was further confirmed by Abraham et al. whose experiment results indicate that usually GA obtains better results than PSO [41]. Their research was not on PV systems but was conducted by using GA for neighbour-selection strategy in peer-to-peer networks. Some more research that uses GA for peer-to-peer type problems in other fields was referred. Therefore, thanks to the previous research along with the fact that GA is simple and easy to modify, GA was chosen over PSO for optimization.

In this project, Genetic Algorithm was chosen due to its advantages over differential evolution along with the fact that GA is widely used in the industry and it is possible to modify each aspect of this algorithm to include benefits from other algorithms. The Genetic Algorithm will be discussed in more detail in the next section.

3.3. Genetic Algorithm

Genetic Algorithm is one of the simplest classical Evolutionary Algorithms. At its core, GA is based on Charles Darwin's theory of natural selection: "survival of the fittest" [39]. It is random-based, meaning some random changes are applied to the current set of solutions to generate the new set. These changes are usually slow and small until the best solution is obtained. As the name suggests, it follows the process of natural selection which starts with the selection of the best/fittest individuals from a population. These individuals mate and produce offspring which inherit random characteristics of their parents. These offspring form the next generation or new population. According to Darwin's theory, fit parents will lead to better and fitter offspring, and thus, will have a better chance at surviving. Hence, over multiple generations, the optimum fitness individual can be found.

The same logic of natural evolution is applied for an optimization search problem. The process starts with a set of random solutions for the problem and selects the best ones amongst them and so on. Thus, GA can be divided into five phases:

- Initial population
- Fitness evaluation
- Selection
- Crossover or reproduction
- Mutation

Each of the above phases mirrors the process of evolution of species in nature. As seen in Figure 3.1, the phases repeat themselves over generations or iterations, where each generation uses the data from the previous one to create the next. The phases are explained in detail below by comparing the natural evolution with mathematical interpretation of genetic algorithm.

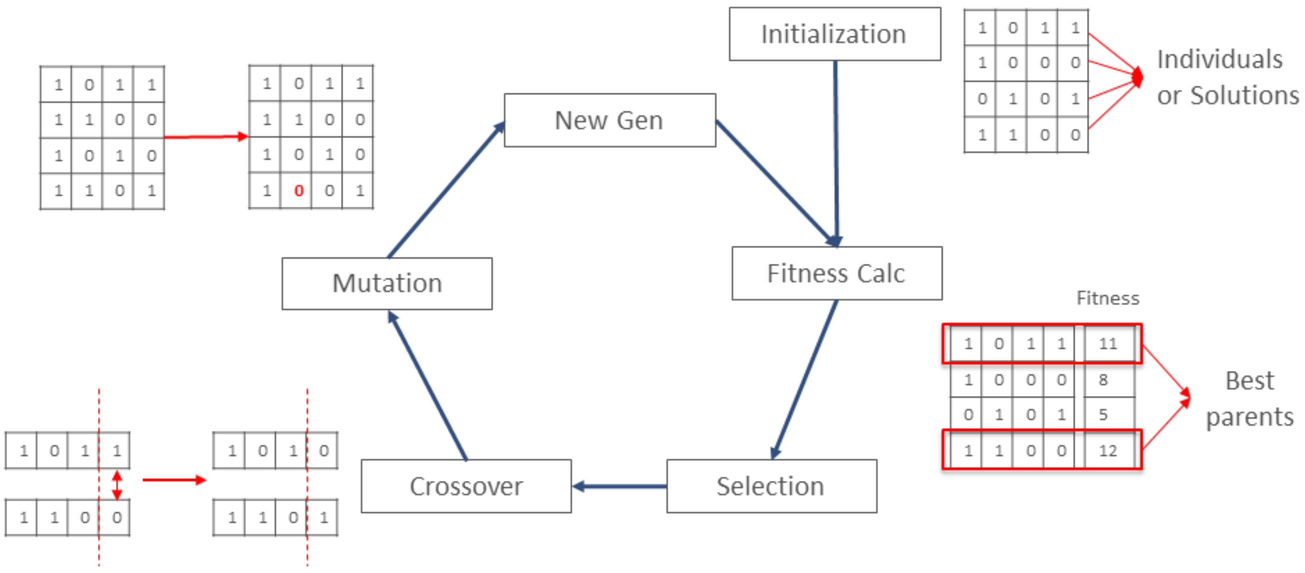


Figure 3.1: Flow of Genetic Algorithm

3.3.1. Initial Population

To start the process of selection and evolution, an initial population is needed. Thus, the initialization includes a set of individuals who are a potential solution to the optimization or design problem. Each of the individuals has a unique set of features also called 'genes' (refer Figure 3.2) which will evolve at every iteration [42]. These features are randomly selected for each individual during initialization. In biological terms, the value of the features depends on their genes and they are randomly chosen such that some features make the individual better than the others while some make them weaker. This creates a randomly distributed initial population. The idea is to have the population evolve into individuals with more good features than bad.

In order to translate the initialization to the optimization problem at hand, refer Figure 3.2. As seen in the figure, the population is a set of all possible solutions to the optimization problem, analogous to the random population of individuals. Each solution is defined by a set of variables (or parameters) called genes. A gene is one element position in a chromosome such that a set of genes joined together form a chromosome [43]. A chromosome is a potential solution to the given problem. In mathematical terms (refer Figure 3.2), a chromosome is represented using bit-string i.e., a string of binary values of 0s and 1s, is used. Thus, like in nature, genes (or features) are encoded in a chromosome (or individual or solution).

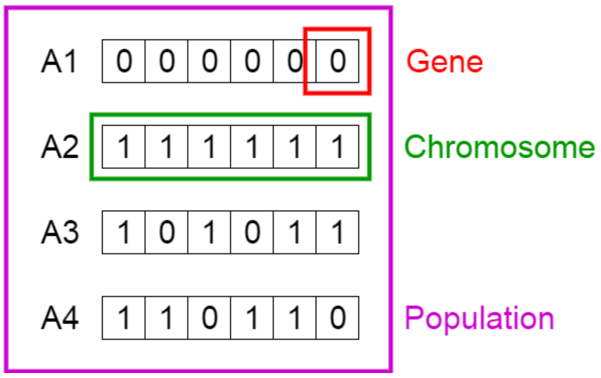


Figure 3.2: GA: Initial population of chromosomes and genes [43].

At this phase, it is necessary to consider the distribution of the initial population, i.e., to consider how the individuals might vary from each other. If there are little or no variations within the population, it is unlikely that the population will evolve into a better ’new’ population. Thus, it is a good idea to have the population size ’large enough’. A general rule of thumb is to set the size to at least three times the number of inputs and to confirm that the results converge [44]. This highly depends on the design problem, the number of features and their range. This is part of the hyper-parameter tuning which will be discussed later.

3.3.2. Fitness Evaluation

A fitness function is used to evaluate each individual or solution in the population. The fitness function (or objective function) is simply a function that determines how fit an individual is in comparison with other individuals in the population. The function takes a solution as input and gives the fitness score for each solution as an output. Better the fitness score, higher the probability that the solution will be selected for the next phase crossover. The aim of GA is to improve the fitness score of the model over each iteration. Thus, defining a precise and efficient fitness function is vital for a proper optimization. While usually they are the same, it is also possible to have different fitness and objective functions based on the problem.

Usually the idea of fitness function is to maximize or minimize a certain metric or parameter. One advantage of GA is that a complicated fitness function will not have an impact on the algorithm’s execution [42]. A single model can have multiple fitness functions such as maximizing the area used by a design while also minimizing the weight. These are ’multi-objective’ optimizations where the fitness function is not same as objective function [39]. An example in nature would be to maximize the strength of the species population or the speed or ability to camouflage. In case of genetic model, it is simply the value of the objective function for each solution or chromosome.

Once the fitness function is defined, the selection phase of a genetic algorithm can begin.

3.3.3. Parent Selection

Under the guideline ’survival of the fittest’, for every generation, only a subset of the population is ’selected’ to reproduce so that only their genes can be passed on their offspring for the next generation. Hence it makes sense that individuals with higher fitness scores have a better chance to be selected for breeding as they have better genes. In this way, good genes are preserved and passed on to next generations. Proper selection is a crucial step in the convergence rate of the algorithm. On the other hand, it is also vital to maintain good diversity in the population. One extremely fit solution could lead to premature convergence and a loss of diversity [43]. This is not desirable in GA.

There are many methods used in the parent selection phase of GA. One of the most popular ways and used in this thesis project is Fitness Proportionate Selection (FPS) [44]. In this method, the probability that an individual can become a parent is proportional to its fitness. That means individuals that are fitter have a higher chance of mating and passing on their features to future generations and thus, evolving into better individuals. Other methods for parent selection were considered, however FPS was chosen since it is easy and straightforward.

3.3.4. Crossover

The crossover phase is analogous to reproduction or breeding in biological species. At basic level, two or more parents are selected and part of their features (or genes) are swapped (or crossed-over) with the others' to produce one or more offspring. These offspring solutions are similar but not identical to the parent solutions [42]. Thus, offspring have a combination of their parents' features, thus ultimately, some of the offspring will be fitter than their parents. Ideally, but not always, after every iteration, the average fitness score of the entire population will increase. Therefore, after enough number of iterations, the genes or features will converge to achieve the best fitness score or in this case, optimization. Crossover, thus, is the heart of genetic algorithm.

Similar to how species reproduce, the fittest individuals based on their fitness scores are chosen for breeding. The new individuals or offspring have the features of either of their parents [42]. Thus for the genetic model [43], a crossover point is chosen at random for each pair of parents (refer Figure 3.3). The offspring are created by exchanging the genes of each parent until the crossover point (refer Figure 3.4). These new offspring are then added in order to repopulate the set or population (refer Figure 3.5). This crossover operator is called as One Point Crossover.

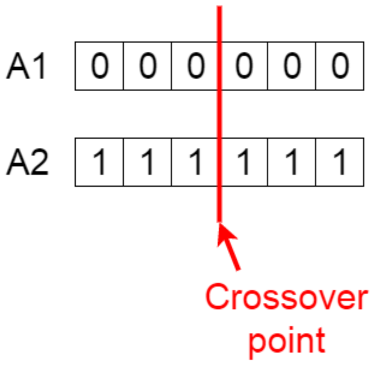


Figure 3.3: GA: Crossover Step 1 [43].

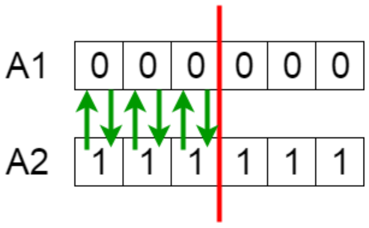


Figure 3.4: GA: Crossover Step 2 [43].

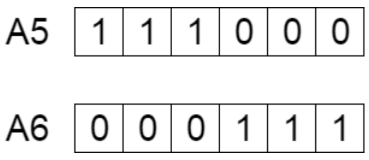


Figure 3.5: GA: Crossover Step 3 [43].

Apart from the One Point Crossover, there are many other crossover operators which can also be chosen. For example, Multi Point Crossover where more than one crossover point is chosen and the crossover alternates between parents. Other example is Uniform Crossover where each gene is treated separately and parent gene is basically chosen like a flip of coin [44]. There are other complicated crossover operators, used for integer genes instead of bitstrings. During this thesis, the One point crossover turned out to be accurate and computationally fast.

3.3.5. Mutation

Mutation is essentially used to introduce randomness to the algorithm. Here, at low probability, certain offspring undergo random mutation after each crossover. This implies that some of their features randomly change and are not similar or inherited from either of their parents. For example, amongst the offspring, one gene might mutate into a new one, which neither of its parents have. Similarly, in the genetic model, some of the bits in the chromosome (bitstring) of offspring can simply be flipped, i.e., 1 becomes 0 or vice versa as seen in Figure 3.6. Please note that the probability of mutation for each gene is not as high as the figure represents.

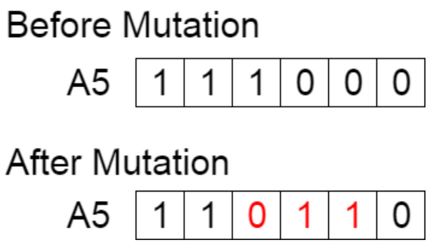


Figure 3.6: GA: Mutation [43].

The motivation behind mutation is to maintain diversity within the population set as this deters premature convergence of the optimization [43]. Sometimes, very rarely, mutation could also lead to an offspring with much better features than the parents. This logic is similar to biological evolution where a mutation led to sea creatures being able to walk on land [42]. In mathematical terms, mutation ensures that the genetic algorithm does not get stuck at a local minimum; a problem observed with other optimization techniques [44]. By assigning an appropriate probability for mutation, the algorithm 'explores' the rest of the candidate space despite having identified a set of good genes early on. Thus, it can break out of the crossover cycle and converge onto the global minimum. If the mutation probability is too high, the algorithm essentially becomes similar to random search. It is a common observation that mutation has a higher impact on convergence of GA than crossover.

As seen in Figure 3.1, the offspring after crossover and mutation become the new population for the next generation and the process repeats itself. The iterations will continue until a termination condition which could either be a limit on number of iterations or a threshold objective function value is achieved or the objective function value has not changed for a certain number of iterations [44]. Genetic Algorithm is a powerful algorithm often used for optimization, however there is no guarantee of the optimality of the solution. If implemented improperly, the algorithm may never converge to an optimal solution, thus, it is important to tune the hyper-parameters of the algorithm.

3.4. Hyper-parameter Tuning

Hyper-parameter tuning is an iterative process to find the optimum arrangement of settings for the optimization algorithm. It highly differs from model to model based on the algorithm. With an increase in model complexity, the number of hyper-parameters also increases [45]. Hyper-parameter tuning can be performed either manually or automatically.

One manual method used for tuning would be a grid search, where all possible configurations of the available hyper-parameters are tried [12]. By calculating the metric value for the objective function for each possible configuration, a grid of metric values can be formed. From this grid, the point with optimal configuration is chosen for the hyper-parameters. While this method guarantees an optimal solution, it can be time consuming when dealing with complex algorithms with more hyper-parameters. A less complex algorithm could also become time consuming when dealing with large datasets.

An alternative approach would be randomized search, which tests for only arbitrary hyper-parameter configurations within a specified range. A solution close to optimal can be found relatively faster by this method. For this project, general recommendations for GA tuning were followed, such as lower probability for mutation, population size at least three times the number of variables, etc. [45]. Upon some manual iterative process, the optimal configuration was found.

3.5. Metrics of Evaluation

The model primarily calculates the numerical coefficients of an equation that minimizes the distance between the fitted line and all of the data points by minimizing the mean absolute error, i.e., the average difference between the actual value and the predicted value. In this section, the metrics used to examine the 'goodness of fit' of a model are explained in detail. The metrics Mean Absolute Error (MAE), Normalized Mean Absolute Error (NMAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), Normalized Root Mean Squared Error (NRMSE) and R-squared error (R^2 score) were considered. Finally, Normalized Root Mean Squared Error and R-squared error were used. These metrics were calculated per system and then the mean of all systems in the testing dataset was used to compare the performance of different models.

It is necessary to establish a concrete method to compare the performance of the model. Hence, error metrics are used to:

- Evaluate the performance of a new model developed
- Compare performance of a new model with other models
- Analyse the various cases of use for the proposed model

3.5.1. Normalized Root Mean Squared Error (NRMSE)

Root mean squared error (RMSE), similar to standard deviation (MSE to variance), tells you how concentrated the data is around the line of perfect fit i.e., how spread out the residuals are [46]. It is an extension of Mean Squared Error (MSE) where RMSE is square-root of MSE. Because the MSE is squared (refer Equation 3.3 [46]), its units do not match the units of original output. Thus, often RMSE is used to convert the error metric back to original units, making interpretation easier. Both MSE and RMSE can range from 0 to positive infinity, thus, it becomes harder to interpret how well your model is performing. Hence, for this project, Normalized RMSE is considered, where normalization is done with the difference between maximum and minimum of actual value (refer Equation 3.4). Thus, the value of NRMSE varies between 0 and 1 such that lower the NRMSE the better. NRMSE enables the comparison between the performance of predictions using different models as well as the comparison between the performances of different systems, especially when they have different scales.

As seen in Equation 3.3, the residual is squared which means NRMSE is largely affected by the presence of outliers in the data. The outliers in the data will contribute to a quadratically higher total error. This is a major drawback for using NRMSE, hence proper pre-processing of the data is vital.

$$RMSE = \sqrt{\sum_{i}^{n} \frac{(y_i - \hat{y}_i)^2}{n}} \tag{3.3}$$

$$Normalized RMSE == \frac{RMSE}{y_{max} - y_{min}} \tag{3.4}$$

where, $\hat{y}_i$ is the predicted value of y for observation i,
y_i is the actual value of y for observation i.
n is the number of observations

y_{max} is the maximum of actual values of y,
y_{min} is the minimum of actual values of y.

3.5.2. R-squared Error

R-squared error (or R^2 score), also known as the coefficient of determination, explains the correlation between the actual and predicted data. It is a statistical measure of how close the data points are to the fitted regression line [47]. It explains how much better your regression line is than a simple horizontal line through the mean of the data. It is usually between 0 and 1 (or 0 and 100%) where "0" (no correlation) indicates that the model explains none of the variability of the data around its mean and "1" (perfect correlation) indicates that the model explains all variability of the data around its mean (refer Figure 3.7 and Equation 3.5). However, R^2 score can have negative values when the predicted data is inversely proportional to the actual data.

$$R^2 = \frac{Explained\, variation}{Total\, variation} \tag{3.5}$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{3.6}$$

where, $\hat{y}_i$ is the predicted value of y for observation i,
$\bar{y}$ is the mean of all y value,
y_i is the actual value of y for observation i.

In general, the higher the R-squared error, the better the model fits the data. However, it is possible to have a low R-squared value for a good model, or a high R-squared value for a model that does not fit the data [47]. Hence, it is necessary to look at the residual plots or use another metric along with R-squared error to determine the goodness-of-fit of the model.

For this project, an R^2 score higher than 0.85 (or 85%) is considered a good fit. Accordingly, the mean R^2 score of all PV systems in the testing dataset as well as total number of PV systems with a R^2 score higher than 0.85 were used as metrics for comparison.

As a final remark, it was found that neither of the metrics was sufficient to be used independently. Hence, it was necessary to look at all of them together when comparing the performance of model. For comparison of different PV systems in this report, Normalized RMSE and R^2 score were used whereas when comparing amongst models (that is on all systems in testing dataset together), mean Normalized RMSE, mean R^2 score and number of systems with good R^2 score (that is > 0.85) were used.

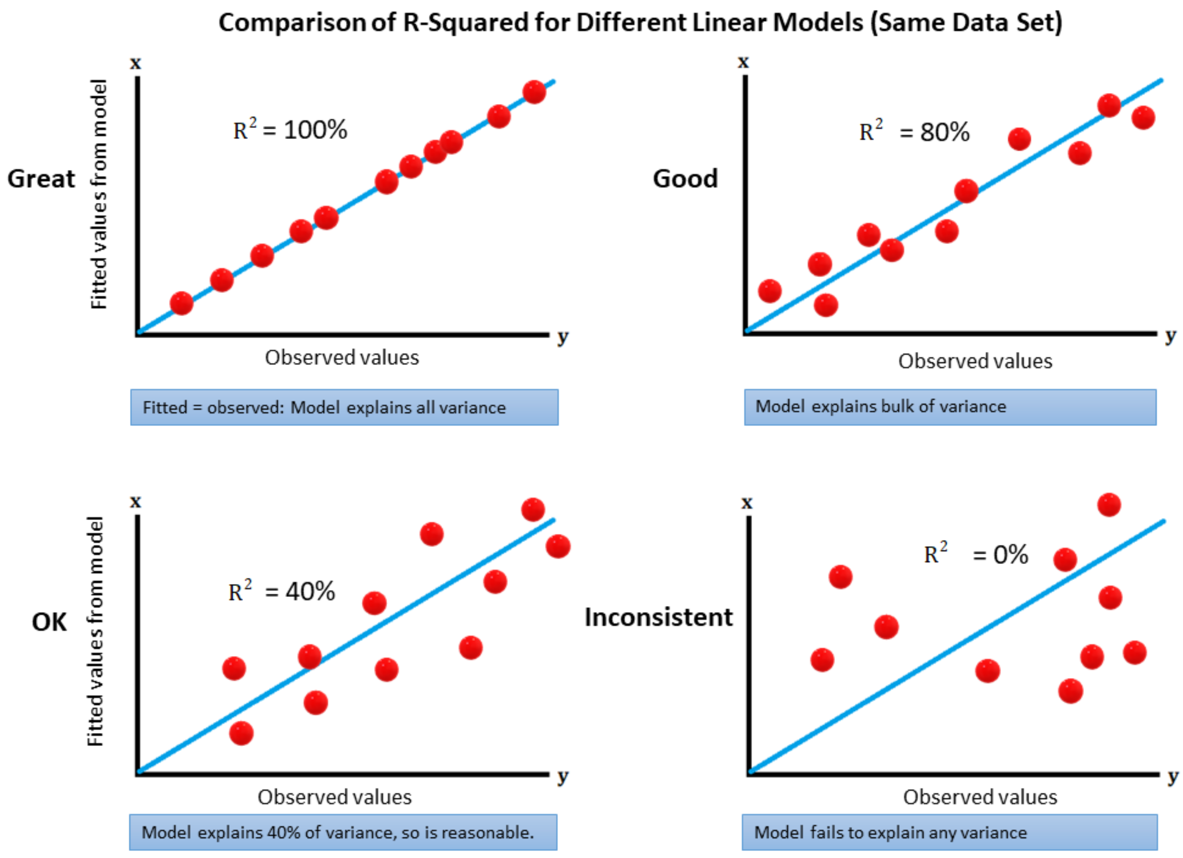


Figure 3.7: Evaluating the goodness-of-fit using R-squared error [48].

4 Model

In this chapter, the different models considered that can be used to predict yields are described. First in section 4.1, the Solar Monkey Model is explained, which is based on the conventional approach. Next in section 4.2, the 'Literature' model is explained. This is also the baseline model used during this project. Lastly, the proposed model based on the Genetic Algorithm is explained in section 4.3.

4.1. Solar Monkey Model

The existing model used by Solar Monkey is an analytical model which outputs the expected yield as explained in section 1.3. The yield is calculated based on the extrapolated weather data for the location where a system is, for every monitored system, every day. This model makes use of textbook solar modelling practices from the scientific literature explained by Smets et al. [5].

The model is represented by the flowchart seen in Figure 4.1. To begin with, Photovoltaic Geographical Information System (PVGIS) and panel orientation is used to calculate the diffuse and direct irradiance on each panel. Along with this, elevation or height data is used to calculate the shading losses for that system. Taking into account the two types of irradiance and their respective shading, the total Plane Of Array (POA) irradiance is calculated for each panel. Next, using the total watt-peak, the inverter efficiency and other losses, the "typical yields" for the whole system are calculated based on this irradiance. The "typical yields" are the yields in a normal year under typical weather conditions for that PV system. The typical yields are used later in the proposed model as will be discussed in section 4.3.

In parallel with the typical yields calculation, Royal Netherlands Meteorological Institute or KNMI daily irradiance is used to calculate a weather factor. The typical yields and weather factor together with system decay are used to calculate "expected yields". These expected yields are used to compare the performance of the proposed model with the existing Solar Monkey model, the results of which can be found in chapter 5.

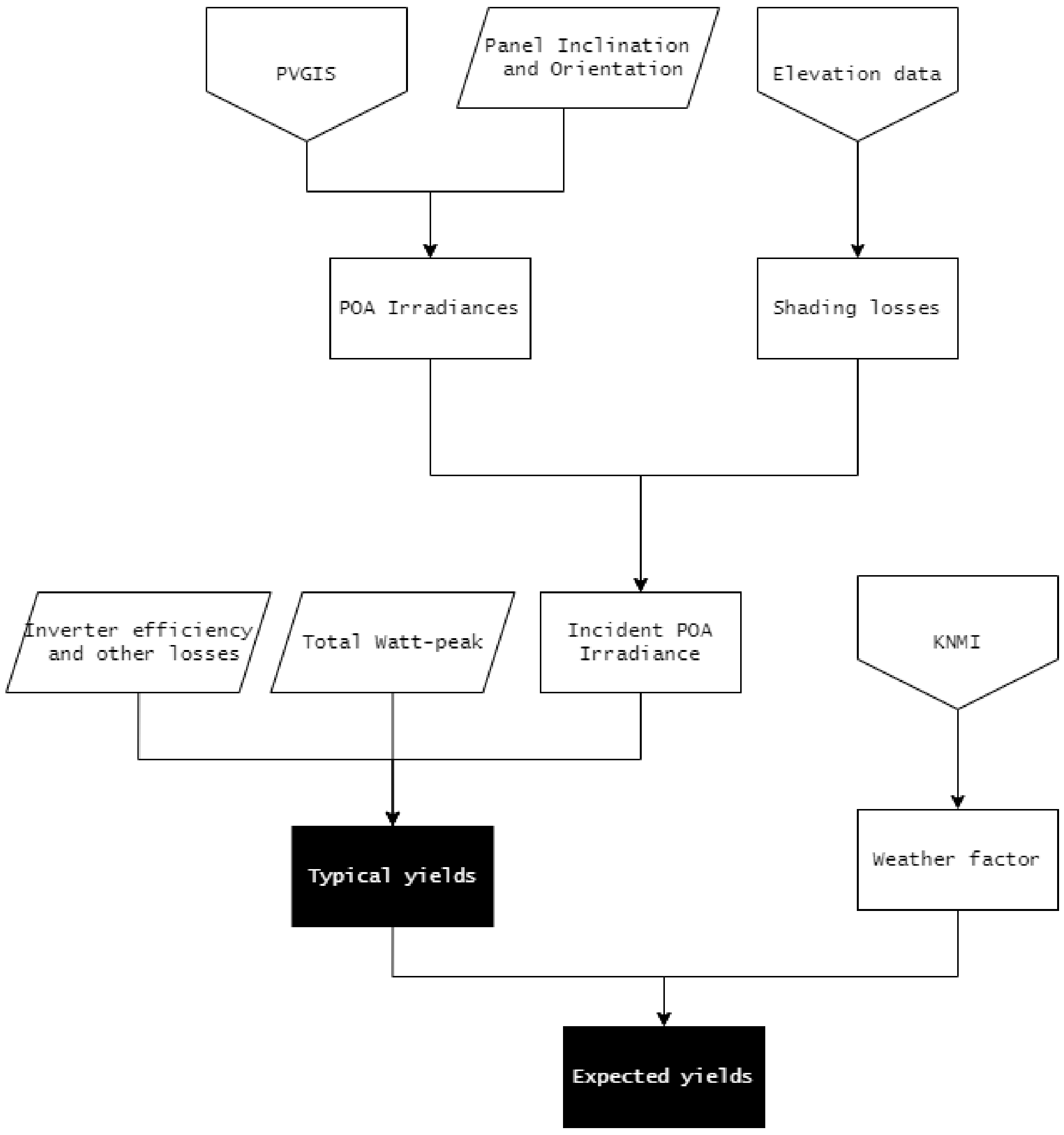


Figure 4.1: Flowchart describing the old Solar Monkey model [49].

One of the key flaws in the above model (refer Figure 4.1), was the weather factor which scales the power produced linearly with irradiance. This is inaccurate as yields not only depend on global irradiance but also on the amount of diffuse or direct irradiance and shading on each panel and other factors. Thus, a new model was developed by the R&D team that uses KNMI irradiance instead of typical PVGIS yields alone. Here, the Erbs model is used to decompose the global irradiance from KNMI into diffuse and direct components to calculate more accurate total POA irradiance [49]. Other changes in the improved algorithm include the Sandia Array Performance Model (SAPM) [50] to compute the module temperature and Huld model [51] to calculate panel power output based on temperature and irradiance.

According to the comparative results of the above two models, the new model is more precise in prediction than the older one. However, due to its complexity, it takes more computation time for the calculation of expected yields. In this thesis, the results of both models were compared with the proposed model (refer chapter 5).

4.2. Literature Model

One of the first steps taken for this project was to find any prior research done on the topic of peer-to-peer monitoring. While this is a relatively new approach for residential PV system monitoring, one research paper was found by Leloux et al. [27], [28]. As mentioned in section 1.3, this paper introduced a new metric Performance to Peers (P2P). In P2P, i.e., literature model, the performance of one system is compared with its peer systems in order to monitor the former system. This was the first model replicated during this project and it acts as the baseline. Several improvements were made to this baseline model to develop the proposed algorithm.

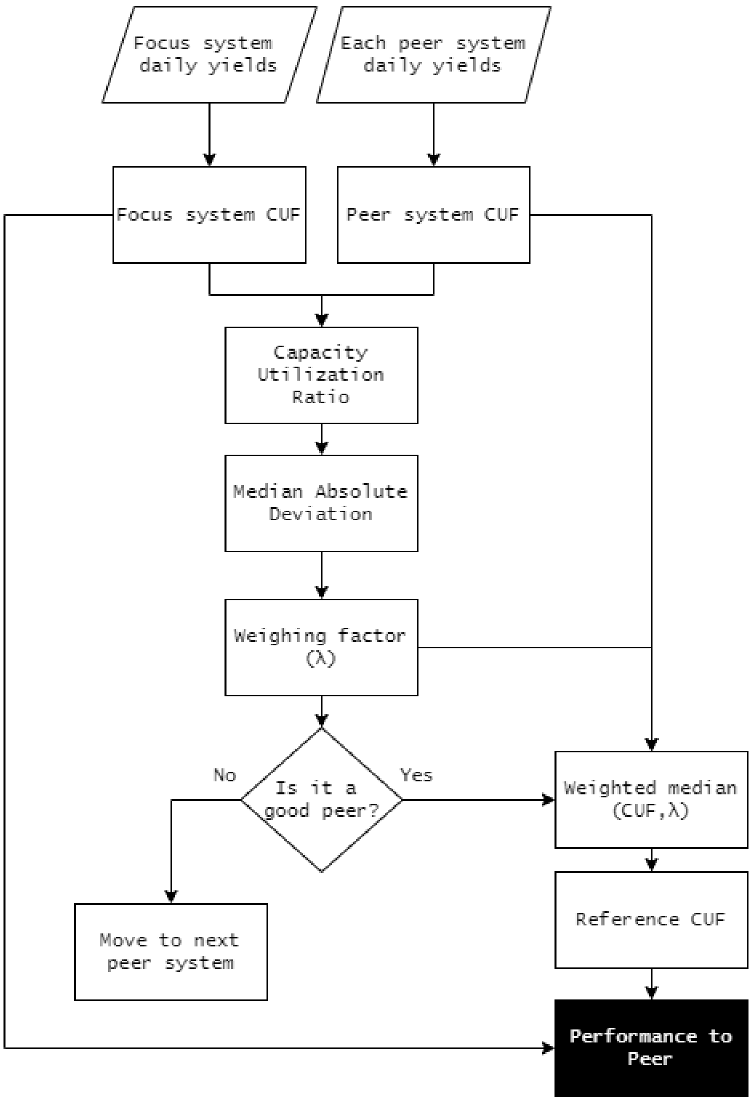


Figure 4.2: Flowchart describing the literature model [28].

According to the literature model [28], the **focus system** is defined as the PV system to be monitored or analyzed, whereas **peer systems** are all the other available PV systems. The data of a good peer system is used to monitor the focus system. A general flow of the literature model can be found in Figure 4.2. The first step is to find the good peers for the focus system. The model uses only the energy yield data of PV systems. The paper suggests using hourly data, however daily yield data is used here, since only daily yield are available for this project. The daily yields are normalized with respect to their total capacity to give the Capacity Utilization Factor (CUF) as shown in Equation 4.1. With the daily yields now normalized to same scale, the Capacity Utilization Ratio (CUR) is calculated for the focus-peer system pair, according to Equation 4.2. Next, the weighing factor is calculated by taking the inverse of the Median Absolute Deviation (MAD) of CUR for the pair (refer Equation 4.3). This weighing factor is used to determine whether the peer system is a good peer system or not. Higher the weighing factor, the better the peer it is for the chosen focus system.

$$CUF = \frac{E_{PV}}{P * T} \tag{4.1}$$

where, E_{PV} is the energy output of the PV system,
P is the peak power of the PV system,
T is the time interval of energy output measurement.

$$CUR = \frac{CUF_{focus}}{CUF_{peer}} \tag{4.2}$$

where, CUF_{focus} is the CUF for the focus PV system,
CUF_{peer} is the CUF for the peer PV system.

$$weighing\ factor = \frac{1}{[MAD(CUR)]^4} \tag{4.3}$$

The second step is to use the data of the good peer systems in order to calculate the expected yields or the Performance-to-Peer (P2P) metric [28]. Once good peers are chosen, a weighted median of the peer systems CUF with respect to their respective weighing factors is calculated to give the Reference CUF. Reference CUF can be recognized as the normalized expected yields for the focus system. Hence, the true expected yields can be estimated by reverse calculation of CUF. Additionally, the model further calculates P2P for each day by taking a ratio of the focus system CUF to Reference CUF. This new metric is proposed instead of the conventional Performance Ratio (PR) for the monitoring of focus system.

4.3. Proposed Model

As mentioned earlier, the proposed model is based on the literature Performance-to-Peer model. However, since more data about the PV system is available at Solar Monkey, the proposed model does not only depend on the system daily yields. It also uses the system design information along with the system's geographic location. Thus, system design information, daily yields and location are used together to find good peers for the chosen focus system. Nevertheless, only peer systems' daily yields are used to estimate the expected yields, similar to the literature model.

Distance Calculation

The model can be divided into three parts, each calculating a different kind of "distance". To begin with, the **"feature distance"** considers the different system design information available. This includes the features number of panels, panel inclination, and panel orientation. Based on the chosen features, the model calculates the Euclidean distance for each focus-peer system pair using Equation 4.4. Once the Euclidean distances are calculated for all pairs of focus-peer systems, they are normalized using a StandardScalar [52]. These normalized distances are then called the "feature distances". All "distances" were normalized as they each have different scales.

$$d(f,p) = \sqrt{(f_1 - p_1)^2 + (f_2 - p_2)^2 + ... + (f_n - p_n)^2} \tag{4.4}$$

where, f is the focus PV system,
p is the peer PV system,
n is the number of dimensions or features.

Next, the **"yield distance"** is calculated in a similar fashion as in the literature model. In the literature model, the daily yields of focus and peer systems are normalized with respect to their total daily capacity, to estimate the Capacity Utilization Factor (CUF). However, in this model, they are normalized with respect to the actual energy yield in the first year of installation of the system (for systems older than one year) or with respect to the Solar Monkey estimated energy yield for the first year of installation of the system (for systems less than a year old). The Solar Monkey estimated energy yield for the first year of installation of the system is the typical yields, as calculated in Figure 4.1. Further, for each focus-peer system pair, the Capacity Utilization Ratio (CUR) is calculated, which, as the name suggests, is the ratio of focus system CUF to peer system CUF for each day. Since a single value is needed for the distance, Median Absolute Deviation (MAD) is calculated from the CUR. Once the MADs are calculated for all pairs of focus-peer systems, they are normalized using a StandardScalar [52]. These normalized MADs are then called the "yield distances".

The third and final distance is the **"geographical distance"**. It is simply the physical distance between the latitude and longitude coordinates of each focus-peer system pair. Due to the curvature of Earth, Haversine distance is used here [53]. Again, once the Haversine distance is calculated for all pairs of focus-peer systems, they are normalized using a StandardScalar [52]. These normalized Haversine distances are then called the "geographical distances".

As mentioned, StandardScalar is used to normalize all "distances" because each of them has different scales. Standardization of a dataset is commonly required for many ML algorithms as they might behave in an absurd manner otherwise [52]. StandardScalar normalizes features by removing the mean and scaling all values to unit variance [52]. Thus, the standard score of a training sample x is calculated using Equation 4.5. Each feature is centered around the mean of training samples and scaled independently.

$$z = \frac{x - u}{s} \tag{4.5}$$

where, u is the mean of the training samples x,
s is the standard deviation of the training samples x.

Here, the **"distance"** indicates how similar a peer system is to the focus system. Thus, the peer systems from focus-peer system pairs with lower distances are chosen as good peers for the particular focus system. After all three distances are available for each focus-peer system pair, the "total distance" is estimated as the weighted sum of the distances according to Equation 4.6. A weighted sum is used because it is possible, for example, for "yield distance" to have more influence than "feature distance" in finding good peers. Since these weights (w) are unknown, they are evaluated using an optimization function as will be explained later in this section. Once the "total distance" is known for each focus-peer system pair, the pairs with the lowest distances, i.e., the peer systems most similar to focus system, are chosen as good peers.

$$total\ distance = (w_f * dist_f) + (w_g * dist_g) + (w_y * dist_y) \tag{4.6}$$

where, w_f is the weight for "feature distance" $dist_f$,
w_g is the weight for "geographical distance" $dist_g$,
w_y is the weight for "yield distance" $dist_y$.

Calculate Expected Yields

The daily yields of only the good peer systems chosen above are used to calculate the expected yields. However, as explained above, each type of distance has a different level of influence on the model, thus they are weighted differently. Similarly, each good peer system has a different level of influence on the expected yields. The peer system most similar to the focus system will have the highest influence. This peer system will be the one having the lowest "total distance". Thus, the weighing factor (λ) per peer system is evaluated as the reciprocal of the corresponding "total distance".

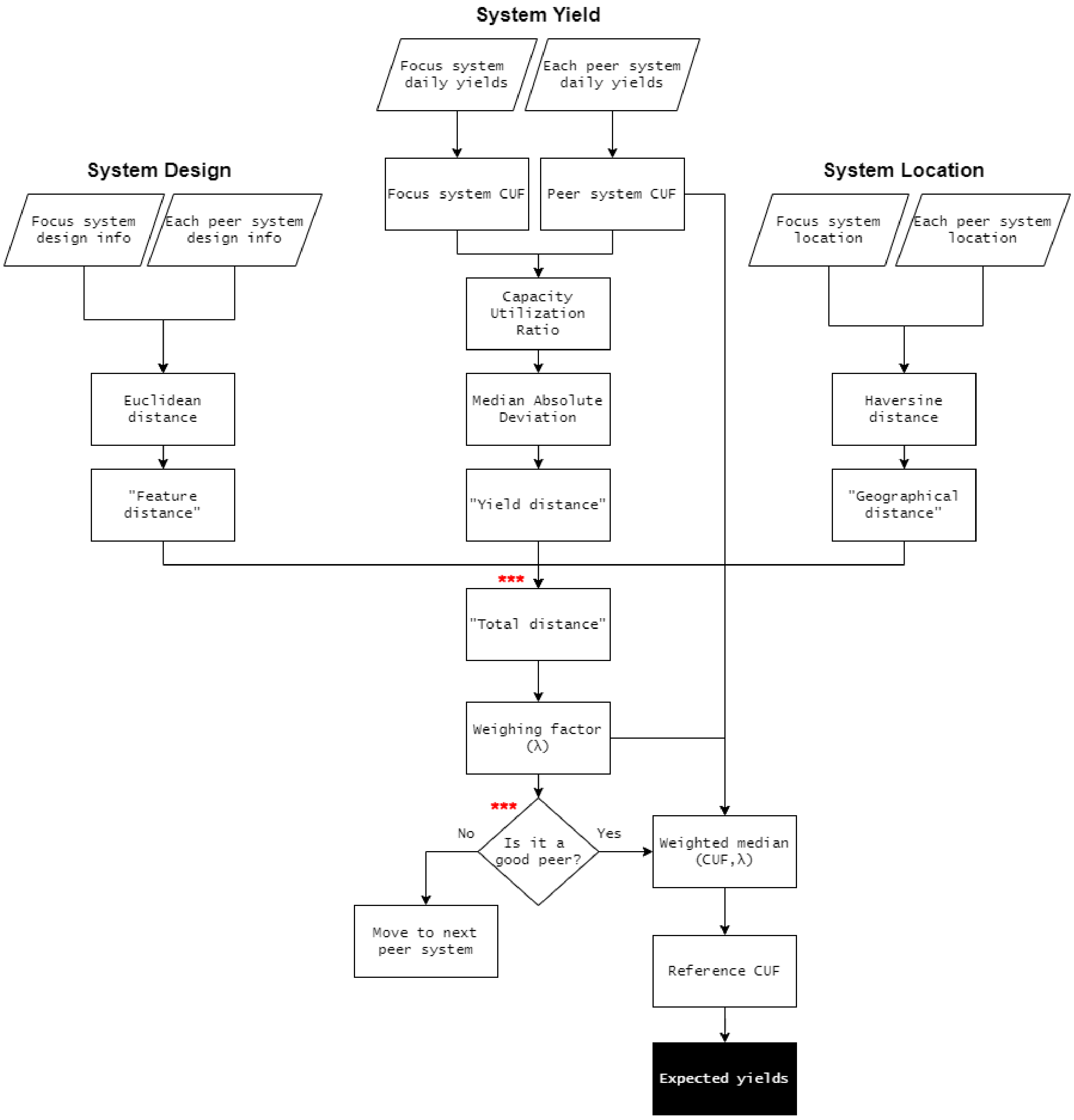


Figure 4.3: Flowchart describing the proposed model (red asterisks represent optimization variables).

Next, the CUFs of good peer systems along with their respective weighing factors (λ) are used together as a weighted median to determine the reference CUF. The reason for using a weighted median over weighted average is that a median is not influenced by an abnormal extreme value that may be present for one of the good peers [54]. This abnormal extreme value could be due to a fault in that particular peer system on that particular day and using a median essentially helps to avoid that fault from being transferred in the predictions of the focus system. Reference CUF can be interpreted as the normalized expected yields for the focus system as predicted by the good peer systems for that focus system. In order to estimate the expected yields, the reference CUF is simply multiplied by the actual energy yield in the first year of installation of the focus system (if system older than one year) or by the Solar Monkey estimated energy yield for the first year of installation of the focus system (if system less than a year old). In this way the expected yields for any focus system can be computed.

Optimization or Training of Model

The unknowns in the above model (refer to red asterisks in Figure 4.3), are the weights of each distance (w) and the number of good peers that need to be chosen for estimating the expected yields. These are the variables that need to be determined by training the model using an optimization algorithm. A simplified outline of the model training can be seen in Figure 4.4. As explained in section 3.2 and section 3.3, for numerous reasons Genetic Algorithm (GA) was used for the training of model. The weights corresponding to "feature distance", "yield distance" and "geographical distance" along with the number of peers are the optimization variables. The goal is to find values for the optimization variables such that the average Mean Absolute Error (MAE) between the actual yields and expected yields of focus system is minimized. This is the objective function used for the GA. The model is optimized on the training dataset. The results of model training, i.e., the optimized values for optimization variables are then used in the proposed model as shown in Figure 4.3, to evaluate the expected yields for the any focus system. The results of optimization and the expected yields from the model are presented in chapter 5.

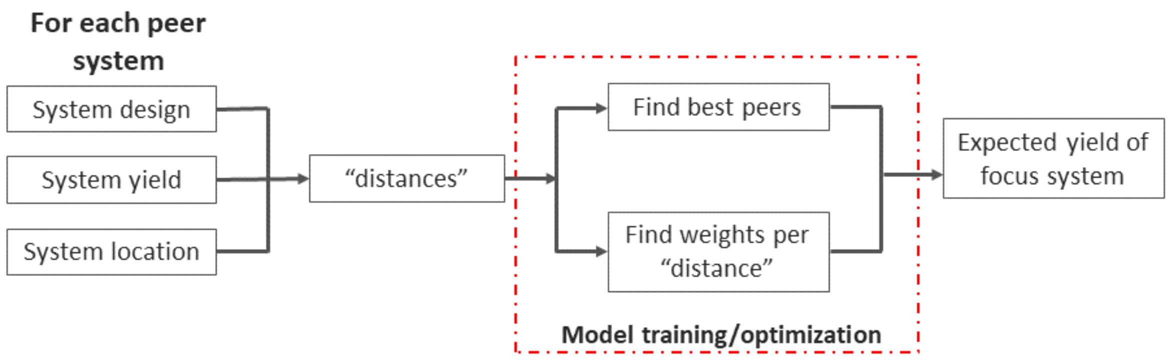


Figure 4.4: General outline for model training or optimization.

Distance Training

The main model training on training dataset of PV systems is as explained above. This training-testing data split is system-dependent. However, due to the nature of the proposed model, there is another part of the model that also trains but it is time dependent. In the calculation of "yield distance" (refer Figure 4.3), the daily energy yields for the focus and peer systems are used. These energy yields are only for a specific time period for all the PV systems in the training dataset. The time period can be changed as necessary. This specific time period is the dataset over which the "yield distance" is calculated also called as distance training. It affects the quality of peer systems found and in turn, also the quality of predictions. This explanation is important for understanding the results of fault detection in chapter 6.

5
Results

In this chapter, first, the results of the model training are presented in section 5.1, which uses the genetic algorithm for optimization. Followed by the model predictions using the above-mentioned optimization results in section 5.2. Predictions for a single PV system as well as results of testing the model on a group of systems are shown in this section. Next in section 5.3, the testing results for the proposed model are compared with the other models explained in chapter 4. Finally, some analysis was done on the use-case of the model, the results of which are described in the final section 5.4.

5.1. Optimization Results

A genetic algorithm was used to train the model and find out the weights for each of the three distances and number of peer systems needed for accurate estimation of the expected yield of the focus system as explained in section 4.3. The model was trained on randomly chosen 5000 systems with each having daily yields for up to one year (here the year 2020). A more distributed selection of old and new systems was ensured, thus, some of them are younger than one year. The results, as seen in Figure 5.1, show that the highest weight amongst the three distances is 0.875 for "yield distance". It is followed by a low weight of 0.125 for "feature distance". It is interesting to note that "geographical distance" has negligible weight compared to the other two distances. Finally, the number of peers needed for accurate calculation of expected yields is in the range of 12-20 peer systems.

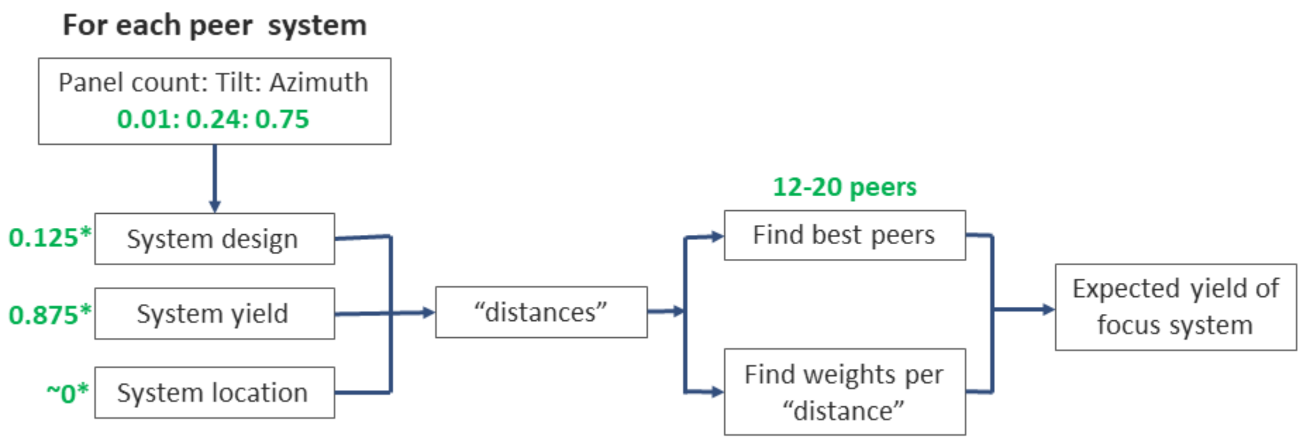


Figure 5.1: Results of model training or optimization model

Along with the three "distances" explained before, a more detailed optimization was done where the system design or "feature distance" is separated further into individual features. As seen in

Figure 5.1, system design was split into three features: panel count, panel tilt, and panel azimuth. Amongst the three, the panel count has a very low weight since it is highly correlated with total watt-peak, which is used in the system yield for normalization (refer section 2.2). The panel tilt and panel azimuth have a 1:3 weight distribution. Note that these individual feature weights are a part of the weight of "feature distance" i.e., 0.125.

Geographical Distance
To investigate the reason for negligible weight for "geographical distance", first, a correlation map was plot, since it is important to not have highly correlated features in ML models, as explained previously in section 2.2. As can be seen in Figure 5.2, "feature distance" is not correlated to either of the other two distances. However, "geographical distance" and "yield distance" have a Spearman correlation of 0.53. While this value does not represent a very high correlation, it is also not low enough to be ignored.

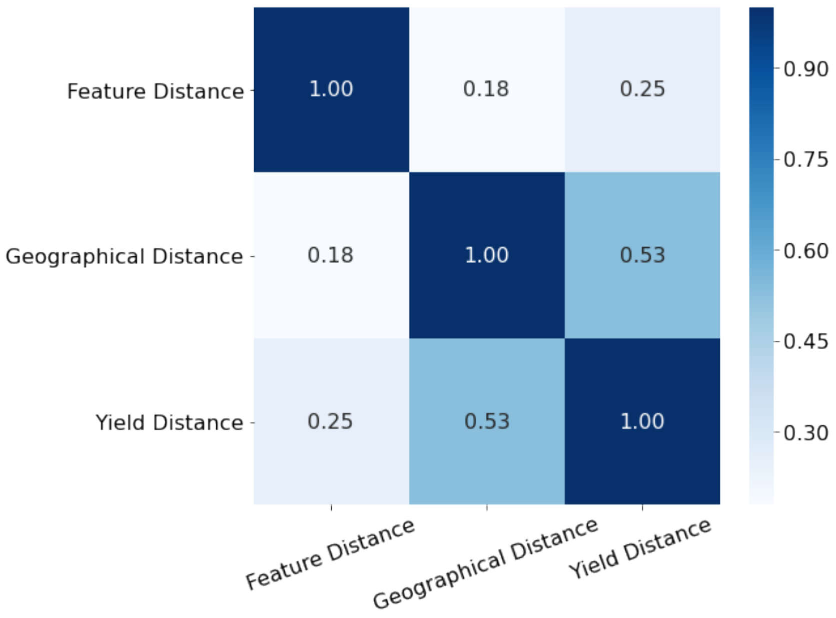


Figure 5.2: "Distances" correlation matrix

Not all ML methods are affected by correlation problems. To verify this for GA, an optimization model was run without the "yield distance". In this case, it was found that the "geographical distance" still has a low weight of 0.067, while now most of the weight is skewed towards "feature distance". Thus, GA does not have a correlation problem. Although it is important to note that having extra features increases the computational requirements for training, it is a good practice to avoid correlated features.

Another possible reasoning considered was due to the data itself. The PV systems available are only from the Netherlands, which is a rather flat and small country. One major influence of "geographical distance" for this project is that it enables the peer systems to account for the changes in very local weather like clouds near the focus system, especially when the weather stations are farther away from the focus system. This is important when dealing with hourly or higher frequency of energy yields. However, for this project, daily energy yields are used and it can be postulated that in the Netherlands, the day-to-day weather is very similar throughout the country. With this logic in mind, it is possible that when dealing with a larger area or with systems outside the Netherlands, "geographical distance" could become more important.

It is interesting to note that despite the "geographical distance" having negligible weight, most of the good peer systems found for a random focus system are geographically closer to it as seen in

Figure 5.3. However, in some cases the good peer systems could be more spread across the country.

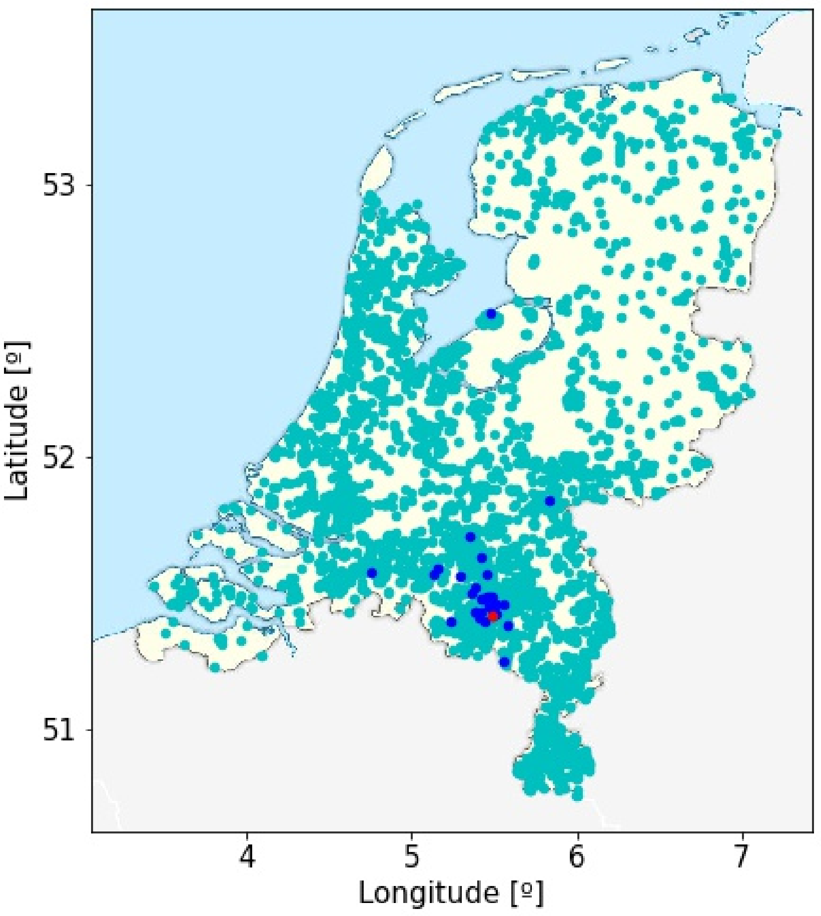


Figure 5.3: Spatial plot of good peer systems for a random focus system; focus system - red, good peer systems - blue, all available systems - green

Number of Peers
As seen in Figure 5.1, the optimization algorithm gives a value range for the number of peers. Upon multiple simulations, it was observed that the value of objective function, i.e., the fitness value of GA, does not change significantly within a given range of number of peers, provided the values for other optimization variables are kept constant. One of the reasons behind this result is that the model uses a weighted median when calculating expected yields. Thus, within a certain range, the value of weighted median does not change the result by a significant amount.

An experiment was carried out to find the ideal range of the number of peers, the results of which can be found in section 5.4.

5.2. Model Predictions

Using the results of model training with GA for optimization, it was possible to perform model testing on a new set of PV systems. The results of model testing, i.e., the expected yields can be seen in Figure 5.4. The figure shows the expected yields and actual yields of a random PV system for a period of 13 months. The expected yields closely follow the actual yields as anticipated. The metrics, as explained in section 3.5, for this exceptionally faultless PV system are: R^2 score of 0.978, Normalized Root Mean Squared Error of 0.0413.

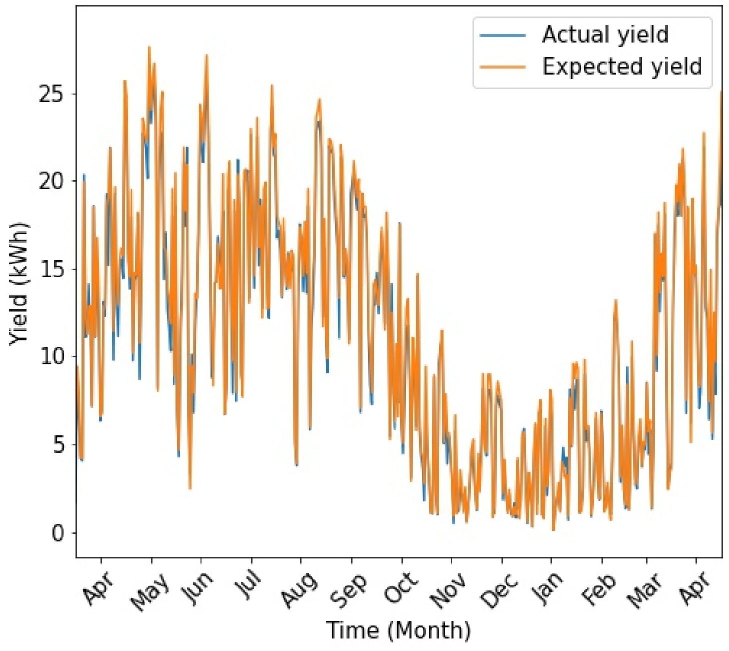


Figure 5.4: Temporal plot of actual yield and expected yield for a random focus system

While training and hyper-parameter tuning the model, it was tested on specific but randomly chosen 500 PV systems called the testing dataset. It was confirmed that this subset had a similar distribution as the total dataset available to ensure an accurate representation. The metrics, as explained in section 3.5, for this testing dataset of PV systems are: average R^2 score of 0.933, average Normalized Root Mean Squared Error of 0.0565 and out of the 500 PV systems, 94.8% have a good R^2 score higher than 0.85. Must note that training and testing has been done on 5000 and 500 PV systems respectively. The whole dataset was not used due to computational time and memory limitations. Nevertheless, the peers are found from the whole set of 9480 PV systems.

5.3. Comparative Results

The process of improving the model involved constant comparison of results with the other models namely, old Solar Monkey model, new Solar Monkey model, and literature - P2P model. The detailed explanation for all four models can be found in chapter 4. Due to the computational time needed for using the new Solar Monkey model, the expected yields calculated by this model are limited. These results are available for 500 PV systems for up to three years (1118 days to be exact from 01st Nov 2017 to 22nd Nov 2020). The comparative results can be found in Figure 5.5 to Figure 5.8.

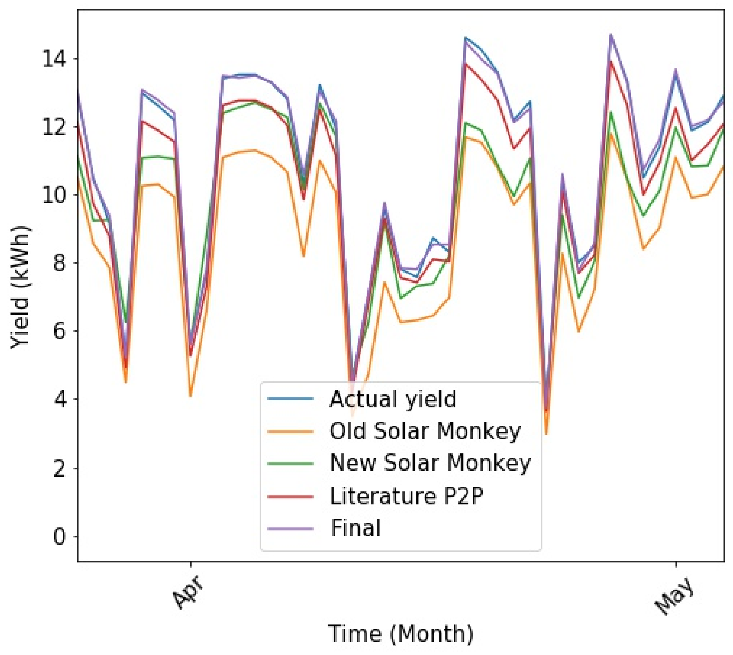


Figure 5.5: Actual yield versus expected yield for one PV system to compare different models

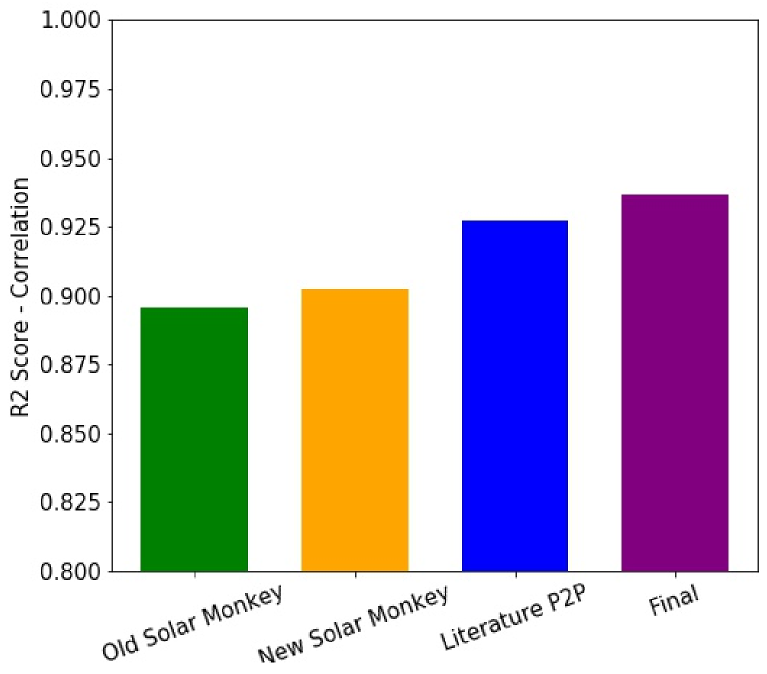


Figure 5.6: Average R^2 score comparison for different models, higher the R^2 score the better

To understand how the predictions for each model look, a representative temporal plot can be seen in Figure 5.5. The actual yield for the particular PV system is plot along with the expected yields obtained via the models - old Solar Monkey model, new Solar Monkey model, literature P2P model and proposed model respectively. The yields are plot for 40 days in the months of April - May. From

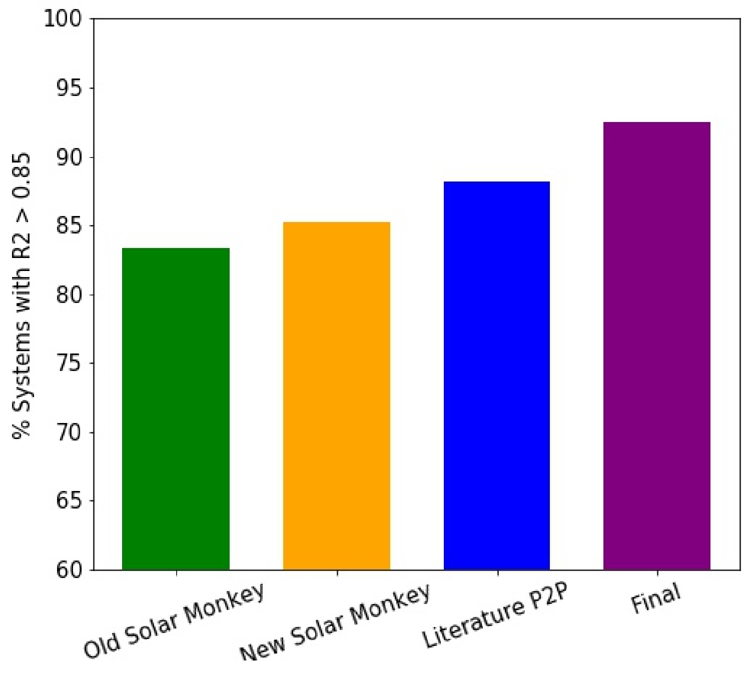


Figure 5.7: Percentage of PV systems with R^2 score greater 0.85 comparison for different models

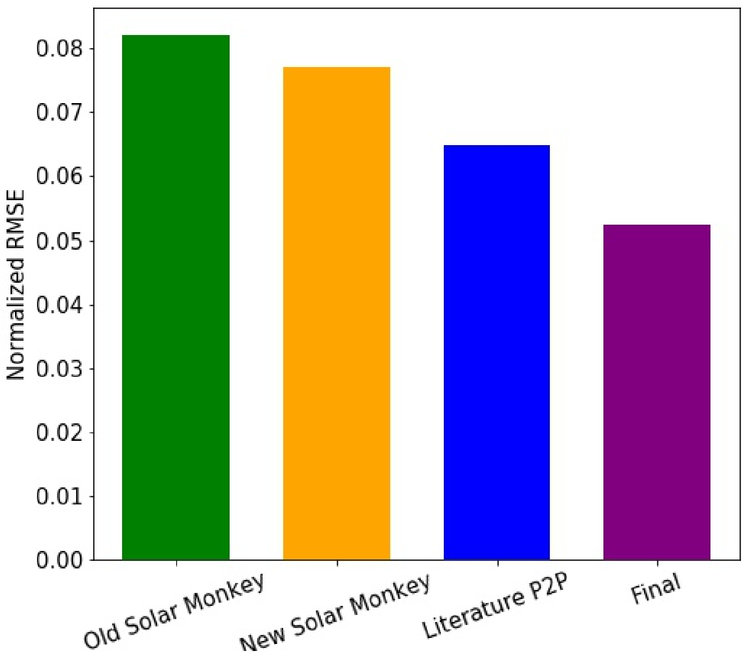


Figure 5.8: Average normalized RMSE comparison for different models, lower the NRMSE the better

the figure, it can be seen that both the Solar Monkey models underestimate the expected yields. On the other hand, the literature P2P model predicts accurately on some days but underestimates on other. While these observations are not always true for all systems at all times, they are just presented to give an idea about how the predictions look before explaining the comparative bar plots generated from results similar to the ones in Figure 5.5.

For the testing dataset, the bar plot results in Figure 5.6, Figure 5.7 and Figure 5.8, show that models based on peer-to-peer approach have higher accuracy than conventional ones. Both literature P2P model and the proposed model have higher average R^2 score and lower normalized RMSE than the Solar Monkey models. For the distribution of R^2 score per system, more systems have an higher R^2 score in peer-to-peer approach than the conventional approach.

The proposed model based on genetic algorithm is inspired by and improved upon the literature P2P model. Thus, as expected, amongst the two models using peer-to-peer approach, the proposed model has better metrics than the literature P2P model. As can be seen from Figure 5.6 and Figure 5.7, the R^2 scores are good for both with a slight edge towards the proposed model. There are more number of PV systems with a good R^2 score (0.85) in the case of proposed model. Finally, the normalized RMSE is lowest for the proposed model (refer Figure 5.8).

5.4. Use-case Analysis

In this section, results of the experiments carried out are presented. First, the minimum data requirements of the proposed model are explored. Followed by the experiment carried out to finalize the number of good peers required.

5.4.1. Minimum Data Requirement

Two experiments were planned in order to understand the minimum data requirements for the model. The first experiment was to determine the minimum number of PV systems required for training the model aka for the optimization function, and the second experiment was to determine the minimum number of days of data per PV system required for training the model. However, upon starting the first experiment, it was clear that the two experiments were not mutually exclusive. Hence, the two experiments were combined into one longer experiment.

As will be seen later in this section, 56 time consuming simulations were done for this experiment. While ideally cross-validation should be done for each simulation, it would mean increasing the simulation time multi-fold. Since this was not possible, as explained in section 2.3, it was confirmed that the training datasets used consisted of randomly chosen PV systems. These datasets have a similar quality distribution as the total dataset available to ensure that the training datasets represent the total dataset accurately. Similarly, the testing dataset also is a set of randomly chosen PV systems different from all training datasets, that represent a similar distribution as total dataset. It was kept the same for each simulation done.

For the experiment, the first parameter was number of PV systems, hence, the parameter could take a value of 100, 250, 500, 1000, 2000, 3000, 4000 or 5000 PV systems. The second parameter was number of days of data per PV system, thus, the parameter varies amongst 60, 91, 183, 275, 365, 549, and 730 days of data, i.e., 2 months, 3 months, 6 months, 9 months, 1 year, 1.5 years and 2 years respectively. Using both these parameters, a grid was formed with 8*7 different combinations of the number of PV systems and the number of days per PV system. Accordingly, 56 runs were set up. The results of these runs can be seen in Figure 5.9 to Figure 5.12.

To interpret the results of the above experiment, heatmaps (refer Figure 5.9 and Figure 5.10) were generated where each square color represents the relative value of the metric (here R^2 score and NRMSE). Additionally, the actual value of the metrics is also annotated in each square. From both the heatmaps for R^2 score and NRMSE, it can be seen that the number of systems does not have a strong influence on the metrics. However, if only a few months of data (< 6 months) is available per system, the performance degrades strongly. Thus, when it comes to training the model, number of days of data per system is important while number of PV systems used for training is not. Sufficient training of the model can be achieved for as low as 100-250 PV systems as well, provided one year data per system is available.

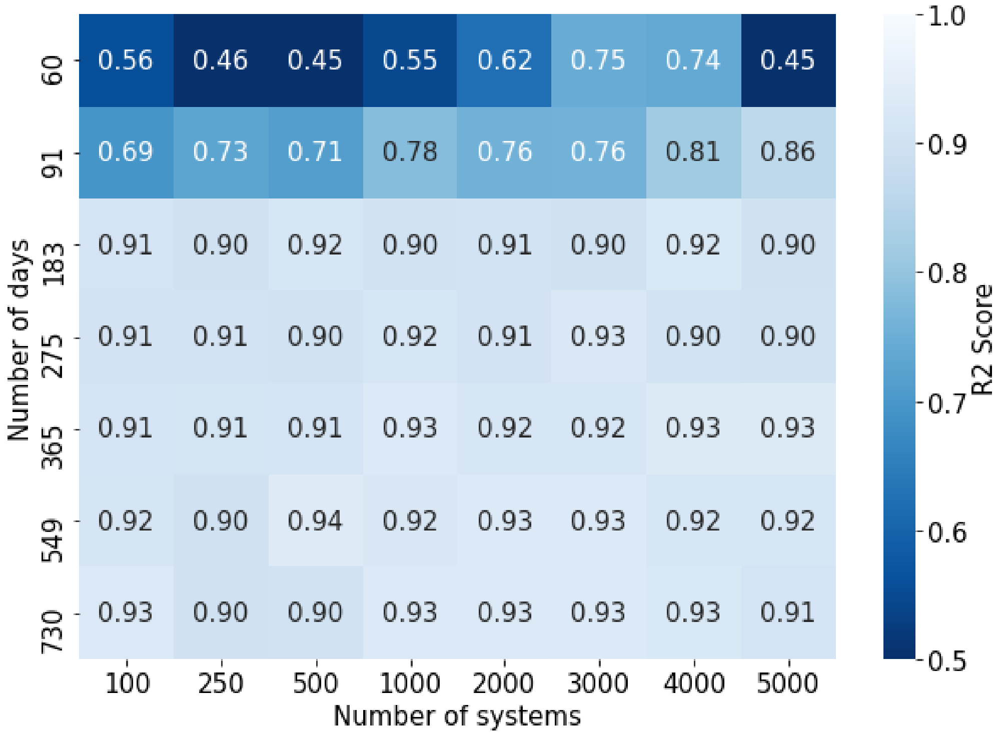


Figure 5.9: Number of days vs number of systems vs R^2 score heatmap for minimum data requirement; higher the R^2 score the better

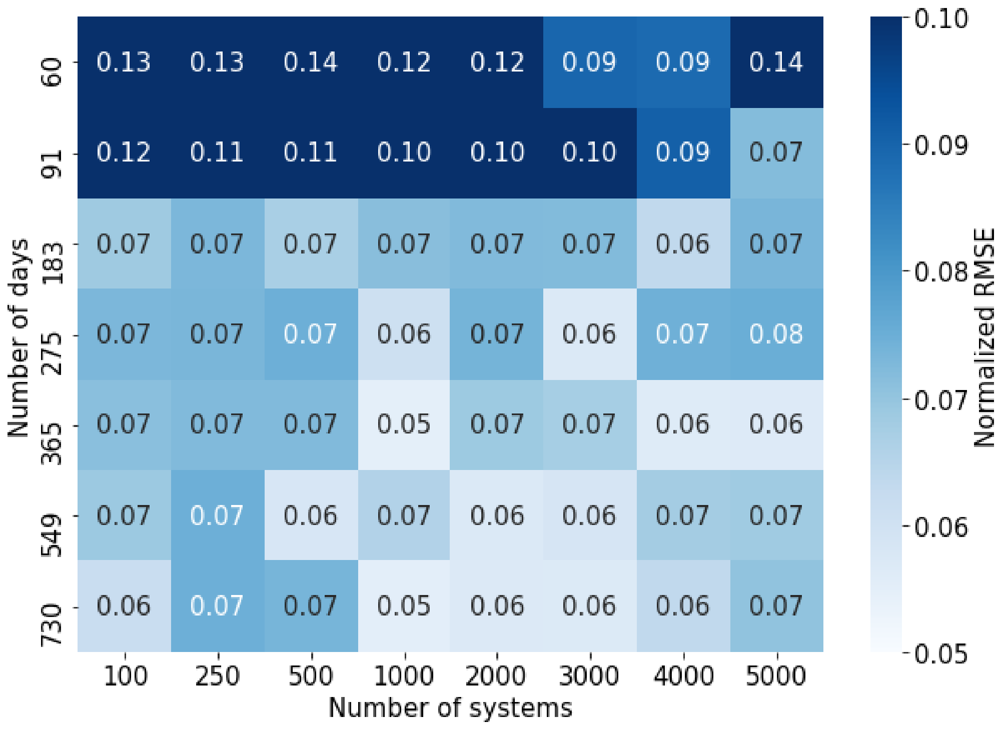


Figure 5.10: Number of days vs number of systems vs normalized RMSE heatmap for minimum data requirement; lower the NRMSE the better

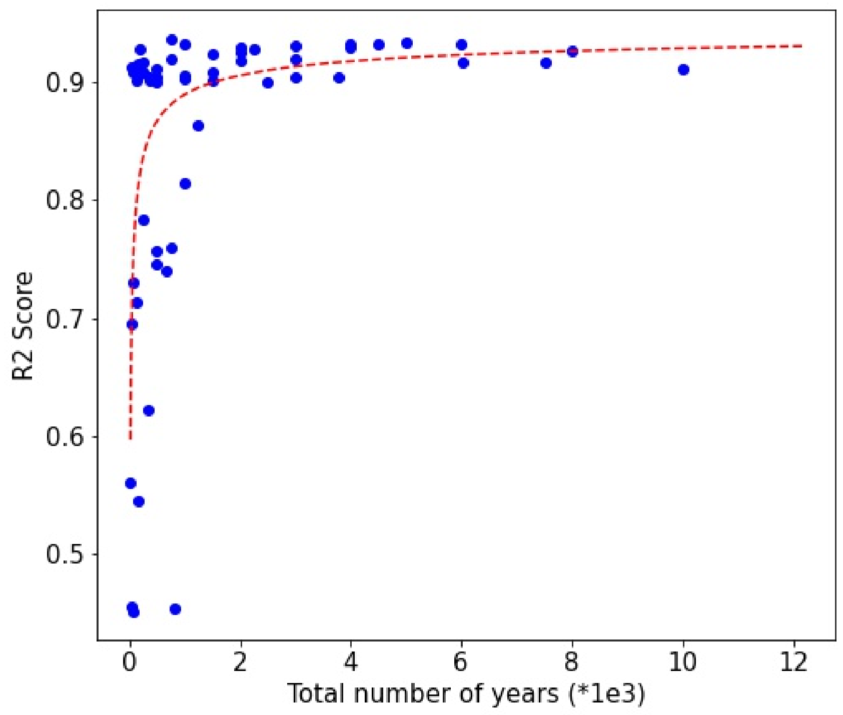


Figure 5.11: Minimum data required: R^2 score vs total data available

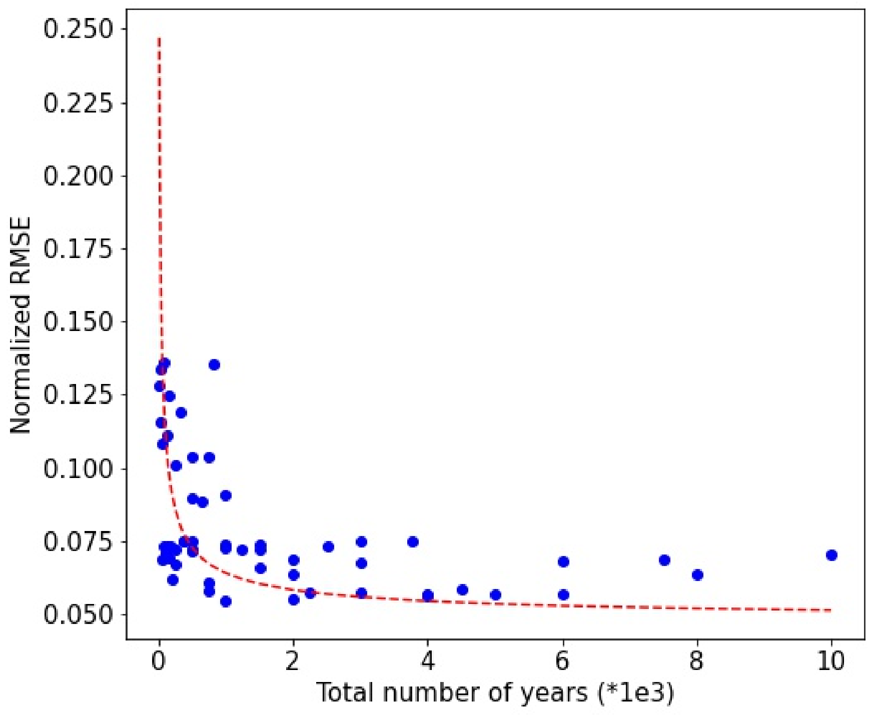


Figure 5.12: Minimum data required: normalized RMSE vs total data available

Although the experiment gives an understanding on the minimum data requirement for optimization, it was interesting to observe that even 100 PV systems might be fairly good enough to train. Thus, the results of the experiment were portrayed differently as seen in Figure 5.11 and Figure 5.12. For these graphs, the x-axis has the total number of days of data for each run. Thus, for grid point (4000, 365) in the heatmap, the corresponding value on x-axis would be 1460000 days that is equivalent to 4000 years of data. Along with the scatter plot, trend lines were generated using curve-fit function on the experiment results. According to the trend lines, data higher than 1700 years is more than sufficient for training the model, i.e., achieve R^2 score greater than 0.9. This translates to 1700 PV systems with one year data each or 850 PV systems with two years of data each and so on.

It is important to note that data lower than 1700 years also sometimes gives good optimization results, but other times it does not. So it might not be as reliable as data higher than 1700 years. The trend line saturates at around 4000 years data, which is effectively about 25% of the total data available.

Another smaller experiment was conducted under the minimum data requirement to test when peer-to-peer approach can be used for an entirely new PV system. This was tested on very few PV systems and the results of this are as follows. When a new PV system is added, energy yield data for first 30 days of that system is sufficient to locate the good peers for it (also called distance training). These peers can be used for estimating the expected yields for the next one to two months with fairly good accuracy. After this time, good peers need to be located again with the new information available. These new good peers can be used accurately for another few months and so on. This process needs to be repeated until sufficient amount of yield data is available, i.e., at least 6-9 months. After one year of active monitoring, the frequency needed for distance training is low as long as the system parameters stay fairly constant.

5.4.2. Number of Peers

During a number of simulations done while hyper-parameter tuning, it was observed that while the other optimization variables i.e., the weights for each "distance" do not usually change, the fourth variable number of peers varies a lot over different simulations. Thus, to figure out the best values for the number of peers a new experiment was done. Simulations were done with different number of peers with 1000, 2000, 3000 PV systems each having one year data. The results were used to plot a learning curve for number of peers.

The learning curves found in Figure 5.13 and Figure 5.14 have the metrics R^2 score or NRMSE on y-axis and the number of peers varying from 2-50 on the x-axis. From both graphs, it can be seen that the simulation has the highest R^2 score or the lowest NRMSE when the number of peers varies between 12-20 good peer systems. Lower than 12 number of peers is too less for accurate prediction due to higher weights on only a few peers. Higher than 20 number of peers also leads to lower accuracy in predictions most likely due to additional noise resulting from not so adequate peers.

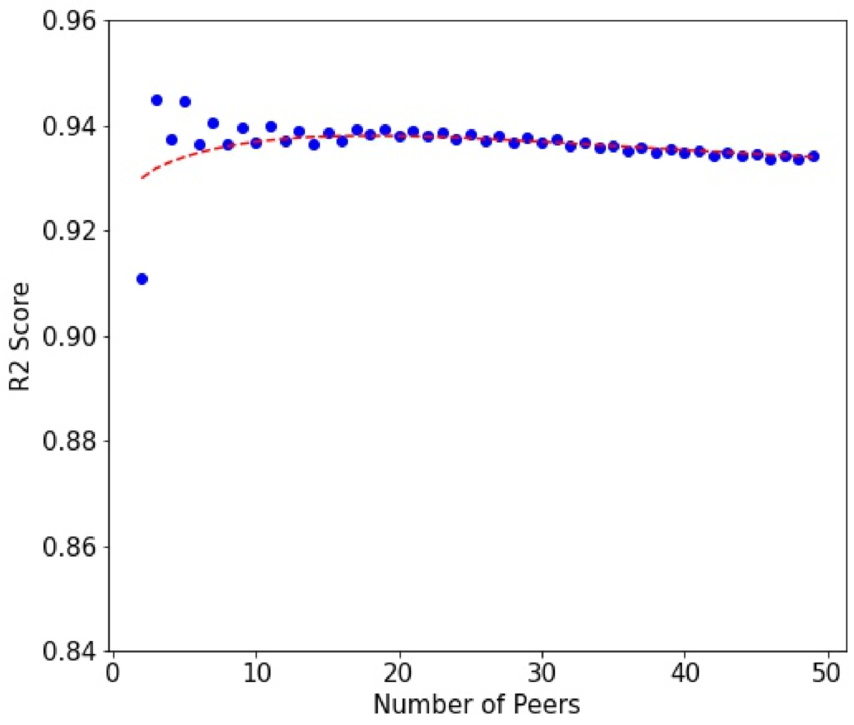


Figure 5.13: Number of peers learning curve with respect to R^2 score

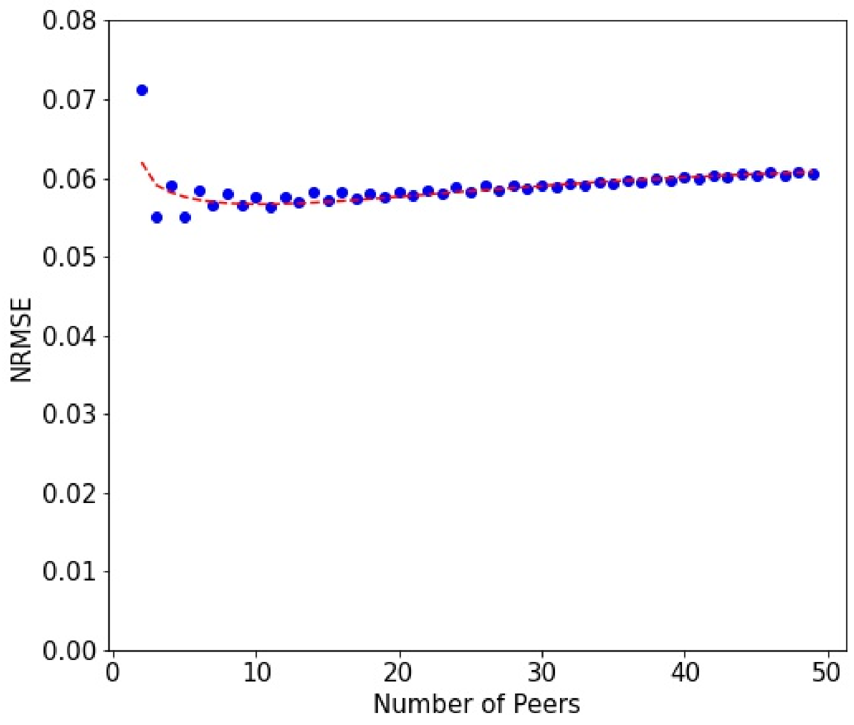


Figure 5.14: Number of peers learning curve with respect to NRMSE

6 Fault Detection

The objective of this thesis was to develop a model for monitoring of residential PV systems using a peer-to-peer approach. Monitoring of PV systems is a two-step process: first, to calculate the expected yield for the PV system and second, to compare the expected yield with the actual yield in order to figure out whether the PV system has any faults. After developing a model for the expected yield calculation as explained in the previous chapters, this chapter focuses on the second step, i.e., fault detection. To begin with, the approach to fault detection currently at Solar Monkey is explained in section 6.1 along with how the model developed in this thesis fits in that approach. Then, a categorization framework was created for easy fault detection. Details about the categorization can be found in section 6.2. Using the framework created, some PV systems were examined. They are presented in section 6.3. It is to be noted that the aim of this thesis is to show how peer-to-peer model can be used for fault detection, not to develop a fully functioning automatic fault detection algorithm.

6.1. Fault Detection Approach

6.1.1. Fault Detection by Solar Monkey

Monitoring is one of the key services offered by Solar Monkey. Currently, monitoring and fault detection is a manual process, also known as System Performance Analysis (SPA), where a member of Operations team filters out poorly performing systems based on certain criteria. These criteria were developed over time based on experience after analyzing the data of thousands of PV systems monitored by the company. The main concern with this process is that it is manual, time consuming, and largely subjective. According to Operations team, it could take up to 10 minutes per system to detect some faults. In addition to that, in past year, the number of PV systems monitored by the company has increased to about 15000 systems [7]. Considering the high time consumption of the process, Solar Monkey has focused on improving SPA by enabling swift fault detection to make it faster and semi-automatic.

Currently, SPA is a two-step process where first, different criteria are used to filter out good or poorly performing systems. The poorly performing PV systems need to be further investigated. The next step is where all available information about these PV systems is inspected. This includes system design information like total watt-peak, number of panels, panel type, inverter capacity, inverter type, panel inclinations and orientations, etc. This is also where all models including the peer-to-peer model developed during this thesis are used. In this step, the diagnosis of poorly performing systems found previously is done.

6.1.2. Thesis Approach

The peer-to-peer approach proposed in this report can be incorporated into the above explained two-step SPA process for fault detection. The criteria used for filtering poorly performing systems can be seen in Table 6.1. To start with, number of panel error, average inclination error and average

Criteria	**Unit**
System ID	-
Number of panel error	-
Average inclination error	^{o}
Average orientation error	^{o}
Actual yield	kWh
Actual yield previous years	kWh
Solar Monkey expected yield	kWh
Solar Monkey PF	-
Peer-to-peer expected yield	kWh
Peer-to-peer PF	-

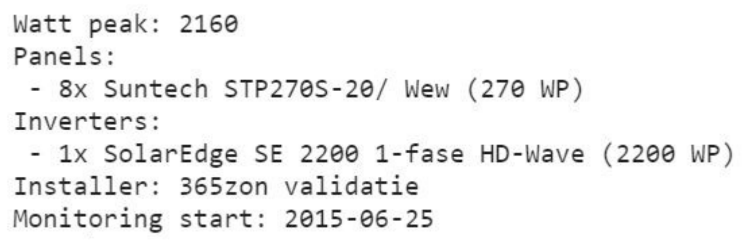
Table 6.1: Criteria used for filtering good or poor performing systems

orientation error denote whether there is a difference between the system design information in Solar Monkey database and the actual design installed on site. The actual design information is estimated using the height data and google earth images by image processing models which are beyond the scope of this thesis. Next, the actual yields of the PV system for last 7 days and last month are compared with Solar Monkey expected yields as well as peer-to-peer expected yields for respective time periods. This is the time period over which fault detection is carried out at the company. These yields are also used to calculate the Performance Factor (PF) for both according to Equation 6.1 used by the company. For the Solar Monkey expected yields, the equation is equivalent to the Performance Ratio. Finally, the actual yields of previous years for the same days are also used as criteria. This compares the performance of this year with previous years to confirm whether there is a new fault in the system.

$$Performance\ Factor = \frac{Actual\ Yield}{Expected\ Yield} \tag{6.1}$$

Once bad performing systems are identified, the next step is to take a comprehensive look at each PV system as shown in Figure 6.1 to Figure 6.3.

This step gives the system information like total watt-peak, number of panels, etc. as seen in Figure 6.1. This particular example system of total 2160 Wp has 8 Suntech panels with a SolarEdge inverter of capacity of 2200 Wp. Along with this information, the design image is compared with the actual image to manually confirm that the panels are installed on the correct roof as well as whether the number of panels in design match the number of panels actually installed. In this example, for the design image generated on the start date of monitoring has 8 panels as designed by the installer. Additionally, when the satellite image from 2018 is compared, it also shows 8 panels truly installed on the correct roof. This check is necessary as sometimes the installers (the customers of Solar Monkey application) might design the PV system on the wrong roof or they fail to update the design in the application which leads to a mismatch.



```
Watt peak: 2160
Panels:
 - 8x Suntech STP270S-20/ Wew (270 WP)
Inverters:
 - 1x SolarEdge SE 2200 1-fase HD-Wave (2200 WP)
Installer: 365zon validatie
Monitoring start: 2015-06-25
```

Figure 6.1: Fault detection: system design information

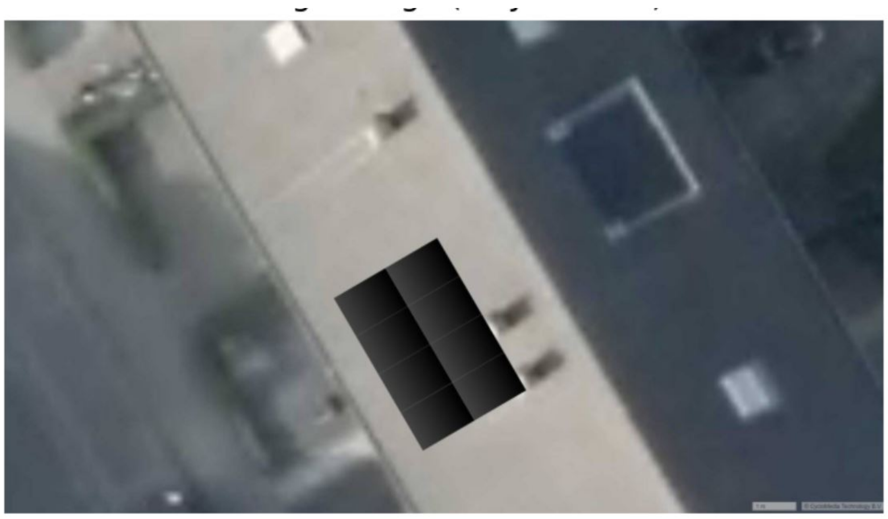


Figure 6.2: Fault detection: design image in the company software completed on start date of monitoring

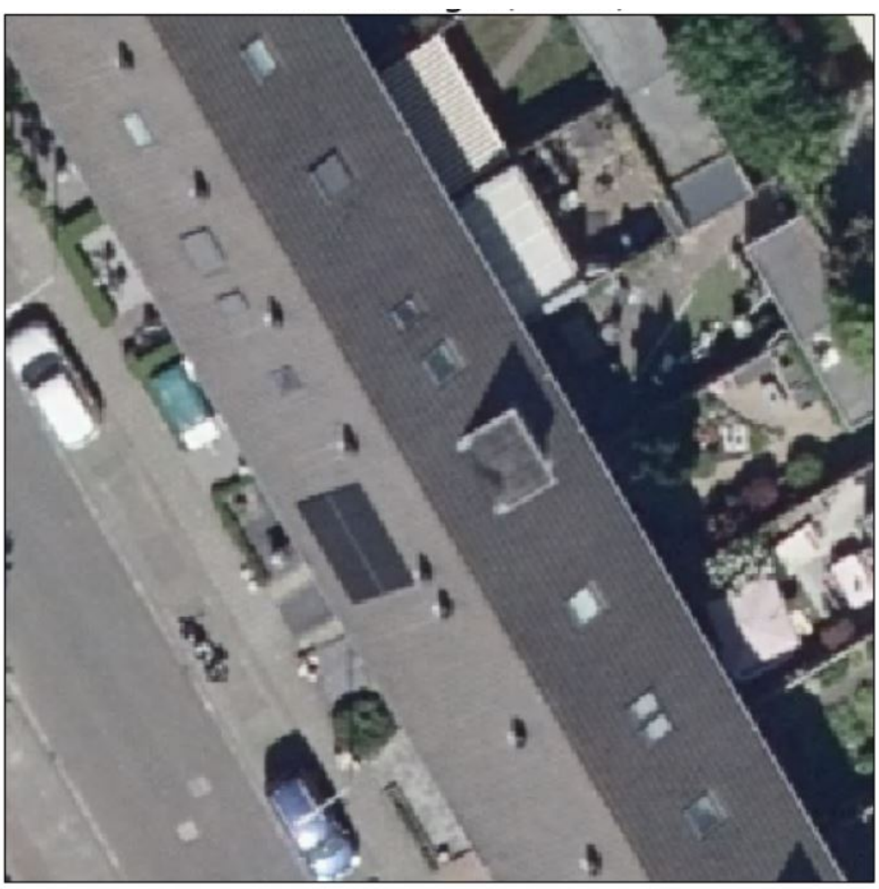


Figure 6.3: Fault detection: latest satellite image available (2018) [55]

While the Figure 6.1, Figure 6.2 and Figure 6.3 only indicate the design information, some automatic checks are used to enable quicker detection. There are eight checks already implemented and one check that was still to be completed at the time of writing this report. The first four checks are to compare the design information in the Solar Monkey app provided by the installer to the actual installation on site. Thus, they are used to detect system design mismatch. They verify the panel azimuth or orientation, roof angle or panel tilt, system position on roof, and inverter brand. The last four checks are for fault detection. They are given below.

1. **Solar Monkey expected yields:** Quantify the performance factor for the last 30 days using Solar Monkey model expected yields.

2. **Year-over-year yields:** Compare the Solar Monkey expected yields of the last 30 days with the actual yields from the last year for the same 30 days.

3. **Sizing yields:** Estimate the system size for the last 7 days by evaluating the size of system, i.e., the number of panels that would generate energy yield equal to the Solar Monkey expected yields for that 7 days. [56].

4. **Peer-to-peer yields:** Quantify the performance factor for the last 30 days using peer-to-peer expected yields calculated by the model presented in this report.

The difference between system design mismatch and fault detection would be that in case of system design mismatch, the solution is to change the design to match the actual installation. It is usually not possible to change the actual installation once it is done. Comparatively, in the case of fault detection, there is a drop in performance due to a problem with the system. It is assumed here that the system was performing as expected before the problem or fault occurred. It is essential to separate the two for the 'Zonnegarant' service of Solar Monkey. In this thesis, the last four checks are used to explain how peer-to-peer approach can be used with other models for fault detection.

Using the above two-step process along with the expected yield and actual yield temporal plots, some PV systems were scanned for finding under- and over-performing systems as well as to scrutinize these systems to diagnose the most common faults encountered. Based on this analysis, a fault categorization framework was developed, and details are provided in the next sections.

6.2. Fault Categorization

As mentioned above, for finding faulty systems, the complete database of about 12300 PV systems should be searched to filter out good PV systems and keep only very poorly performing systems to be checked individually. However, for this section, about 120 randomly chosen PV systems were checked to gain more insight into how often these faults occur. At the same time, based on the observations, certain guidelines were developed to categorize systems based on their faults.

As mentioned, a total of 120 PV systems were chosen for this experiment. They were chosen from a database of 9500 systems randomly selected to ensure that they have a good distribution of new and old systems. These 120 PV systems were monitored for the month of June 2021. Although the energy yields for the entire lifetime of the PV system was looked at, for the categorization, only the outcome of monitoring in the month of July is used. The reason for choosing June was that in the summer months, PV systems are expected to produce the most energy yield in the year. A malfunction in summer could lead to significantly large energy and monetary loss for the owner than a malfunction in winter. Furthermore in winter, fault detection becomes relatively harder due to very low energy yields.

As explained above, four criteria were used to separate the systems into groups, namely, Solar Monkey expected yields, year-over-year yields, sizing yields, and peer-to-peer yields. The fault categories found were no fault, missing data, under-performance, over-performance, and false positive as usual faults along with additional, peer-to-peer failure. Each of these categories can have multiple combinations of the four criteria and they may or may not lead to a different fault diagnosis as can be seen in Table 6.2.

To begin with, a baseline of no fault would be when all four criteria are within an acceptable range, here between 93% and 120%. There is also a small possibility for 'False Negative' but amongst the 120 PV systems considered, none presented a clear case of 'False Negative'. Next, for missing data fault, there were two observations. First, if there have been a few days of missing data within the last 7 days, only sizing yields can flag it. This is because all other criteria are calculated for 30 days, hence the impact of a few days of missing data is not noticed. However, if the missing data persists for more than 15 days, it is flagged by both year-over-year yields and peer-to-peer yields. Interestingly, here neither Solar Monkey yields nor sizing yields present any problem because when no data is received from the inverter, the code automatically equates the actual yields with Solar Monkey expected yields. This is done, perhaps because it is very easy to detect missing data and neither of these criteria are necessary for it. Thus, both these criteria become closer to 100%.

Then, for under-performance, there were again two observations. In the case where the fault occurred within the last one year, it was reflected in all four criteria as lower actual yields than the expected criteria. Yet if the fault persists for longer than one year, year-over-year yields fail to flag the fault. This is anticipated, as year-over-year yields compare the actual yields for the last 30 days with the actual yields from previous years for the same 30 days. If the same fault existed last year, year-over-year yields would expect them this year as well. A similar observation can be made for over-performance, where it is reflected in all four criteria or all except year-over-year yields as actual yields higher than expected criteria yields. Depending on the magnitude of under-performance, the diagnosis could either be a small, temporary problem or else it could be due to broken panels or string as was the case in subsection 6.3.2. If it is the case of broken panels, the magnitude of PF should drop by an equivalent amount. For example, if 3 panels out of total 6 in a PV system break down, the performance usually drops to around 50%, depending on the string configuration. Similarly, on the rare occasion of over-performance as described in subsection 6.3.3, depending on the magnitude of the increase in actual yields, it could be a case of system size change, i.e., new panels may be added to an existing PV system. This should always be verified by taking a look at the latest satellite images when available.

The fifth category to discuss would be a false positive. It is not a true fault as here the Solar Monkey yields, year-over-year yields, and/or sizing yields suggest that the system is either under- or over-performing but in reality it is working same as ever. In case the system is older than a year, year-over-year yields and peer-to-peer yields are within an acceptable range, whereas the other two criteria are either high or low. On the other hand, if the system was installed less than a year ago, only peer-to-peer yields are in an acceptable range with year-over-year yields being unavailable. Both Solar Monkey yields and sizing yields flag this as a fault, but on further investigation it was found that both of them have been consistently low or high from the time the system was installed. Therefore, as will be explained in subsection 6.3.4, there is no malfunction in the system but most likely a mismatch between the actual installation and the system design details in the company database. Another case of false positive is when all except the year-over-year yields are within the acceptable range. Here, the possible diagnosis is that either due to, for example, unusually sunny or cloudy days, the actual yields are higher or lower than the last year. Since both Solar Monkey yields and sizing yields depend on weather data, they adjust accordingly to the unusual irradiance. Similarly, in this case, most of its peer systems likely experienced the unusual weather and thus, peer-to-peer yields also finds itself within the acceptable range.

The final category and the one checked with caution would be the case of peer-to-peer failure. In this case, all except the peer-to-peer yields are within acceptable range, however, peer-to-peer yields are either too high or too low. It was noticed that this occurred mainly due to poor distance training, i.e., on data of poor quality or because the system was installed in the last winter and was too new to be properly trained. Another possible cause could be for very old systems (> 5 years) with some degradation in the performance. Since the distance training for peer-to-peer was done in early years of the PV system, it does not adjust itself to the performance degradation over the years and thus, detects it as a fault. Compared to that, both Solar Monkey yields and sizing yields already include system decay per year in the calculation. For the year-over-year yields, the degradation might be too slow to drop below the acceptable limit.

Another goal for reviewing the PV systems was to get an overview of how often these faults occur. As seen in Table 6.2, out of the 120 PV systems checked, about 63% have no faults. The most occurring fault detected was missing data, with 17% systems experiencing the problem on a minor (14.5%) or major level (2.5%). Only about 6% systems might have under-performance issues while

Table 6.2: Criteria for fault categorization of PV systems

Fault	Solar Monkey Yields	Year-over-Year Yields	Sizing Yields	Peer-to-Peer Yields	Occurrence
No Fault	✓	✓	✓	✓	62.5%
Missing Data	✓	✓	Low	✓	16.7%
	✓	Low	✓	Low	
Under-performance	Low	Low	Low	Low	5.8%
	Low	✓	Low	Low	
Over-performance	High	High	High	High	0.8%
	High	✓	High	High	
False Positive	High/Low	–	High/Low	✓	7.5%
	High/Low	✓	High/Low	✓	
	✓	High/Low	✓	✓	
Peer-to-Peer Failure	✓	✓	✓	Too High/Too Low	3.3%

over-performance occurs rarely. About 7.5% systems had the case of false positives, where peer-to-peer has the biggest advantage. Finally, for about 3% systems, peer-to-peer suggests faults incorrectly. Often these systems need to undergo distance retraining again in order to ensure proper peer-to-peer yields.

It was expected to find some examples of inverter limited systems amongst the 120 checked. There were a few systems present whose inverter capacity was lower than the total system watt-peak. Yet none of them seemed to have an energy generation high enough to be limited by the inverter. Please note, that while these systems represent the total dataset, the statistics should be taken with a grain of salt. With addition of newer systems to the database, the relative percentages could change.

6.3. Fault Detection Examples

In this section, some examples of PV systems with certain faults are presented. Examining these PV systems will help understand how peer-to-peer approach fits into the categorization framework and can be a useful addition to fault detection. While peer-to-peer expected yields and PF notice the faults in PV systems in most cases, they are used together with other checks mentioned above to determine the kind of fault i.e., for fault diagnosis. In the following subsections, examples are provided for some faults discussed in section 6.2.

6.3.1. No Fault

The PV system discussed here has had no particular faults in the last 30 days (here for June 2021). This example is to create a baseline before focusing on systems with faults. Looking at Table 6.3, all boxes are ticked. Since this file is still work in progress, sometimes some of the checks may not be available for all PV systems in the database. When the four fault detection checks are ticked, it almost always means that the system has no faults for last 30 days. To confirm this, in Figure 6.4, the temporal plot for last six months can be seen.

Table 6.3: Categorization checks when no fault present

Criteria	System Mismatch	Solar Monkey Yields	Year-over-Year Yields	Sizing Yields	Peer-to-Peer Yields
Value	✓	102.6%	96.7%	101.5%	99.5%
Check	✓	✓	✓	✓	✓

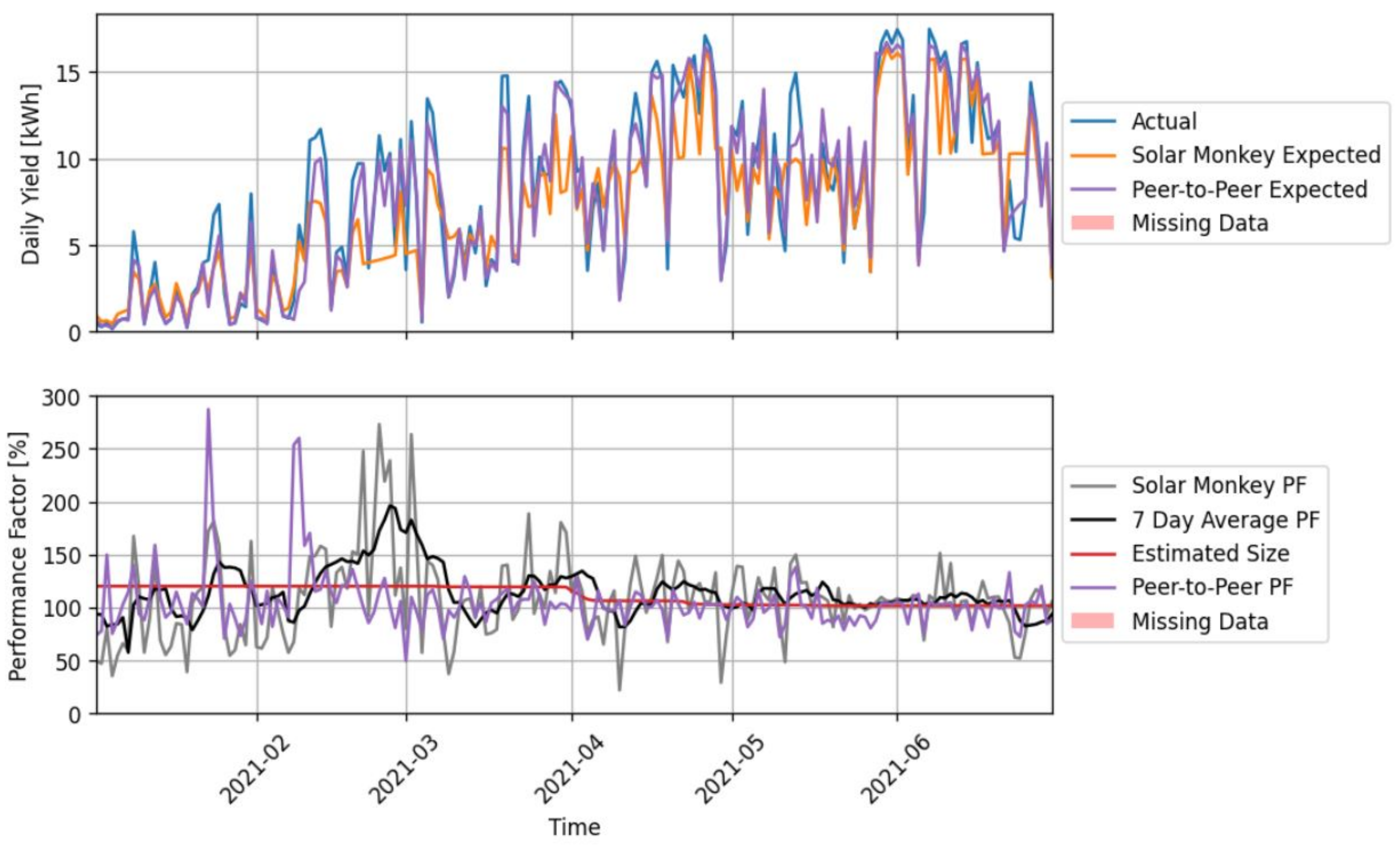


Figure 6.4: Temporal yield plots: no fault system

As can be observed in the first plot of Figure 6.4, Solar Monkey expected yield as well as peer-to-peer expected yield follow the actual yield very closely in the last 30 days. Thus, the system is performing as expected. The second plot in this figure shows the Performance Factor in % on y-axis. This plot has the daily Solar Monkey PF with a 7 day average for a less noisy trend along with the daily peer-to-peer PF. Estimated size is also depicted in the plot which is based on the sizing yields check explained earlier. This estimated size also denotes 100% which means the system size matches the actual yield. For convenience, the missing data is also always flagged in both plots. Thus, it is not presented here separately.

6.3.2. Broken Panels or String

The next type of fault is under-performance, here due to broken panels or strings. This example PV system has 58 panels with a total capacity of 18095 Wp. It has separate sets of 31 panels and 27 panels installed with different types of panels as well as different orientation as can be seen in Figure 6.5.

Amongst the system design mismatch checks enlisted in Table 6.4, all system mismatch checks give a green flag. Next, all the fault detection checks give a red flag with all suggesting the system has been under-performing up to 40%. This is also apparent from the temporal yield plots in Figure 6.6. There is a sudden drop in both the Solar Monkey PF and peer-to-peer PF plot around the month of September in 2020. The performance factors have been below 50% since then until at least the 30th of June 2021. Next, as the fault occurred within the last one year, year-over-year yields are able to flag the issue. In case this fault persists beyond September 2021, year-over-year yields will not be able to flag the fault anymore.

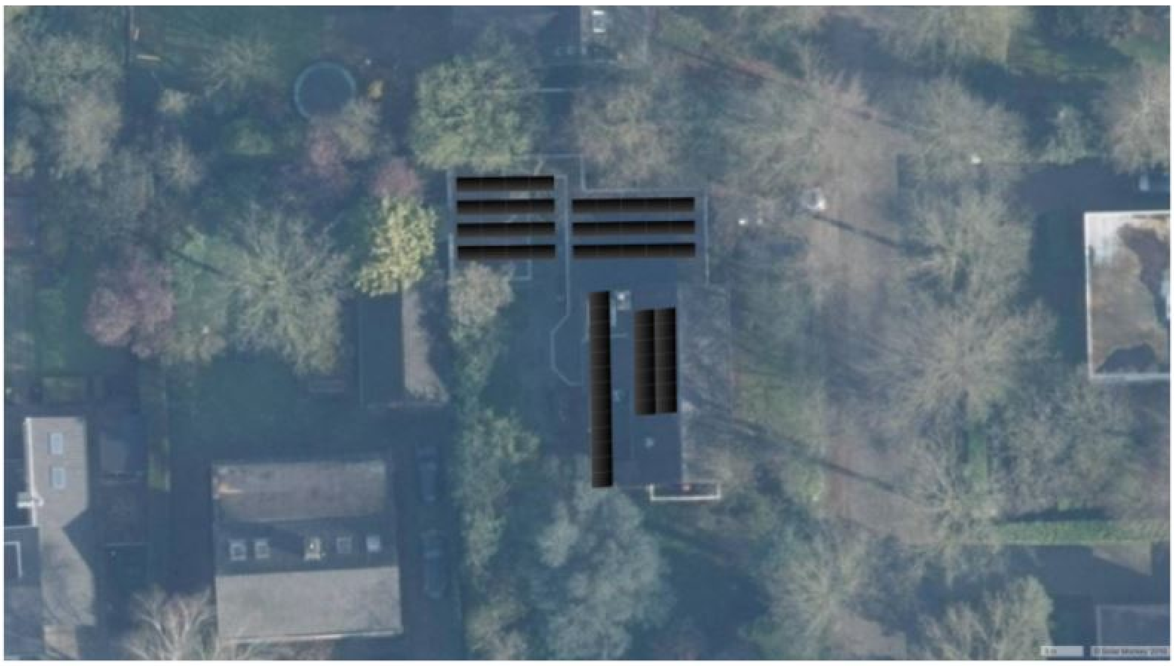


Figure 6.5: Design image in the company software completed on start date of monitoring

Despite the drop in both the performance factors in September 2020, the estimated size detects the drop by November 2020 (refer Figure 6.6). It simply lags in flagging the fault. It is also worth noting that for the winter months of November-December-January, it partly recovers close to a 100%. This could be due to lower yield generations in the winter months where the relative difference between actual and expected yields becomes quantitatively lower than summer months.

While all these checks simply suggest that the system is under-performing, the fact that under-performance has been consistent or in other words the performance factor being consistently around 40%, suggests that only a part of the PV system has developed a fault. This can also be corroborated by the sizing yields which estimate the system size as 25 panels instead of 58 ($\approx$ 40%) that were installed. This indicates that part of the system is broken meaning some panels or string likely has broken down.

Table 6.4: Categorization checks when broken panels or string present

Criteria	**System Mismatch**	**Solar Monkey Yields**	**Year-over-Year Yields**	**Sizing Yields**	**Peer-to-Peer Yields**
Value	✓	37.0%	47.0%	40.1%	46.9%
Check	✓	x	x	x	x

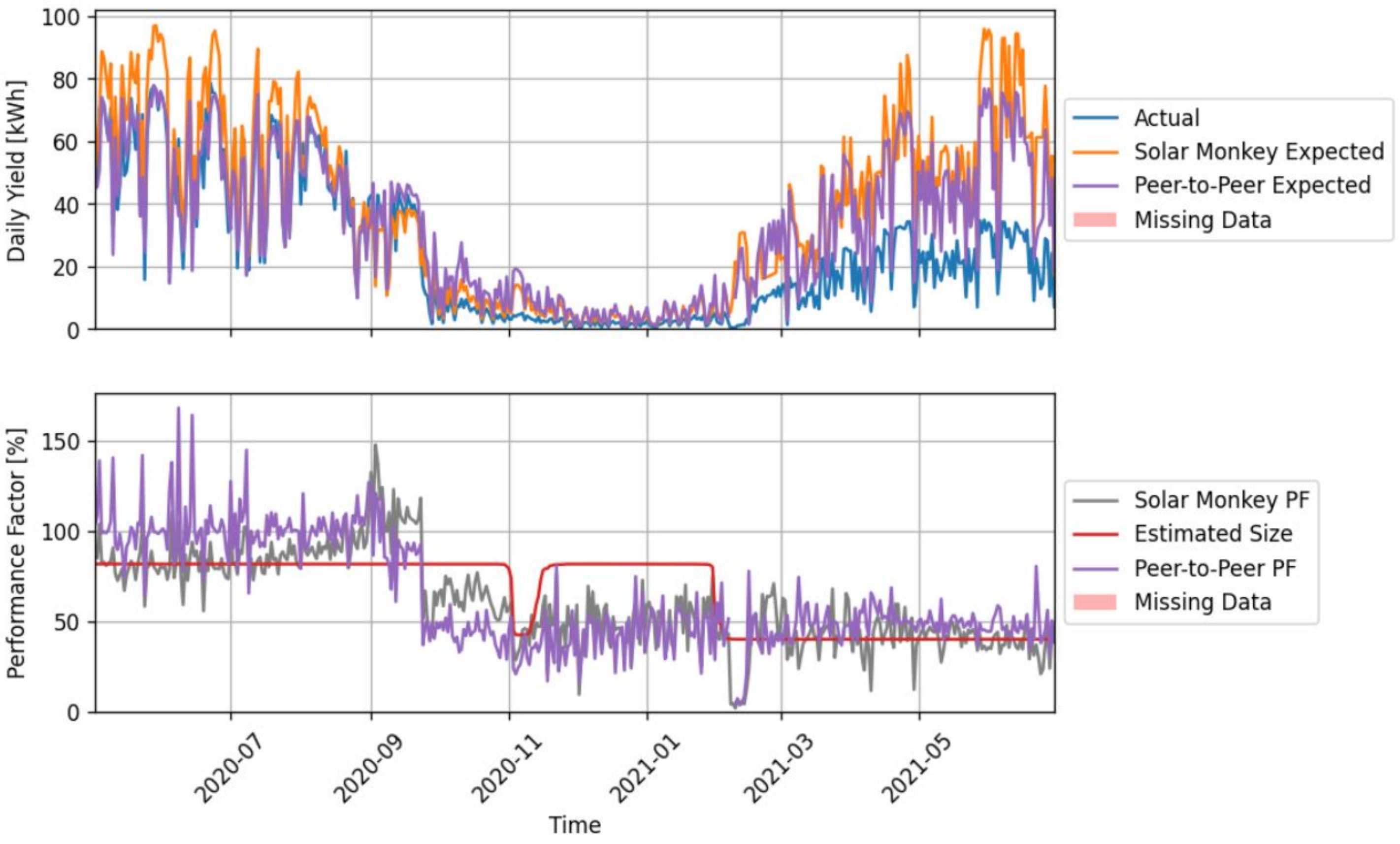


Figure 6.6: Temporal yield plots: broken panels or string

6.3.3. System Size Change

Due to the capital-intensive nature of PV systems, it can be preferred by residential owners to install them in steps on their roofs. Thus, a few years after PV system has been installed at the location, there have been instances when the owner has decided to increase capacity of the PV system by adding more panels or strings. This is usually not reported back to Solar Monkey monitoring system which leads to system over-performing. In this particular example, according to Solar Monkey database, the system has 8 panels as seen in the design image in Figure 6.7. However, in the latest satellite google image in Figure 6.8, clearly three new panels have been added in addition to the existing 8 panels [55].

Table 6.5: Categorization checks when system size change present

Criteria	System Mismatch	Solar Monkey Yields	Year-over-Year Yields	Sizing Yields	Peer-to-Peer Yields
Value	No. of Panels	143.0%	106.9%	136.2%	139.3%
Check	x	x	✓	x	x

Although it is simply convenient to look at the latest satellite image to verify the number of panels, which is also what one of the system design mismatch checks does (refer Table 6.5), it is limited by the availability of the satellite image. For instance, the image could be older and may not yet reflect the new addition of panels. Thus, it is better to look at all the checks for common issues as in Table 6.5. Considering the fault detection checks, all except year-over-year yields flag the system as over-performing in the last 30 days. When the yield plots in Figure 6.9 are checked it is clear why this is the case. This system has been over-performing since April of 2019 by 140%. As verified by Figure 6.8, 140% is also equivalent to a size of 11 panels instead of 8.

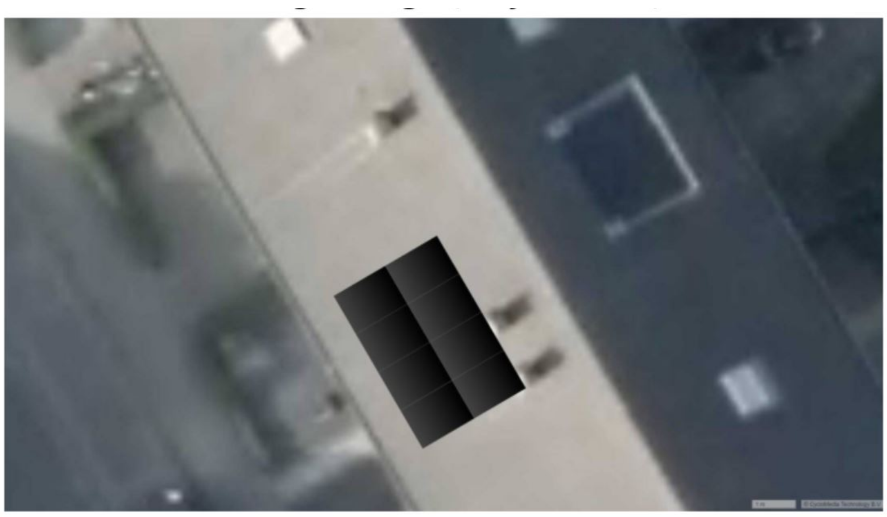


Figure 6.7: Design image in the company software completed on start date of monitoring

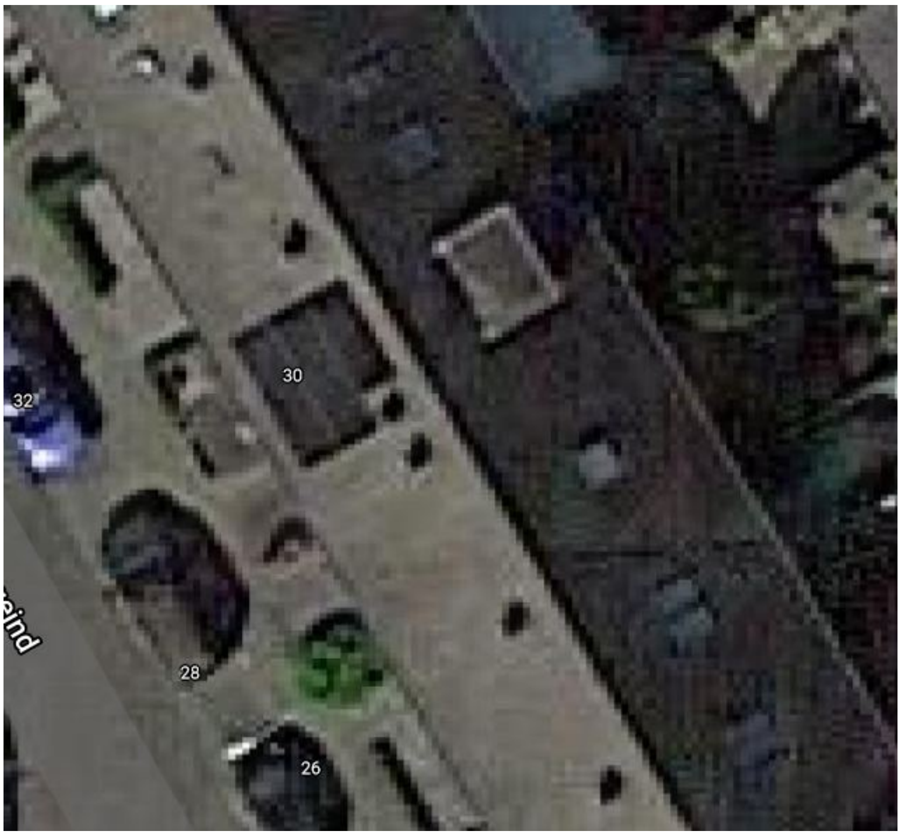


Figure 6.8: Latest satellite image available (2020) [55]

A particular advantage that peer-to-peer yields has in such a scenario is that its usefulness can be tweaked by changing the time period over which the distance training is done. Distance training is essentially the calculation of "total distance" in order to find the best peer systems for the focus system. Since the "yield distance" part of the calculation is time-dependent, distance training can be altered for different fault detection goals. To elaborate, in the above case, distance training was done prior to the system change, i.e., before April 2019. Thus, here as well as normally any change post the distance training period is red flagged as an issue. However, in this special case, system size change is not technically a fault. It is a change in system itself but there is no malfunction in the

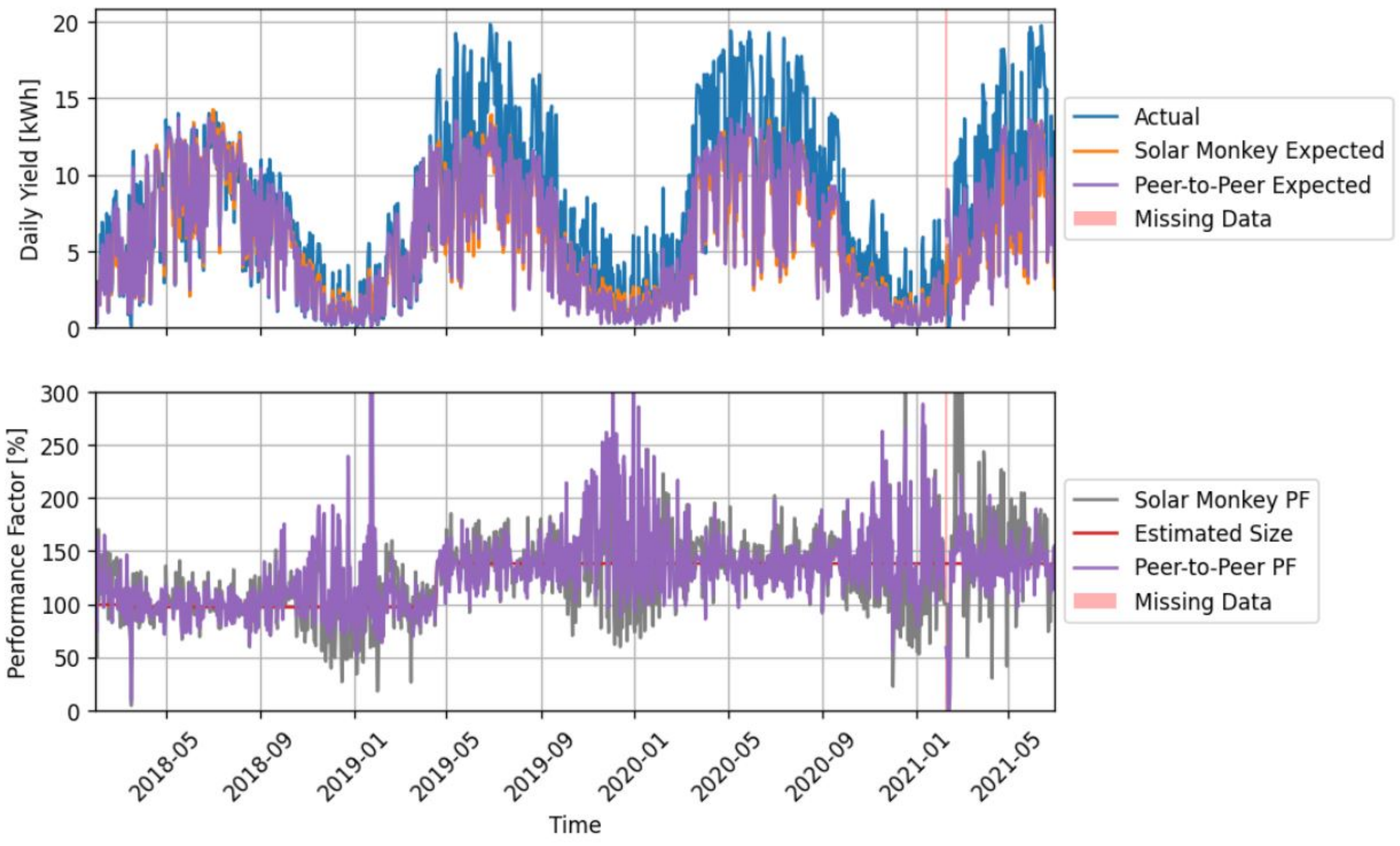


Figure 6.9: Temporal yield plots: system size change

system that degrades the performance. Hence, if the distance training is done after the system change, it will find different peer systems that indicate this focus system is performing as expected. As seen in Figure 6.10, this system underwent distance training again in April 2019, post the system size change. While the Solar Monkey yields and sizing yields checks in Table 6.6 still suggest the system is over-performing, peer-to-peer yields do not flag any faults. Thus, in the temporal yield plots in Figure 6.10, the peer-to-peer expected yields closely match the actual yields, whereas Solar Monkey yields under-predict. This distance retraining is especially important if a true fault occurs now, it will be flagged by peer-to-peer yields as under-performing while the Solar Monkey yields might suggest that the system is simply less over-performing.

Table 6.6: Categorization checks when system size change present, after distance retraining

Criteria	**System Mismatch**	**Solar Monkey Yields**	**Year-over-Year Yields**	**Sizing Yields**	**Peer-to-Peer Yields**
Value	No. of Panels	143.0%	106.9%	136.2%	99.5%
Check	x	x	✓	x	✓

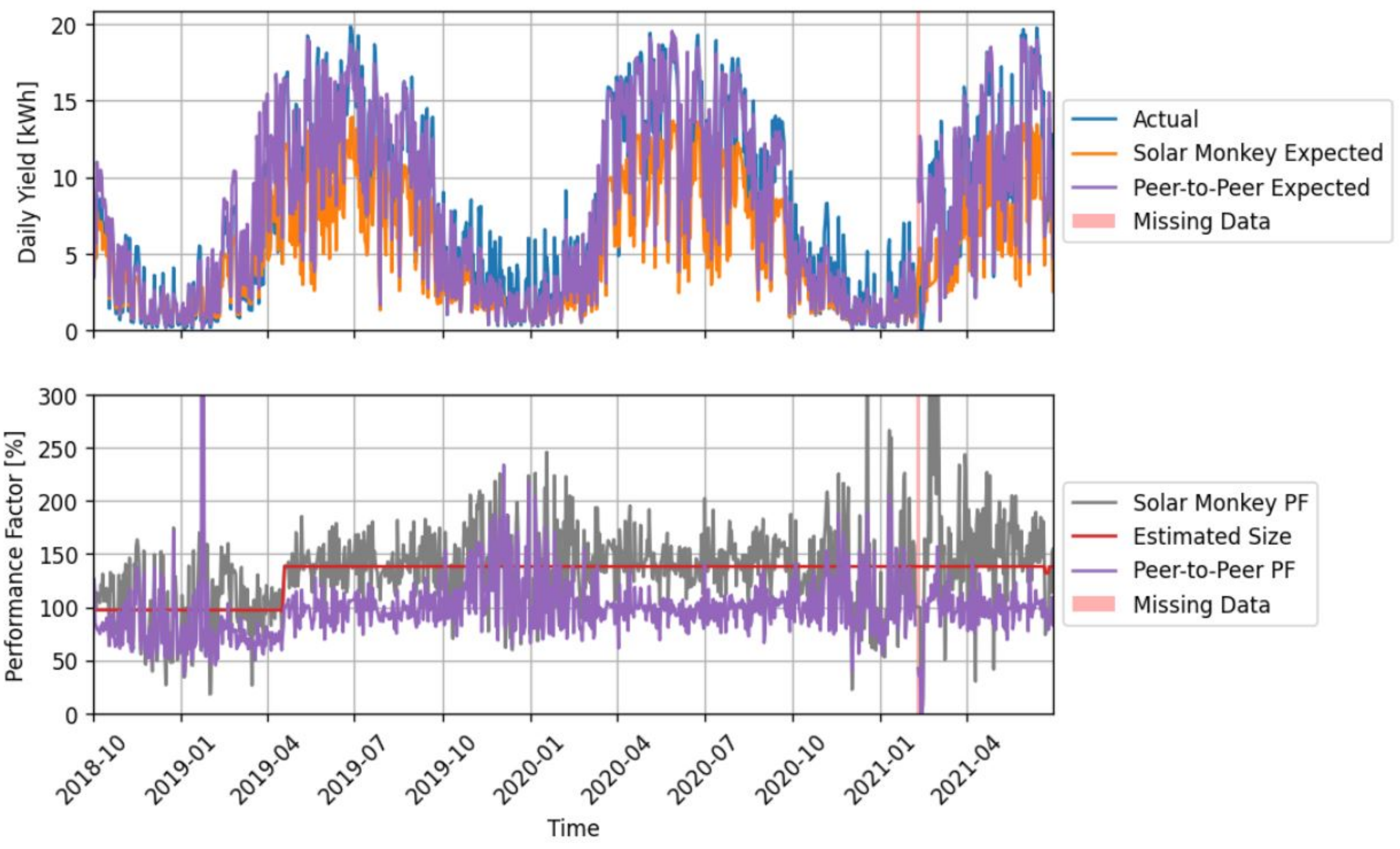


Figure 6.10: Temporal yield plots: system size change after distance retraining

6.3.4. False Positive

One of the strengths of using peer-to-peer yields for fault detection is the ability to detect false positives. A false positive means that the Solar Monkey yields suggest that there is an issue with the system despite the system not having any particular malfunctions. To illustrate, the PV system has 14 panels according to the database. To confirm that the same number of panels were actually installed, Figure 6.11 and Figure 6.12 are examined. Clearly, the panels match the design details.

Furthermore, the common system design mismatch checks (refer Table 6.7), number of panels, panel orientation, panel inclination, system location on the correct roof as well as the inverter brand suggest no problems in the system. However, from the fault detection checks, Solar Monkey yields, year-over-year yields as well as sizing yields suggest that the system is under-performing. When these three checks give a red flag, the first conclusion would be that there might be a few broken panels. However, the peer-to-peer yields suggest that the system is performing perfectly fine with a 97% performance factor in the last 30 days. To make sense of this discrepancy, the yield plots are inspected.

From Figure 6.13, it can be deduced that the system has been under-performing since the day it was installed. Nevertheless, since the peer-to-peer yields were distance trained during this alleged under-performance, the peer-to-peer approach does not consider this to be any fault. From the yield plots, it can be said that either the system was never installed properly or else there is a mismatch in the system design details and the actual installation, which cannot be visually verified like panel type and capacity. This suggests that the under-performance flag by Solar Monkey yields is a false positive as the system has been performing as historically expected according to peer-to-peer yields and there is no malfunction in the system that might degrade its performance.

This kind of mismatch fault is the cause for many persistently under-performing systems. These systems are a time-sink for the Operations team and thus it is key to simplify the diagnosis. Peer-to-peer yields can be very handy in such scenarios.

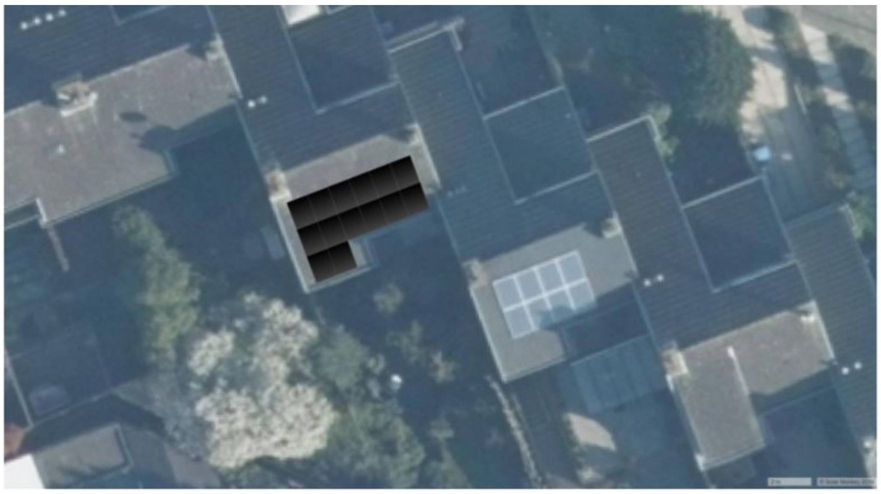


Figure 6.11: Design image in the company software completed on start date of monitoring

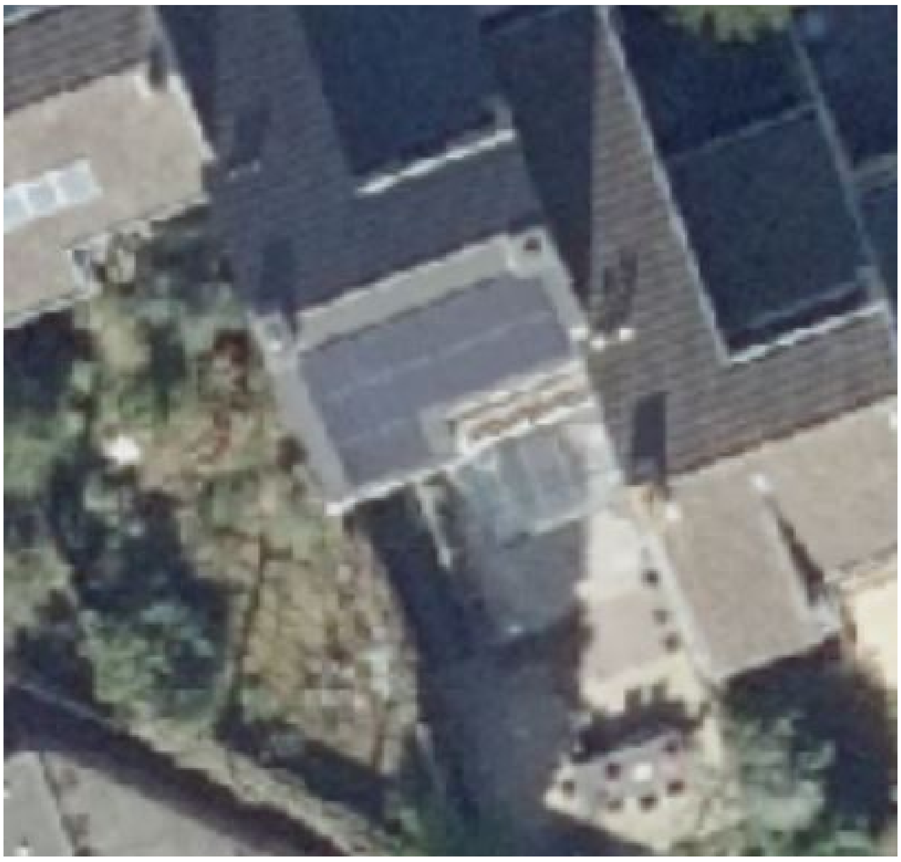


Figure 6.12: Latest satellite image available (2020) [55]

Table 6.7: Categorization checks when false present

Criteria	System Mismatch	Solar Monkey Yields	Year-over-Year Yields	Sizing Yields	Peer-to-Peer Yields
Value	✓	53.7%	82.0%	55.8%	97.0%
Check	✓	x	x	x	✓

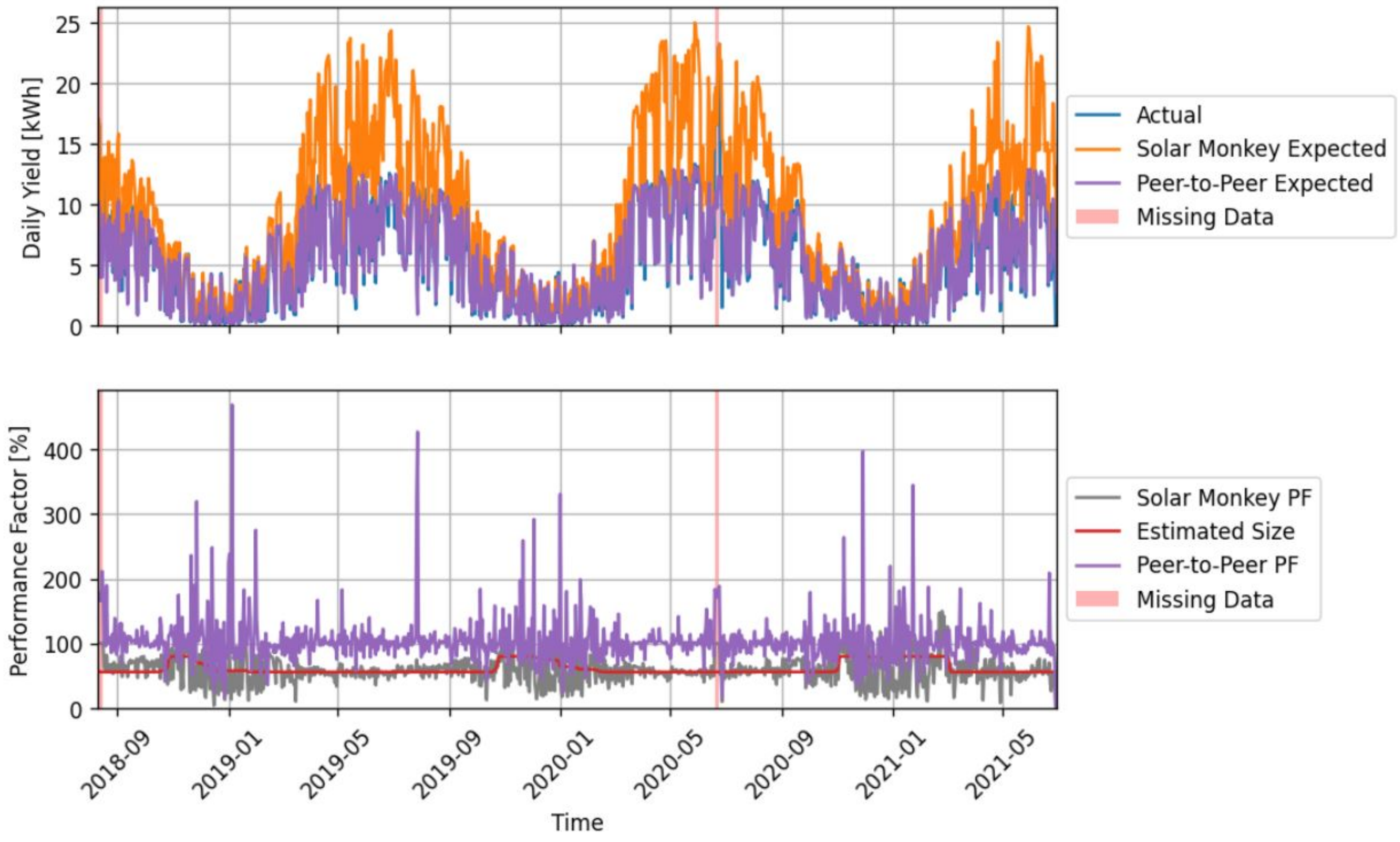


Figure 6.13: Temporal yield plots: false positive

The categorization framework and the examples given are based on the theory, discussions and experience of Operations team. Unfortunately, there was no possibility to validate the conclusions as this would entail physically visiting 120 PV installation sites all over the country or contacting the corresponding home owners. Nonetheless, the objective for undertaking fault detection in this thesis, i.e., to show how the peer-to-peer approach can be used for fault detection, was achieved.

7 Conclusions & Recommendations

After developing a model for monitoring of residential PV systems using neighbouring or peer systems and evaluating the results of the said model, in this chapter, the answers to the research questions posed at the beginning of this thesis are discussed. The section 7.1 concludes the research conducted during this thesis. Next, section 7.2 presents the recommendations and further research suggestions for the project.

7.1. Conclusions

The main objective of this project was to create a model for monitoring and fault detection of a PV system by comparing and using the data of other similar or peer systems. Accordingly, the main research question was divided into five sub-questions. The answers to these questions are provided below.

Q1: What methods presently exist for PV monitoring and which of them use the data from other similar systems?
It was found from the literature that any method for PV monitoring can be classified into four types, namely, physical methods, time-series statistical methods, artificial intelligence methods, and ensemble methods [8], [10]. These types are widely researched for the residential PV sector. The analytical methods work similar to prediction methods used by commercial solar plants. They use the system design information like panel count, tilt, azimuth, etc. along with the weather data at the location of installation to calculate the expected energy yield using physics laws. Next, the statistical and AI methods are based largely on the historical performance of the same system. They use the past energy yields of the same system along with system design information for the predictions. PV power prediction methods can also be used for fault detection if the forecast horizon is decreased to the present or very recent past or very near future. Finally, ensemble methods combine two of the earlier three such that analytical methods are used for pre-processing of data and statistical or AI methods are used for predictions.

The recent developments in 'using similar systems data' methods are based on the comparison of similar systems. These methods often depend only on the energy yield data of the one and similar PV systems for prediction. One method studied in this project proposed a new metric for performance called Performance-to-Peer which uses the energy yield data for finding peers and for prediction [28]. Another method finds neighbouring systems that are geographically close to the one system and have similar system characteristics, namely, the orientation of the system and whether or not the system is inverter limited [26].

Aligned with the goal of this project, a combination of the two methods under the peer-to-peer approach were used as inspiration to develop the proposed model.

Q2: What criteria or what parameters can be used to find similar PV systems?
The first intuitive parameter for finding similar PV systems is the location, i.e., to find systems that are geographically nearby. This was supported by literature as well. Along with this, system design parameters such as panel inclination and orientation, etc. were also considered to filter out very similar systems. However, as postulated, systems which are farther away could also act as good peer systems. Thus, it was found that one of the most important parameters in finding similar PV systems is the historical energy yield data.

Thus, for the proposed model, the geographical location of the PV systems along with the system design data, here panel count, panel inclination, and panel orientation were used. The daily energy yield data for the monitored system and the potential peer systems was also used. From the results of model training, it was found that the daily energy yield data has the highest importance amongst all parameters considered. Peculiarly, the lowest importance was for geographical distance. This outcome was further investigated. The daily energy yield data was further used for predicting the yields for the monitored PV system.

Q3: How can the chosen parameters be used to predict the yields for a PV system?
To answer this question, a model was developed that uses the chosen parameters, location, system data and energy yields, to first find similar or peer systems for the monitored system. These similar systems and their corresponding energy yields were then used by the model to estimate the expected yields for that monitored PV system.

This proposed model uses a genetic algorithm optimization for training the model. In this training, the importance of each of the chosen parameters is calculated. This optimization part of the model was developed as a single general model rather than developing models for all systems individually. This enables the general model to make predictions for any new system soon after installation without waiting for the gathering of large amount of data as long as the new system is within the same region.

The outcome of the optimization, i.e., the chosen parameters according to their importance are then used on individual systems to find good peer systems. The energy yield data of these good peer systems are then used for predicting the energy yields for the monitored PV system. Note that not all good peer systems found have equal importance. The peer system most similar to the monitored system is assigned the highest importance.

Q4: What is the possible way to apply and validate the proposed model?
To answer this question, some experiments were carried out. From these experiments, it was found that for model training, a total of 1700 years of data or 1700 PV systems each with one year of data, is sufficient to ensure the optimized values, i.e., have R^2 score > 0.9. A different combination of number of PV systems and the number of days of data, equivalent to or more than 1700 PV systems each with one year of data, also performs very well. As GA is a random-based search algorithm, a lower amount of data can also be used for training, however, the optimum can not be guaranteed. Must note that a minimum of 6-9 months of data per PV system should be used. This ensures that the model training is not entirely done in winter months which was observed to be unreliable.

The parameter corresponding to the energy yield in model training is time-dependent. As mentioned, this factor is important for model optimization, but it is also necessary post-optimization for finding the best peer systems. Hence, when using the model on a new system, even though model training results are available, a minimum of 30 days of data is needed to find good enough peer systems for the new system with sufficient accuracy and therefore, also for prediction in the immediate future, i.e., the next 30 days. This process should be repeated every 30 days until total data of one year is available. Thus, the model can be used sufficiently on a new system after about 30 days of installation for prediction.

For model retraining frequency, it is important to ensure that the PV systems chosen for model training have a good mix of old and new systems. This is true for all ML based models, where both the training and testing dataset should come from the same distribution. If new systems dominate the training dataset, the algorithm could lead to significantly worse predictions for older systems. Considering this and the results of the experiments done, it was found that the model training saturates at only 25% of the total data available.

Q5: How can the model predictions by this approach be used to monitor and detect faults in a PV system?
The model was designed to automatically estimate the expected yields for a PV system. For the fault detection, a semi-automatic framework was developed. This framework uses the peer-to-peer approach along with other available models for fault detection. The framework enables personnel to estimate when a specific system is under-performing, the magnitude of this under-performance as well as an indicator about the possible cause of the malfunction. The faults considered here are missing data, under-performance, over-performance and false positives.

While this model, based on peer-to-peer approach, cannot be used alone for fault detection yet, it is clear that the model is an important addition to the System Performance Analysis carried out by Solar Monkey. The model is especially useful for separating system design mismatch from actual system malfunctions.

Main research question: Can the accuracy of PV systems monitoring be improved by creating a model that can detect malfunctions of one system by comparing the data of this system with that of neighbouring/peer systems?
A model that can monitor one PV system by comparing its data with the data of other similar or peer systems was successfully developed. This model can be used for detecting malfunctioning systems by using the expected yields and performance factors of the monitored PV system. Data from over 12000 PV systems with some systems as old as seven years was used for this model. Along with several other models available at Solar Monkey, the model developed in this thesis can be used to identify faults in the system.

Along with the research questions, the intended outcomes of the thesis were also achieved. A robust algorithm was developed to find peer systems and to predict energy yields for any monitored PV system. Further, a categorization framework was developed to detect faults in the monitored system.

The proposed model has a mean normalized RMSE of 0.057 and about 95% of the systems tested had an R^2 score higher than 0.85. On the other hand, the existing commercial software at Solar Monkey has a mean normalized RMSE of 0.082 and about 83% of the systems tested had an R^2 score higher than 0.85. Thus, the expected yields calculated by the peer-to-peer approach were found to be more accurate than the other models. However, the model can be further improved as will be discussed in the next section.

7.2. Recommendations

This section presents the recommendations and possible further research that can be done on the topic of peer-to-peer monitoring and fault detection.

System Features: In the proposed model, only panel count, panel inclination, and panel orientation were used amongst the many system design parameters available. This decision was taken to keep the model simple and to limit the computation time required for training within reason. However, it is worth investigating the use of more (such as system age, module technology, etc.) or less of the system design parameters depending on the accuracy of the model.

Geographical Distance: Another suggestion would be to further explore the importance of "geographical distance". While it was concluded that the importance of this distance was low after optimization because daily yield data was used and because Netherlands is a small country with low weather variation, it is possible that "geographical distance" might have higher importance when hourly yield data is used. It is also suggested to examine the optimization results when monitoring systems outside the Netherlands, even if only daily yields are available. Another possibility could be to monitor those PV systems which are in south-west hilly regions of the Netherlands. However, this largely limits the database of potential peer systems.

Distance Training: While the total database available for this project was 12229 PV systems, not all of them could be used due to their quality as well as due to the time taken for distance training. Distance training is the part of model training where the algorithm calculates how similar or dissimilar all potential peer systems are with respect to the monitored system. This is the most time-intensive and memory-intensive part of the code, taking about 10 hours. Thus, during the multiple simulations, the distance training was pre-calculated on 9480 PV systems to save time. Although this means that for any monitored system, the peer systems are found from only these 9480 systems rather than the entire database of 12229 systems. This part of the code needed to be highly optimized to simply use it. However, it is possible and would be necessary to optimize the distance training further when newer systems are added to the database.

Fitness Function: The fitness function used for GA minimizes the average MAE between the expected and actual yields. Fitness functions to minimize the overall MAE, average RMSE and overall RMSE were also tried, which gave slightly different results. In curve fitting functions, often, least squares is used. Thus, it is a recommendation to try other variations of the fitness function to find the best way to define the problem statement.

Other ML Models: The general progression towards the decision to choose genetic algorithm is explained in chapter 3. There is a large database of machine learning based algorithms that can be used with novel algorithms being developed very frequently. There could be some algorithms that perform the optimization faster and better. Within GA as well, there are variations that could not be tried due to the limited time available and could enhance the model.

Retraining: In order to maintain the quality of the results as well as to enable addition of newer systems, the model has to be retrained. Retraining for the proposed model includes recalculation of the "distances" and using them to find good peers for the monitored system periodically. This would be important in situations where a particular system starts degrading and can no longer function as a good peer or if newer and more similar systems are added to the dataset. Retraining helps ML models to learn and improve performance. However, no experiments could be conducted to test the retraining of the optimization algorithm as it is a time-consuming simulation. It is possible that the model also requires hyper-parameter tuning and retraining if the dataset distribution changes significantly. Thus, it is recommended to verify how often the model should be retrained. For the same reasons, cross-validation could not be done on the model training. While the results were verified differently, it should be considered for further work.

Fault Detection Database: It was shown how the model developed in this thesis can be used for fault detection. However, due to the lack of systems with verified faults, it was not possible to validate the diagnosis. A database of systems with their diagnosed faults should be created to induce further confidence in the fault detection process. It was also noticed that as there is no database of system design mismatch problems, the energy yields of these systems or their design remains incorrect. This affects the data quality of the PV systems and consequently, the results of model training. Another suggestion here would be to check the fault categorization framework on more systems. In this thesis, due to time constraints, it was tested on only 120 PV systems and only in July 2021. It would be interesting to verify and further develop it on the bigger database and for winter months. Ultimately, further research could enable an automatic fault detection system with minimal human intervention needed.

Bibliography

[1] NOAA, Carbon Dioxide | Vital Signs – Climate Change: Vital Signs of the Planet, 2019. [Online]. Available: `https://climate.nasa.gov/vital-signs/carbon-dioxide/`.

[2] R. Wadanambi, L. Wandana, K. Chathumini, N. Dassanayake, D. Preethika, and U. S. Arachchige, "The effects of industrialization on climate change,"

[3] G. Van Calster and L. Reins, The Paris Agreement on Climate Change. 2021. DOI: `10.4337/9781788979191`. [Online]. Available: `http://edgar.jrc.ec.europa.eu/overview..`

[4] UNFCC, "Adoption of the Paris Agreement," Tech. Rep. December, 2015, p. 32. [Online]. Available: `http://unfccc.int/resource/docs/2015/cop21/eng/l09r01.pdf`.

[5] A. Smets, K. Jäger, O. Isabella, R. van Swaaij, and M. Zeman, Solar Energy The physics and engineering of photovoltaic conversion, technologies and systems. 2012, vol. 20, pp. 195–230, ISBN: 9783642209505.

[6] SolarPower Europe, "EU Market Outlook for Solar Power 2020-2024," Tech. Rep., 2020, pp. 1–60. [Online]. Available: `www.solarpowereurope.org`.

[7] Solar Monkey - Market leader in solar panel software. [Online]. Available: `https://solarmonkey.io/`.

[8] S. Sobri, S. Koohi-Kamali, and N. A. Rahim, Solar photovoltaic generation forecasting methods: A review, 2018. DOI: `10.1016/j.enconman.2017.11.019`. [Online]. Available: `https://doi.org/10.1016/j.enconman.2017.11.019`.

[9] D. S. Kumar, G. M. Yagli, M. Kashyap, and D. Srinivasan, "Solar irradiance resource and forecasting: a comprehensive review," IET Renewable Power Generation, vol. 14, no. 10, pp. 1641–1656, Jul. 2020, ISSN: 1752-1416. DOI: `10.1049/iet-rpg.2019.1227`.

[10] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-de-Pison, and F. Antonanzas-Torres, Review of photovoltaic power forecasting, 2016. DOI: `10.1016/j.solener.2016.06.069`. [Online]. Available: `http://dx.doi.org/10.1016/j.solener.2016.06.069`.

[11] A. R. Pazikadin, D. Rifai, K. Ali, M. Z. Malik, A. N. Abdalla, and M. A. Faraj, "Solar irradiance measurement instrumentation and power solar generation forecasting based on Artificial Neural Networks (ANN): A review of five years research trend," Science of the Total Environment, vol. 715, p. 136 848, May 2020, ISSN: 18791026. DOI: `10.1016/j.scitotenv.2020.136848`.

[12] D. Grzebyk, "Photovoltaic Yield Nowcasting For Residential Solar Systems In The Netherlands Using A Machine Learning Approach," 2020, p. 129.

[13] M. Bouzerdoum, A. Mellit, and A. Massi Pavan, "A hybrid model (SARIMA-SVM) for short-term power forecasting of a small-scale grid-connected photovoltaic plant," Solar Energy, vol. 98, no. PC, pp. 226–235, Dec. 2013, ISSN: 0038092X. DOI: `10.1016/j.solener.2013.10.002`.

[14] P. Ramsami and V. Oree, "A hybrid method for forecasting the energy output of photovoltaic systems," Energy Conversion and Management, vol. 95, pp. 406–413, May 2015, ISSN: 01968904. DOI: `10.1016/j.enconman.2015.02.052`.

[15] V. Berdugo, C. Chaussin, L. Dubus, G. Hebrail, and V. Leboucher, "Analog method for collaborative very-short-term forecasting of power generation from photovoltaic systems. next gener. data min. summit ubiquitous knowl-edge discov," Energy Manage. Smart Grids Intell. Mach. to-Machine Telemat, 2011.

[16] C. Yang and L. Xie, "A novel arx-based multi-scale spatio-temporal solar power forecast model," in 2012 North American Power Symposium (NAPS), IEEE, 2012, pp. 1–6.

[17] A. G. Vaz, B. Elsinga, W. G. van Sark, and M. C. Brito, "An artificial neural network to assess the impact of neighbouring photovoltaic systems in power forecasting in Utrecht, the Netherlands," Renewable Energy, vol. 85, pp. 631–641, Jan. 2016, ISSN: 18790682. DOI: 10.1016/j.renene.2015.06.061.

[18] F. Mallor, T. León, L. De Boeck, S. Van Gulck, M. Meulders, and B. Van der Meerssche, "A method for detecting malfunctions in PV solar panels based on electricity production monitoring," Solar Energy, vol. 153, pp. 51–63, 2017, ISSN: 0038092X. DOI: 10 . 1016 / j . solener . 2017 . 05 . 014. [Online]. Available: http://dx.doi.org/10.1016/j.solener.2017.05.014.

[19] V. P. Lonij, A. E. Brooks, A. D. Cronin, M. Leuthold, and K. Koch, "Intra-hour forecasts of solar power production using measurements from a network of irradiance sensors," Solar Energy, vol. 97, pp. 58–66, Nov. 2013, ISSN: 0038092X. DOI: 10.1016/j.solener.2013.08.002.

[20] B. Elsinga and W. van Sark, "Spatial power fluctuation correlations in urban rooftop photovoltaic systems," Progress in Photovoltaics: Research and Applications, vol. 23, no. 10, pp. 1390–1397, 2015.

[21] "The Semivariogram," in Geostatistics for Engineers and Earth Scientists, Springer, Boston, MA, 1999, pp. 67–90. DOI: 10.1007/978-1-4615-5001-3_5.

[22] B. Elsinga and W. G. van Sark, "Short-term peer-to-peer solar forecasting in a network of photovoltaic systems," Applied Energy, vol. 206, pp. 1464–1483, 2017.

[23] A. Golnas, J. Bryan, R. Wimbrow, C. Hansen, and S. Voss, "Performance assessment without pyranometers: Predicting energy output based on historical correlation," Conference Record of the IEEE Photovoltaic Specialists Conference, no. June, pp. 002 006–002 010, 2011, ISSN: 01608371. DOI: 10.1109/PVSC.2011.6186347.

[24] O. Tsafarakis, K. Sinapis, and W. G. Van Sark, "PV system performance evaluation by clustering production data to normal and non-normal operation," Energies, vol. 11, no. 4, 2018, ISSN: 19961073. DOI: 10.3390/en11040977. [Online]. Available: www.mdpi.com/journal/energies.

[25] I. Popovic and I. Radovanovic, "Methodology for detection of photovoltaic systems underperformance operation based on the correlation of irradiance estimates of neighboring systems," Journal of Renewable and Sustainable Energy, vol. 10, no. 5, p. 053 701, Sep. 2018, ISSN: 19417012. DOI: 10 . 1063 / 1 . 5042579. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/1.5042579.

[26] R. C. Nijman, "Automatically and real-time identifying malfunctioning PV systems using massive on-line PV yield data," 2018.

[27] J. Leloux, L. Narvarte, A. Luna, and A. Desportes, "Automatic fault detection on bipv systems without solar irradiation data," arXiv preprint arXiv:1410.6946, 2014.

[28] J. Leloux, L. Narvarte, A. Desportes, and D. Trebosc, "Performance to Peers (P2P): A benchmark approach to fault detections applied to photovoltaic system fleets," Solar Energy, vol. 202, pp. 522–539, May 2020, ISSN: 0038092X. DOI: 10.1016/j.solener.2020.03.015.

[29] A. Triki-Lahiani, A. Bennani-Ben Abdelghani, and I. Slama-Belkhodja, Fault detection and monitoring systems for photovoltaic installations: A review, 2018. DOI: 10 . 1016 / j . rser . 2017 . 09 . 101. [Online]. Available: http://dx.doi.org/10.1016/j.rser.2017.09.101.

[30] W. Badr, Why Feature Correlation Matters . . . . A Lot! 2019.

[31] "Machine learning basics," in Machine Reading Comprehension, Elsevier, Jan. 2021, pp. 209–211, ISBN: 978-0-323-90118-5. DOI: 10.1016/b978-0-323-90118-5.00014-x.

[32] A. Ng, Structuring Machine Learning Projects - Coursera. [Online]. Available: https://www.coursera.org/learn/machine-learning-projects.

[33] Scikit-Learn Developers, 3.1. Cross-validation: evaluating estimator performance, 2011. [Online]. Available: https://scikit-learn.org/stable/modules/cross%7B%5C_%7Dvalidation.html.

[34] J. Grus, Data Science from Scratch. 2015, vol. 1542, pp. 33–36, ISBN: 9788578110796. arXiv: arXiv:1011.1669v3.

[35] J. Brownlee, Train-Test Split for Evaluating Machine Learning Algorithms, 2020. [Online]. Available: https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/.

[36] M. Stojiljkovic, "Stochastic Gradient Descent Algorithm With Python and NumPy," Real Python, 2021. [Online]. Available: https://realpython.com/gradient-descent-algorithm-python/.

[37] SciPy v1.6.3 Reference Guide. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.rosen.html%7B%5C#%7Dscipy.optimize.rosen.

[38] S. Raschka, J. Patterson, and C. Nolet, Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence, 2020. DOI: 10.3390/info11040193. arXiv: 2002.04803. [Online]. Available: www.mdpi.com/journal/information.

[39] A. Gad, "Introduction to Optimization with Genetic Algorithm," Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/introduction-to-optimization-with-genetic-algorithm-2f5001d9964b.

[40] S. Panda and N. P. Padhy, "Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design," Applied Soft Computing Journal, vol. 8, no. 4, pp. 1418–1427, Sep. 2008, ISSN: 15684946. DOI: 10.1016/j.asoc.2007.10.009.

[41] A. Abraham, B. Yue, C. Xian, H. Liu, and M. Pant, "Multi-objective peer-to-peer neighbor-selection strategy using genetic algorithm," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4873 LNCS, Springer, Berlin, Heidelberg, Dec. 2007, pp. 443–451, ISBN: 9783540772194. DOI: 10.1007/978-3-540-77220-0_41. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-77220-0%7B%5C_%7D41.

[42] L. Smith, Genetic Algorithms - Generative Design Primer. 2020. [Online]. Available: https://www.generativedesign.org/02-deeper-dive/02-04%7B%5C_%7Dgenetic-algorithms.

[43] V. Mallawaarachchi, "Introduction to Genetic Algorithms," Towards Data Science, p. 1, 2017. [Online]. Available: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3.

[44] Tutorialspoint, Genetic Algorithms Tutorial. 2016, p. 36. [Online]. Available: https://www.tutorialspoint.com/genetic%7B%5C_%7Dalgorithms/index.htm.

[45] D. J. Toal, N. W. Bressloff, and A. J. Keane, "Kriging hyperparameter tuning strategies," AIAA journal, vol. 46, no. 5, pp. 1240–1252, 2008.

[46] C. Pascoal, Tutorial: Understanding Linear Regression and Regression Error Metrics, 2019. [Online]. Available: https://www.dataquest.io/blog/understanding-regression-error-metrics/.

[47] Minitab, "Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?" Regression Analysis, pp. 1–6, 2013. [Online]. Available: https://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit.

[48] S. Glen, R-Squared in One Picture - Data Science Central, 2019. [Online]. Available: https://www.datasciencecentral.com/profiles/blogs/r-squared-in-one-picture.

[49] Snow Y., "A Better Analytical Model," Solar Monkey BV, Tech. Rep., 2020.

[50] D. L. King, W. E. Boyson, and J. A. Kratochvil, "Photovoltaic array performance model," Sandia Report No. 2004-3535, vol. 8, pp. 1–19, 2004. DOI: 10.2172/919131. [Online]. Available: http://www.osti.gov/bridge/product.biblio.jsp?osti%7B%5C_%7Did=919131.

[51] T. Huld, G. Friesen, A. Skoczek, R. P. Kenny, T. Sample, M. Field, and E. D. Dunlop, "A power-rating model for crystalline silicon PV modules," Solar Energy Materials and Solar Cells, vol. 95, no. 12, pp. 3359–3369, Dec. 2011, ISSN: 09270248. DOI: 10.1016/j.solmat.2011.07.026.

[52] Scikit-Learn Developers, StandardScaler, 2021. [Online]. Available: `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html`.

[53] G. A. Korn and T. M. Korn, Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review, 76. Dover Publications, 1961, vol. 15, p. 421, ISBN: 9780486411477. DOI: `10.2307/2003035`.

[54] J. Bhattacharjee, Basics Of Statistics For Machine Learning Engineers. [Online]. Available: `https://medium.com/technology-nineleaps/basics-of-statistics-for-machine-learning-engineers-bf2887ac716c`.

[55] Google Maps, 2008. DOI: `10.1007/978-1-4302-0637-8_8`. [Online]. Available: `https://www.google.nl/maps/place/Netherlands/`.

[56] Y. V. Montfort, “Automated tagging of corrupt data of daily yield from residential PV systems,” 2020.