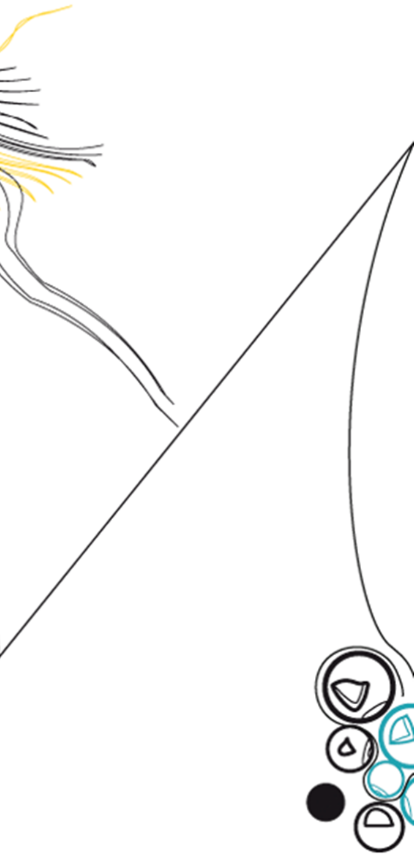
UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Dependable Probabilistic Energy Forecasting of Solar Energy for Energy Management Systems

Cas Jesse Doornkamp

M.Sc. Thesis
December, 2023



Supervisors:
dr. ir. M. E. T. Gerards
dr. ir. G. Hoogsteen
ir. I. A. M. Varenhorst
dr. ing. Y. Huang

CAES
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
Drienerlolaan 5
7522 NB Enschede
The Netherlands

ABSTRACT

In this thesis, a methodology has been researched and developed to aid in the decision-making process of Energy Management Systems (EMS). Current solar energy forecasting methods forecast the expected energy production of solar panels at certain points in the future, called point forecasts. However, due to a lack of information about the environment the solar panels will be in, caused by the unpredictable nature of the weather, point forecasts cannot fully describe the future.

Therefore, a different modelling method has been researched and developed that takes this uncertainty into account and provides additional information to the energy management system about this uncertainty. This is done by forecasting a probability distribution of the expected solar energy generation of the PV installation. Probability distributions describe possible future outcomes and their likelihood, instead of point forecasts which represent one possible future. Given the different nature of probabilistic forecasts, the traditional methodology and data used for point forecasts need not apply to probabilistic forecasts. Therefore, new forecasting methods and data sources have been investigated that are best suited for probabilistic solar power forecasting, without it being based on assumptions on point forecasts.

The models designed in this thesis have been analysed to see how performance and reliability is influenced by weather conditions and how model specific features are related to this behaviour. The models were implemented and tested on an embedded device to verify the feasibility of the methodology for real-life applications. Additionally, different use cases and methods are provided in this thesis on how these probabilistic forecasts can be interpreted to enhance an EMS' operation and how it allows for unique applications that are not possible using point forecasts. Based on the findings in this thesis, new research directions to improve EMS robustness can be identified.

ACKNOWLEDGEMENT

A lot of time has passed since I first started working on this thesis. During that time, many things have changed, and I personally have changed as well. This thesis has been an arduous process for me, but I have had the pleasure to have been in such good company during this time and I have enjoyed the support of good people. Because of that, I would like to take the time to thank some of them here personally.

First, I would like to thank Gerwin Hoogsteen and Bart Nijenhuis, who were my daily supervisors when I first started with my thesis. Because of the great advice and mentorship they provided during this time, I have had the opportunity and confidence to put my thesis on hold and take the time necessary to figure out personal matters first. I cannot thank them enough for supporting me and setting me on that path.

When I was ready to continue with my thesis again, I was welcomed by Gerwin Hoogsteen and Ivo Varenhorst as my new daily supervisors. During the continuation of my thesis, I could always count on their aid and ideas. Although most of the meetings we had were taken up by updates on the multitude of figures and experiments I had done in between meetings, you both always had an eye on the bigger picture and kept my flighty pursuits to a healthy amount and made sure I set realistic goals for this master thesis.

I would like to thank Marco Gerards and Yanqiu Huang for their helpful feedback, support, and ideas during this thesis.

I would also like to thank the Energy Group and the students that I worked alongside with. From the Friday meetings to the coffee breaks or lunch walks, I enjoyed the warm and friendly atmosphere that was always present.

Lastly, I would like to thank my family for supporting me during these times and for being there for me outside of the thesis as well. I would not have been able to reach this point in my studies without your unwavering support, and I cannot thank you enough for this.

Contents

ACRONYMS

(A)NN (Artificial) Neural Networks.

AE Autoencoder.

AI Artificial Intelligence.

BNI Beam Normal Irradiance.

CAMS Copernicus Atmosphere Monitoring Service.

CDF Cumulative Distribution Function.

CNN Convolutional Neural Networks.

CRPS Continuous Ranked Probability Score.

DHI Diffuse Horizontal Irradiance.

DNI Direct Normal Irradiance.

DSM Demand Side Management.

ECMWF European Centre for Medium-Range Weather Forecasts.

EMS Energy Management System.

ENS Ensemble.

EUMETSAT European Organisation for the Exploitation of Meteorological Satellites.

EV Electric Vehicle.

GHI Global Horizontal Irradiance.

KNMI Koninklijk Nederlands Meteorologisch Instituut.

LSTM Long Short-Term Memory.

MAE Mean Absolute Error.

MAPE Mean Absolute Percentage Error.

MBE Mean Bias Error.

MCQRNN Monotone Composite Quantile Regression Neural Network.

MSE Mean Square Error.

MSG Meteosat Second Generation.

NWP Numerical Weather Prediction.

PDF Probability Distribution Function.

POA Plane Of Array.

PV Photovoltaic.

QL Quantile Loss.

QNN Quantile Neural Networks.

QR Quantile Regression.

QRF Quantile Random Forests.

RF Random Forests.

RH Relative Humidity.

RMSE Root Mean Square Error.

RNN Recurrent Neural Networks.

SEVIRI Spinning Enhanced Visible and InfraRed Image.

SG Smart Grids.

SVM Support Vector Machine.

TOA Top Of Atmosphere.

CHAPTER 1

INTRODUCTION & PROBLEM STATEMENT

In the last few years, a rising trend is observed in the energy market towards sustainability, as is concluded by the International Energy Agency in their World Energy Outlook 2023 report [1]. They found that the global investment into clean energy has risen by 40% between 2020 and 2022. One of the key players in this transition is solar power, of which a total of 220GW capacity worldwide was installed in 2022 alone. This market has seen many positive developments in their manufacturing cost, energy efficiency, and durability. These key factors made choosing solar panels more attractive and as a result the solar market has been gaining popularity.

There are however issues that still need to be addressed in order to make solar power a better fit for the energy market. Current fossil based power plants have the advantage of being predictable and adjustable in their power output. This is in contrast to solar and wind power, whose power generation depends on external factors and are as such not always a good fit for new power plants. The main problem is that solar power, like wind power, is dependent on the weather around its location. The evolution of weather over time is however known to be difficult to predict, two big issues here are the approximation of the actual behaviour of the weather and the uncertainty in current conditions [2]. These predictions always have a level of uncertainty and as a result make predicting the expected solar power output difficult as well. This uncertainty is a problem for the stability of the electricity grid [3]. Mainly because fluctuations in the production give rise to degradation of the distribution infrastructure and an imbalance to the supply and demand of electricity can be costly for grid operators to rectify. Therefore, a more active type of control is needed to reduce these imbalances and fluctuations, Smart Grids can fulfil this need.

1.1 Smart Grids

With the growing need for more robust energy grids and increased energy efficiency, many improvements have been made to the systems that control how energy is used. Stimulated by the rising contribution of renewable energy, smarter and more reactive systems have been developed to tackle the problems caused by the intermittent and unpredictable power production. These digital systems span from simple solar power prediction applications to various other applications in the energy sector, these systems are as a whole called Smart Grids (SG) [4].

SG is a broad term and applies to numerous applications, this however does not really give an indication of the variety and possibilities that they enable. In general, they are digital systems that measure, monitor, and manage energy usage of interconnected components connected to the same energy grid. These networks can actively collaborate, making the grid more resilient to disruptions and optimize grid operation to ensure each user receives an efficient and reliable service. Smart grids can range from small to large energy girds, or consist of a many small grids called microgrids that together form a larger grid. Another facet of SG is Demand Side

Management (DSM) [5] that, with the use of Energy Management Systems (EMS), aim to steer energy usage, for example to reduce peaks in energy consumption, or to reschedule energy consumption when energy is abundant. By doing this, operational costs can be reduced and the stability of the energy grid can be improved. Steering energy usage can be done dynamically by controlling energy prices so users who are able to reschedule are incentivised to do so, or are compensated if they cooperate.

Energy Management Systems (EMS) are the systems which provide the services and functionality needed to manage or control the Smart Grids. On a smaller scale and in a similar vein to microgrids, an EMS can be used to manage a few locally interconnected devices. This could be for example a solar park with battery storage, or a car park with Electric Vehicle (EV) chargers [6]. For an EMS scheduling the charging of EVs, which can use locally generated power from solar panels, the influence of inaccurate solar energy forecasts can be more significant on this smaller scale, as is described in [3]. In [7] such an algorithm is proposed for scheduling EV charging by using the available headroom between the base load and a maximum charging power variable called the *fill-level*. This *fill-level* variable is set according to the amount of energy requested to charge the vehicle within the available time and the availability of energy. This methodology produces accurate results in most cases, except where the electric vehicles are charged with PV power. The algorithm is less effective for cases with a smaller energy request due to the variability in available solar energy and low energy requirements. The research done in [8] builds on top of this algorithm by combining solar power forecasts with the power availability from the grid, but also notes that the actual charging profile deviates from the planning due to solar power forecast errors.

1.2 Problem Statement

The aim of this thesis is to provide more robust and reliable solar energy forecasts for energy management systems, such that issues caused by inaccurate forecasts can be mitigated or taken into account. To do this, a hypothetical EMS example is introduced in this thesis to function as a reference point when design decisions need to be made, it also aids as a demonstrative tool to indicate how each part fits into the bigger picture. This EMS is based on an EMS used by the University of Twente for research purposes called SlimPark [9]. The SlimPark site is a car park located on the University's campus, which is partially covered by a total of 27 kWp solar panel array which are connected to 9 EV chargers, a 30 kWh battery, and the local power grid [6]. The demonstrative EMS application in this thesis would be responsible for charging electrical vehicles using as much local solar energy as possible, with the goal to provide a reliable and robust service. The definition and the control algorithms of the EMS itself is outside the scope of this thesis, the focus will remain on the forecasting part of the system.

In Figure 1.1, the evolution of power generated by the SlimPark solar panels is shown for four different days to give an idea of how much the generated power by solar panels can vary between days, as well as within a day. Predicting the behaviour of the power is difficult due to the imperfect information known about the future, as well as the many facets that influence the power output of solar panels. The research field has seen many endeavours to improve the quality of these forecasts. Most research focusses on accuracy to achieve better quality forecasts, but forecast accuracy does not directly imply better performance by the systems that use these forecasts. This is concluded by [10], which promotes choosing and optimising forecasting models based on the consequences of forecast errors alongside overall accuracy of the forecast.

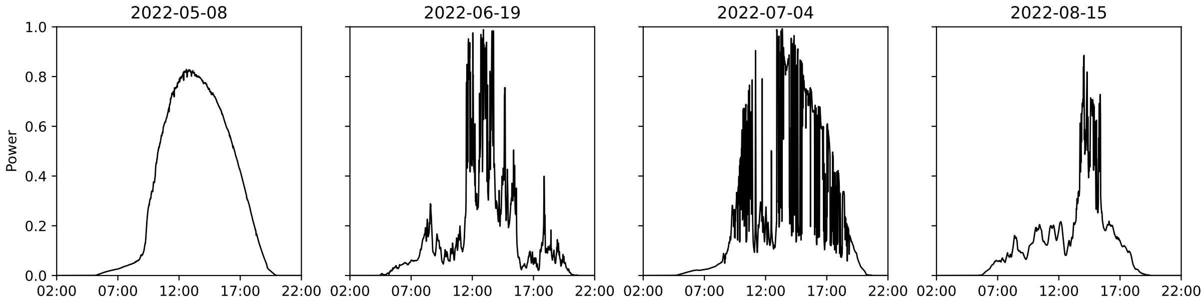


Figure 1.1: Generated power by solar panels on four different days at the SlimPark car park.

Many energy management systems currently use point forecast models to tackle the issues caused by the ever-changing weather on PV power generation. Point forecasts predict the power generated by solar panels, which an EMS uses to base its control decisions on. Point forecasts are however limited in the information they can provide, as only the expected value of the future power production is forecasted by the model. These forecast models are not perfect and might differ from reality. These inaccuracies in the forecasts can lead to an EMS making a bad decision, which can have consequences for the service that these systems provide. If an EMS could have more information about what might happen in the future and how likely that is to happen, it could make better decisions that in turn makes the system more robust to these anomalies.

Recently, new wind forecast methodologies are preferring probabilistic forecasts to point forecasts as a way to tackle this issue [11]. Probabilistic forecasts give insight into the uncertainty of the future power generation, by estimating the spread of values the future power can take and how likely each value is to happen. In 2014 the Global Energy Forecasting Competition was held, a competition to forecast load, price, and solar and wind energy. The results of this competition, together with a small survey of the state of probabilistic forecasting in this field, have been bundled in [12]. For probabilistic wind power forecasting, the results did show good maturity, attested to the fact that wind power is closer to meteorological forecasting where probabilistic forecasting is well-established. However, the probabilistic forecasting of solar energy was deemed not mature enough at the time. A more recent survey [13] however, has shown an increased interest and improvements in probabilistic forecasting of solar energy due to recent advancements in artificial intelligence.

The main theme in probabilistic forecasting of solar energy has been to increase its forecasting accuracy [13]. Less attention has been given to practicality and feasibility of these forecasts as most studies kept their contributions as theoretical models without implementation in mind, as was noted by [10]. Therefore, more research needs to be done to see how these forecasts would fit in real-life EMS applications and how their requirements or behaviour would differ with point forecasts.

1.3 Research Questions

Weather is inherently unpredictable and as such the weather forecasts also have a degree of uncertainty. This level of uncertainty is however not always represented when looking at the results. The output of these forecasts are the most likely values to happen in the near future. But these situations do not always happen, and the predictions can sometimes be off by a significant amount. This makes it difficult to have solar energy as a predictable energy source. In order to reduce the errors of these predictions, an effort can be made to improve the accuracy of the solar energy production. Although this would be the best solution, it might be very difficult or even unachievable given spatial, temporal or computational constraints. But instead of trying to predict the correct future, it might be enough to provide information of how likely that prediction is to happen or the range of values the future might be in. Thus, instead of focusing on accurate predictions, a focus on informing the user of the uncertainty in each forecast might be enough. This would allow an EMS to be aware of the possible futures and thus make strategic choices, minimising possible risks and thus making these systems more robust to this ambiguity. This overall objective is best represented by the following main research question:

"How can the reliability of energy prediction of solar panels be improved?"

This question would, on its own, be difficult to answer and depends on the context. As such, some accompanying questions are made to guide the research and give scope to the directions that are explored.

Question 1: What data can have a meaningful contribution to predicting the expected solar output power of solar panels and its uncertainty?
To answer this question, research has to be done on what data is available and can be used in real time as well. These types of data can then be tested to see if it shows correlation with the solar output. And thus give an answer to this question. It is possible that the model used to predict solar power might have an influence on the actual usefulness of the data, as such the outcome of this question should not be regarded as a hard truth, but as a guide for selecting data that should be tested with in this thesis.

Question 2: What methods can be used to make these predictions more reliable?
To answer this question, a list of possible methods must be researched that can help achieve the desired results. Attention should be taken to how these models can provide additional data about the end result, as they don't have to produce the same metrics.

Question 3: How can the reliability of the predictions be assessed?
For this question, first the performance metrics found in the literature study are used to find a common ground on what reliability entails and how it can be quantified. The found models should then be evaluated using these metrics to find possible drawbacks these models might have. This can be done for example by correlating the models' accuracy with potential states the environment can be in.

Question 4: Are the found models practical and resource efficient solutions for real-world applications implemented on an embedded device?
Embedded devices have limited resources available to make these types of predictions with. Therefore, the models made in this thesis should also adhere to these architectural constraints. Otherwise, the models would perform well in theory, but have no practical use in a real world application. There may be trade-offs necessary to make the models work on an embedded device, hence the size and computational complexity of the models should be taken into account during the selection process.

1.4 Thesis Outline

- **Background:** In this chapter, the necessary background information is given on the current state of solar energy prediction, as well as the core concepts the thesis builds upon. In Section 2.1 the different types of solar energy predictions are discussed. Then, in Section 2.2 the different types of data that are commonly used for these predictions are listed. In Section 2.3 the noteworthy methodologies that are used to predict solar energy are discussed. Alongside this, in Section 2.4 an introduction into probability is given explaining the necessary concepts of probability used in this thesis and how probability can be used for predicting solar energy.

- **Methodology:** Chapter 3 describes the methodology used for creating the forecasting models, together with the different data sources that are used in this thesis. In Section 3.2 the description of the overall model is given alongside explanations of its subcomponents and how they are connected as a whole. Each component has their design and parameters discussed separately in their own sections.
- **Experiments & Results:** This chapter explores different model configurations based on the parameters discussed in Section 3.2 and selects the best performing model configuration for further analysis. First, the test setup is described in Section 4.1, after which the experiments exploring the different model configurations are discussed in Section 4.2. The final models are then compared in Section 4.3 with a benchmark model to assess the reliability of the found models, these models are tested based on overall performance, data dependencies, and embedded characteristics. From these results, the practical application of these forecasts in energy management systems are discussed in Section 4.4. At the end of the chapter, the results are discussed and summarized in Section 4.5.
- **Conclusion:** Lastly, in the conclusion, the research questions are answered in this chapter to wrap up the thesis along with recommendations for future work.

CHAPTER 2

BACKGROUND

In this section, more information is given on the different types of solar energy forecasting, how they are impacted by environmental factors, and the current state of PV-power forecasting methodology. The information discussed here is supported by three survey papers that focused on this field of research. The first survey [11] focused on the different deep learning methods used to predict solar and wind energy and what data is used to make these predictions. The second survey [14] focused on the data-mining methods used in this field. The third survey [13] is aimed at probabilistic forecasting of solar energy and energy consumption, showing similarities and gaps in research and providing individual discussions of the results in the studies. This last survey is a recommended read for a wider and more in depth background on solar energy prediction. In addition to these surveys, more specific example studies related to answering the research questions were explored. The results of this research are divided into the relevant subjects for this thesis.

2.1 Types of Predictions

Current endeavours in the field of energy prediction focus on a wide range of applications. For example, accurate solar power generation forecasts for better scheduling in energy markets [15], or integrating PV in smart grids [16, 17, 18]. The different types of predictions specific to solar energy used in literature can broadly be described by three, and often related, characteristics. They consist of the type of energy it predicts, the spatial, and the temporal characteristics of the prediction.

1. **Energy type** In literature, there are two types of energy predicted by models. The first predicts the amount of solar energy radiated onto an area, as is done in [19, 20, 21, 18]. The second type predicts the electrical energy generated by the solar panel [22, 23]. The models predicting the electrical energy often use the output of solar energy prediction models as an input for their own predictions. Some studies present models for both types, as is done in [15].

2. **Spatial** The spatial horizon and resolution characteristic is the scale and detail for which the prediction is meant. Smaller scale prediction models are more common for single PV installations, which can be optimised for their specific environment. This characteristic is more prevalent in studies predicting the output power, as local variables can influence the power production [24, 25]. Larger scale predictions predict for single or multiple areas and are more common in solar irradiance prediction [21, 19, 20] as it is more universally applicable and varies less over larger areas compared to the power production of individual solar panels.

3. **Temporal** The temporal horizon and resolution characteristics describes the amount of time the complete prediction spans and the time between consecutive samples. This can be for the intraday forecasts with a resolution in the minute or hour range [26, 27, 18, 23, 28], or for day-ahead predictions [15, 21, 29] which span one or more days in the future, often with a one-hour resolution [13]. Several studies [30, 19, 27] state that

models predicting up to 4 hours in the future generally perform better when using past measurements and observations, while 6+ hour predictions using data from Numerical Weather Prediction (NWP) models give better results.

2.2 Data

Each application has its own requirements and as such its own set of data it relies on. The literature reviewed for this thesis select their data from a wide assortment of sources and parameters. These parameter selections can differ significantly from study to study, as is evident from the three surveys [11, 14, 13] where the datasets and application are listed per study. These choices are commonly based on the availability of the data, the application, and prediction requirements, its hypothesized significance to the output, or are determined by a correlation analysis or heuristic. In this section, first an overview of these types of data and their uses are given. After which a summary of possible sources for this data is listed and discussed. Lastly, in the conclusion, a summary is given relating the research to the application of this thesis.

2.2.1 Types of Data

The two most prominent types of data used in PV-power forecasting are past power observations and meteorological variables that describe the weather in the area. There are however many possible parameters used to describe the weather, but not all of them are relevant to PV-power generation. As such, the most commonly used meteorological parameters as found in [13] are discussed here.

2.2.1.1 Meteorology

The influence of weather on the performance of solar panels is a well acknowledged concept in solar energy prediction. As such, the inclusion of meteorological variables is very common in these studies. Meteorological variables describe the atmospheric properties that drive weather processes around the world. These variables such as temperature, humidity, cloud cover, and solar irradiance can have a direct impact on the power output of solar panels. Other meteorological variables also have an impact, although indirectly or help give a better impression on how the future weather will evolve over time, such as air pressure and wind speed impacting the movement of clouds for example. Here, the most commonly used meteorological variables used in literature are explained by group and how they are used in practice.

Generic Parameters

- **Temperature** is derived for both the surface and air. The air temperature ($°C$ or K) is often given at different pressure levels or altitudes in the atmosphere, the most commonly used being the air temperature at a height of 2m. The temperature directly impacts the electrical power generated by PV-panels, as is discussed in [24] where an increase in temperature negatively impacts the produced power of a solar panel. But as noted in [25] and [24], the ambient temperature can differ significantly from the PV panel temperature. Both conclude the PV panel temperature is more closely correlated to the power production, but temperature sensors are not always present at every PV installation.
- **Air pressure** or atmospheric (surface) pressure comes from the weight of the air above pressing down. As a more general description, pressure is the force applied to an area ($1N/m^2 = 1Pa$). The atmospheric pressure given in weather reports is expressed as if

measured at the average sea level, not at the actual surface altitude of the region. The latter is also used in models and is called the atmospheric surface pressure. It is however not always clear which version is used in studies when air pressure is mentioned.

- **Wind speed** is expressed at multiple pressure or altitude levels, with 10m elevation being the most common. There are two common ways the wind speed is expressed. Either as UV components where the wind speed is considered from two orthogonal directions, for example North/South and East/West each have their own wind speed(m/s). The other is the wind speed(m/s) relative to the surface of the earth and the direction it is going relative to north (°).
- **Total Column Water** is the amount of liquid water, ice, rain, water vapour and snow present in an imaginary vertical column expressed as (kg/m^2). Each component listed here can also have its own separate variable, as each component can behave differently. In the case of solar radiation, each component has different absorption and scattering characteristics altering the sunlight passing through the volume, which in turn influences the path and intensity of sunlight reaching the surface depending on the amount of water present in the column.
- **Relative Humidity (RH)** is the concentration of water vapour pressure in the air compared to the saturation pressure point of air. At 100% relative humidity, when the air is fully saturated, water vapour will turn through condensation into water droplets. Or when it is cold enough and at low enough pressure, turn water vapour into ice through deposition. This variable indicates possible accumulation of water on the PV panels through condensation, which refract and scatter light and influences the productivity of the PV panel. This saturation pressure point depends on the temperature and pressure of the air, colder air can hold less water vapour while warmer air can hold more. Its inverse, the dew point, is the temperature at which the air at its current water vapour concentration and pressure would reach 100% relative humidity and saturate. This measure is more akin to how we would experience “dry” or “humid” air. At higher altitudes, both the air pressure and temperature are lower, which reduces the saturation point and in the right circumstances allows condensation or deposition to form clouds.

Clouds have multiple characterizations in weather descriptions that try to capture its complex nature while giving useful and quantifiable measures that apply on a more macro scale. Common examples used in solar energy prediction are:

- **(Total) Cloud Cover** is a value between (0-1) which describes the amount of cloud cover over a certain region relative to the size of the region. For total cloud cover this spans the entire vertical height of the atmosphere, but can also be given for separate levels in pressure. A weighted version of cloud cover exists where the weight is the amount of light let through by the cloud, this gives an indication of how much light is blocked by the cloud as seen from the surface.
- **Cloud height** expressed in meters (m) gives the top of the cloud in that region. Its inverse, cloud base height, gives the lowest altitude a cloud can be found, excluding fog.
- **Total Column Cloud** is very similar to the previously mentioned Total Column Water. But only the liquid water and ice water, excluding rain and snow, that are part of the clouds in the same vertical column are estimated. Rain and snow would fall under Precipitation.
- **Precipitation** is the amount of liquid or frozen water that falls on the earths surface within a certain time frame and area. Its unit is commonly a height where the volume spread evenly over a unit area has one dimension remaining $m^3/m^2 = m$, but it can also be

expressed with respect to its mass as (kg/m^2). The precipitation is given for two distinct types based on the atmospheric conditions at that time, called large-scale and convective precipitation. Large-scale precipitation releases its water more evenly over time, whereas convective precipitation can release large amounts of water in a short amounts of time but nothing outside of that. As the precipitation describes the accumulated amount, this behaviour would not be evident when looking at the combined value alone.

Solar Irradiance is a meteorological variable used in most studies on solar energy prediction. It describes the amount of electromagnetic energy radiated by the sun and going through or hitting a unit of surface per unit of time $(J/(m^2 \cdot s))$, or in the more frequently used form (W/m^2). Solar Radiation on the other hand would refer to the total amount of solar energy received by that surface accumulated over time (J/m^2) and should not be confused. Solar Irradiance is used as an input for PV modelling, or as an output to predict solar irradiance for areas where sensors are not necessarily available. It comes in multiple forms:

- **Direct Normal Irradiance (DNI)** describes the radiation hitting a surface on earth which is directly facing the sun. This variable is also called Beam Normal Irradiance in some cases, or Beam Horizontal Irradiance when measuring irradiance relative to a horizontal surface.
- **Diffuse Horizontal Irradiance (DHI)** describes the radiation hitting a horizontal surface at ground level from all directions, excluding the direction of the sun. This radiation is generalised to be uniform for all other angles, as the term diffuse implies.
- **Global Horizontal Irradiance (GHI)** is the total amount of radiation from the sun hitting a horizontal surface at ground level and is the weighted sum of DNI and DHI:

  $$GHI = DHI + DNI * cos(zenith)$$

  Where $zenith$ is the angle between the direction of the sun and the direction orthogonal to the horizontal surface. The cosine factor rectifies the contribution of DNI to match what the direct irradiance would be if measured on a horizontal plane.
- **Plane Of Array (POA)** is the total amount of radiation from the sun hitting a tilted surface at ground level. This is most similar to a PV panel installation. This can directly be measured by sensors or approximated from the previous variables. A simple and common model for approximating the POA is the sum of direct radiation, reflected radiation, and radiation from the atmosphere. It is formulated as follows:

  $$\begin{aligned} POA &= direct + reflected + diffuse \\ &= DNI * cos(AOI) + GHI * albedo * \frac{1 - cos(tilt)}{2} + DHI * \frac{1 + cos(tilt)}{2} \end{aligned}$$

  Here AOI is the angle of incidence between the direction of the sun and the direction the surface is facing, $tilt$ is the angle of incidence between the surface and the horizontal ground, $albedo$ is the reflection coefficient of the ground, i.e. the ratio between the radiation hitting the ground and the amount of radiation reflected off the ground. The DNI, DHI and GHI variables give more generic descriptions of the solar irradiance compared to POA. This is because POA depends on a specific tilt and albedo of the environment for it to work, which is more akin to a PV installation. Hence, when predicting the expected power production of PV panels, the POA can be used as a method to reduce the complexity of the model by reducing the number of variables from 3 to 1 and embedding the PV configuration in the data. Evidence for this is given in [25] where the measured POA shows a higher correlation with respect to the output power when compared to the measured GHI.

- **Top-Of-Atmosphere (TOA)** refers to solar irradiation as if measured at the top of the atmosphere. This variable represents the amount of sunlight reaching the earth before it is influenced by the atmosphere and can be a starting point to determine the DNI, DHI, or GHI from. It can also be used to estimate how much solar energy would reach earth as if it is a cloudless sky.

- **Clear Sky Irradiance** models the same variables as the DNI, GHI, and DHI for example. It uses the same atmospheric conditions as the normal counterpart, but assumes a cloudless sky in its model. This value is more stable over time compared to its counterpart, as clouds can cause sharp fluctuations in the received irradiance within minutes [3]. As it ignores the influence from clouds, the impact of clouds on the irradiance can then be modelled independently. As clouds attenuate the amount of sunlight by either absorbing or scattering the light away, the Clear Sky Irradiance will generally follow the upper bound of the irradiance that could be received at that point in time. In certain conditions, clouds can scatter sunlight on the same surface as direct sunlight that is not blocked by clouds, increasing the total irradiance received compared to the same situation with a clear sky. This situation is most noticeable in short time spans and small areas where clouds are just about to cover or just passed over the solar panel. For larger time intervals and areas this would not show up as the event only lasts a few minutes and would have been averaged out. It is important to note that the Clear Sky Irradiance is modelled and cannot be measured, unless it is a cloudless day, so the quality of the model is important and should be taken into account when used.

- **Clear Sky Index** is calculated as the ratio between the actual DNI/GHI/DHI received at the ground level and the Clear Sky Irradiance counterpart. It represents how much clouds are interfering with the irradiance at the ground. Normally the ratio is in the (0-1) range where 0 means fully occluded and 1 means clear sky, but as discussed before can also produce peeks exceeding 1 in the right situations. In [28] the use of the Clear Sky Index is discussed and instances are shown where the use of the Clear Sky Index is discouraged in PV modelling. Although the Clear Sky Index does produce values closer to the measurements in clear sky settings, the downside is that the uncertainty of the ratio is amplified around sunrise and sunset where the Clear Sky Irradiance is small or even 0 at night. And as discussed before, the Clear Sky Index will show large peaks and fluctuations for small areas and time intervals when it's cloudy, making it more useful when predicting solar irradiation for larger areas and time intervals. The Clear Sky Index is used in [20] as an analysis tool to show the relative performance of these model for different clear sky conditions to see how well they handle varying cloud coverage.

- **Clearness Index** is very similar to the Clear Sky Index, but here it is the ratio between the solar irradiance received at the ground relative to the irradiance at the top of the atmosphere. It models the total influence of the atmosphere on the irradiance on the ground. The Clearness Index generally has lower values compared to the Clear Sky Index as in most cases the atmosphere attenuates the received irradiance, which was already taken into account for the Clear Sky Index.

2.2.2 Data Sources

There are a few different sources where both the meteorological variables and PV-power data can be obtained from. Each source has its advantages, disadvantages, and spatial and temporal characteristics. Each source is discussed in their own section.

2.2.2.1 Weather Stations

Weather stations are a reliable source of local meteorological variables with frequent and accurate measurements. Their downside however is that the distance of these weather stations relative to the point of interest influences how significantly their readings are correlated to the point of interest as is shown in [15]. But these readings are still used in [20, 21, 18, 27] to predict the solar irradiance. Or as is used in [15] the measurements are used to correct for biases in solar irradiance forecasts with the actual observations improving performance. These weather stations are often managed by national weather institutes such as the KNMI [31] managing the weather stations in the Netherlands.

2.2.2.2 Radar and Satellite

As weather stations cannot be placed anywhere due to practical and economical constraints, the gaps in measurements can be filled by satellite imagery and radar systems which observe these meteorological phenomena from a distance. The most natural and well known example of such an observation is cloud cover, where either a satellite's or radar's reading gets transformed into cloud density based on the changed scattering of light due to clouds. Another purpose of these satellites or radars is to predict a cloud's density and their likelihood to produce rain. With their high spatial and temporal resolution covering large areas, their data are useful for near future predictions, as is stated by the survey paper [13]. These readings are also directly used in [32] for predicting future GHI based on a local sky imager and lagged GHI readings. A review paper [33] which covers the use of satellite imagery independently recommends persistence models near the minute prediction range, where it outperforms statistical methods and neural networks.

A satellite specifically employed for this purpose is the SEVIRI satellite, shorthand for Spinning Enhanced Visible and InfraRed Imager. It has 12 channels to measure parts of both the infrared and visible light spectrum with 3 km or 1.5 km spatial resolution, providing information about both temperature and atmospheric composition. The data from this satellite is used by [19] to provide live irradiance forecasting based on cloud movement, it outperforms standard weather forecasts in the first 2–3 hours and also outperforms the smart persistence benchmark model in the first hour. These readings together with weather stations form the basis of the CAMS solar radiation service, a reevaluation of the irradiance in the past made by the Copernicus Atmosphere Monitoring Service.

2.2.2.3 Numerical Weather Predictions

The most common source of data are from Numerical Weather Predictions (NWP), they predict how the weather will evolve over time. These predictions are made available by different meteorological institutes around the world. These predictions are based on data from weather stations and satellite data around the world. Each institute has its own focus and/or speciality, this difference can be based on the region it operates in or its purpose, like predicting naval tides or rainfall. The type of forecasts mostly dictates the temporal and spatial resolution of the weather prediction, as well as how far into the future the prediction goes. Therefore, these forecasts can be obtained from different institutes around the world and are available in different prediction formats, for example a 14-day ahead weather prediction with a per day statistical description or a day ahead prediction with variables predicted for each time step into the future. ECMWF [34] is one of such institutes focused on Europe and provides multiple datasets of its past and present forecasts to be used in research. Often, near future predictions can be made with higher temporal and spatial resolutions and overall accuracy, but only span a day at most. These near future predictions are commonly used in solar power prediction, as the short term predictions are most important for regulating the energy grid and for giving an estimate of the production for the day-ahead market.

The main advantages of using NWPs for solar irradiance predictions is that the data can be used on its own without the need of additional sensors and can be accessed from anywhere on earth as long as the installation has an internet connection. This allows for smaller models that can be run on an embedded device as the heavy computation is offloaded onto a central computer. Although the technology of NWP is steadily advancing and is made available with greater accuracy, the data can still span multiple kilometres and is predicting the average weather for that region, which might not apply for a location within that region due to nearby environmental factors. This observation has been an inspiration for the research done in [15] to improve the accuracy of the prediction. The improvement is based on site specific observations and on neighbouring installations. There the RMSE score improved from a 36% for a single station to 13% for the complete area of Germany.

The most common sources of weather forecasts are by the ECMWF with their High resolution forecast (HRES) and Ensemble (ENS) forecast. The ECMWF also provides another weather forecast on behalf of the Copernicus Atmosphere Monitoring Service, which uses the same base model as ECMWF but with additional parameters estimating atmospheric compositions. The national weather institute of the Netherlands also has its own NWP called HARMONIE-AROME with a higher spatial resolution compared to the other NWP. The specifications of each prediction model are listed in Table 2.1.

	ECMWF HRES [35]	ECMWF ENS [36]	HARMONIE-AROME [37, 38]	CAMS [39]
Version	47r3	47r3	Cy40	47r3
#Forecasts	1	51	1	1
Time Horizon	10 days	15 days	2 days	5 days

Time Resolution	1h	3h	1h	1h(single level)
				3h(multi level)

Spatial Horizon	Global		Western Europe	Global
Spatial Resolution	0.1° x 0.1°	0.2° x 0.2°	0.05° x 0.05°	0.4°x0.4°
Vertical Levels	137		N.A.	137
Forecast start	00:00, 06:00, 12:00, 18:00		00:00, 06:00, 12:00, 18:00	00:00, 12:00
Available after	6h	7h	3h	10h

Table 2.1: NWP specifications

2.2.2.4 Local Measurements

Next to the NWP, it has also been observed that including local weather measurements into the solar power predictions can give better accuracy, as shown in [24]. These improvements are only significant for near future predictions (up to 3 hours) as the paper showed that taking the state of the PV-installation itself has a significant impact on the power output. One such variable is the temperature of the solar panels which might not match the ambient temperature as computed by the NWP, which impacts the power output of the solar cells. Another study [25] showed the impact of several variables that can influence the performance of solar cells, here the correlation between humidity and the output power was noted to be a noticeable factor in the prediction.

However, depending on the PV-installation it might be that these measurements are not available as it requires additional sensors that have to be installed near the installation. For large solar parks this might be a feasible addition, but for households an additional array of sensors might not be worth the cost. Therefore, [30] employs the spatial and temporal correlation of multiple PV-installations nearby to fill this gap by using the realized power production of these installations as a substitute for a local weather measurement setup.

It should be noted that the correlation between measurements can differ significantly based on the time resolution of the measurements and their distance apart. This has been studied in [40] where the evolution of two clearness indices correlated over time is plotted as a function of the distance they are apart, their correlation decreases significantly faster over distance as the time resolution of the measurements increases from 60 minutes to 1 minute. Therefore, the number of useful sources are limited to their locality and update frequency as temporal resolution requirements increase.

2.2.2.5 Solar Power Logs

To predict the generated power of a PV-panel, one needs actual datasets of these measurements. There are however not many publicly available datasets that have this information. Hence, most papers opt for a locally sourced dataset or private datasets from solar companies that cannot be shared. These measurements are either from one up to a few solar installations or span a certain region, as can be seen in the datasets listed in the review papers [14, 13]. These recording usually don’t last more than a year. As a result, the training data is from the same location and time as the data used to validate the model, leaving models that perform well but are specialized for the data it has encountered. Meaning, the model is trained for that specific location and time period, but when used in practice might not hold for other locations and time periods as the model has never been configured with that data before. Hence, it is difficult to compare models and justify the validity of said model, as is the conclusion of two survey papers [14, 13].

An Australian study [3] from 2012 showed the possible pitfalls of integrating PV-installations into the Australian energy grid. Based on measurements from solar parks, it discusses the current situation, the problems that the volatility and unpredictable nature of PV-energy cause and possible remedies for these problems. Key observations include that cloud coverage is the main cause of the variability, which cause sharp fluctuations in the PV-output in less than a minute, for which most NWP do not have the spatial or temporal resolution to be able to predict these changes. It states that higher (temporal) resolution data is required to manage solar intermittency issues. This would not necessarily apply to large solar parks due to the area the panels cover, as the shade cast over the panels would change more gradually over time compared to a single panel. It therefore proposes to also use the previous PV-output as an input to the model to give a better estimate in which the range of future predictions lie, as these can be measured directly and should always be available. It also states that high impedance energy grids (small, local or rural grids) experience more frequent voltage swings that can cause degradation of performance, or in some cases trip safety functions in the solar inverters stopping the PV-output all together due to these quick fluctuations. Therefore, PV-panel measurements with a temporal resolution of less than 1 minute are preferred.

2.2.3 Summary Data

In the field of solar and wind power prediction, there is a lack of commonality in benchmark methods, datasets, and parameters used by these studies. The differences range from the parameters used to the locality, time and time interval of the dataset. A big influence on the locality and time aspect is the lack of publicly available datasets of the actual power measurements of these PV-panels. Therefore, most papers use their own datasets or have requested private access to the data of local providers, resulting in a segmented state where comparing methods becomes difficult and methods are adapted to the available data. This is a conclusion which both the second and third survey share [14, 13]. This is not a problem if the purpose is to use the model for those situations and share the findings, but claiming that one method is generally better than another is more difficult to support. Next to this, a common issue among the studies from the surveys and the specific ones listed here is that the data used is not documented well enough to reproduce the results. This is again a conclusion that both previously named surveys share.

Furthermore, most parameters are selected based on their correlation with the output, this however can limit the accuracy of the model as is shown in [25] where, although RH is not correlated with the output its inclusion does lead to a better and more accurate model. The (Pearson) Correlation test determines how linearly correlated two parameters are. When this score is utilized in the selection procedure it assumes that the best parameters share a linear relation with the output, this is however a rather limited view on the problem. Firstly, the assumption of a linear relationship would exclude non-linear relationships such as parabolas, which can carry additional information. Secondly, two parameters together might give more information about the environment than they would on their own, which is not covered in the selection process. Lastly, the parameters might not directly correlate with the current output, but can give better predictions for the future output if those parameters indirectly influence other parameters in the future predictions. This makes selecting parameters rather restrictive and can lead to models which might not perform at the best of their ability. So more tests with different parameters per model need to be done before parameters are prematurely written off or use a different method to select these features which improves on these drawbacks.

The common strategy in dataset selection is the use of an NWP as an input for the models and are often augmented with local measurements if available, either from on-site measurements or from weather stations nearby. Information about the installation environment is almost always used to either pre-process the data or used in the model itself. Combining data from multiple sources often show better performance in short term predictions compared to inputs on their own, and are explored further in this thesis.

2.3 Current Prediction Methods

Applying AI methods in solar power prediction is not new. Many studies show implementations following popular AI methods or derivations of them. Both [11] and [14] conclude and recommend that hybrid combinations of these methods give better accuracy and should be pursued over the single method approach. Both papers cite over 50 papers each as a basis for their conclusions. For more in depth information about the different methods and combinations, it is recommended to look into these surveys.

In this section, some well regarded methods are described in Section 2.3.1 to give an idea of what has been attempted before and their outcomes. After that, a few noteworthy architectures related to the use cases of this thesis are discussed in Section 2.3.2.

2.3.1 Common methods

2.3.1.1 Linear Regression

Linear regression is a modelling approach where it is hypothesized that the output variable y has a strong linear relationship with n other independent variables $\boldsymbol{x} = \{1, x_1, x_2, ..., x_n\}$. As each variable has its own relation with the output, the contribution of each variable to the final result has to be weighted by its significance as well, this weight is denoted as $\boldsymbol{\beta}$. This relationship is best described as the sum of all contributions where each input variable has its own individually weighted contribution and can be formulated as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon \tag{2.1}$$

$$y = \boldsymbol{x}^T \boldsymbol{\beta} + \epsilon \tag{2.2}$$

$$\hat{y} = \boldsymbol{x}^T \boldsymbol{\beta} \tag{2.3}$$

$$y = \hat{y} + \epsilon \tag{2.4}$$

Here β_1 is the weight that is assigned to x_1, β_0 is the bias or offset parameter and is independent of the input variables, ϵ denotes the error between the prediction $\hat{y}$ and the actual output value y.

In linear regression, the goal is to make the error ϵ as small as possible, such that the prediction $\hat{y}$ matches y as best as possible. To do this a loss function is used, loss functions give a score, or loss, of how good the predictions made by the model are with respect to their error. Common loss functions are the sum of the squared errors $L(\boldsymbol{\epsilon}) = \sum_{i=m}^{M} \epsilon_i^2 = ||\boldsymbol{\epsilon}||_2^2$ or any other norm function. To find the optimal values of $\boldsymbol{\beta}$ the gradient of the loss with respect to the weights is taken $\frac{\delta L(\boldsymbol{\epsilon})}{\delta \boldsymbol{\beta}}$. Solving for $\frac{\delta L(\boldsymbol{\epsilon})}{\delta \boldsymbol{\beta}} = 0$ with respect to $\boldsymbol{\beta}$ give the optimal values for these weights.

Linear regression is often expanded upon by changing how the inputs are presented to the algorithm. For example, one might instead of just providing x also provide x^2 as an input, this way the model $\boldsymbol{x}^T \boldsymbol{\beta}$ is a polynomial fit of x to y. In this case x was squared, but any function can be used to transform x first, as long as there exists a derivative of that function.

Some nice benefits of using linear regression are that its weights give direct insight into how y would change relative to a change in x because of this linear relationship. Now the significance of each input variable to the output can be understood immediately. A large magnitude for a certain weight means a higher correlation with the output compared to a small magnitude. As such, linear regression can be used for feature selection by sorting the input variables by their weights and selecting the variables with the most significant contributions.

2.3.1.2 Support Vector Machine

A Support Vector Machine (SVM) in its original form is a supervised learning method used for classification problems. Classification is done by separating the two different classes with a hyperplane and decide which side of the plane the sample belongs to. A hyperplane is defined as: $\mathbf{w}^T\mathbf{x} - b = 0$, where $\mathbf{x}$ is the input space or the feature space and $\mathbf{w}$ is a learned weight vector that determines the slope of the hyperplane, b gives an offset to the plane. A sample x_i can be classified by taking the dot product of the learned weight vector w with the sample and subtracting the bias b from it. Based on the sign of the result, the sample either lies above or below the hyperplane, meaning it either belongs to class A or B. A 2D visual representation is shown in Figure 2.1.

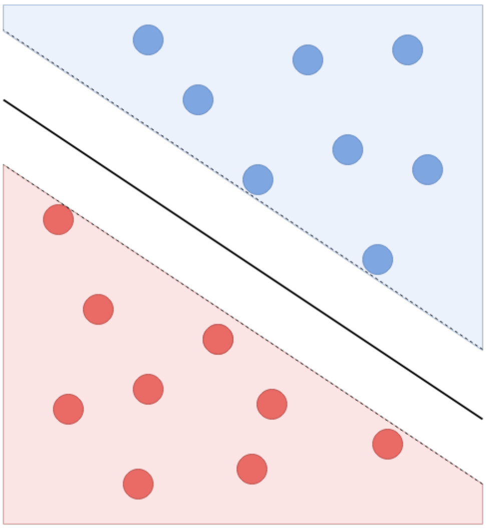


Figure 2.1: An example of a 2D SVM classification problem. Here the two classes are separated by a hyperplane, of which the zero intersection is shown as the black line. The red and blue shaded areas form the classification regions, where if a sample falls in that area it is classified as such. In the transition area no classification is given to the sample.

In the original version the classification needs to have a minimal boundary between the intersection of the plane and the nearest sample, the optimization function requires this as the goal of SVMs is to maximize the gap that is defined by the boundary between the two classes that the hyperplane creates. This was defined as follows:

$$y_i = \begin{cases} 1, & \mathbf{w}^T\mathbf{x}_i - b >= 1 \\ -1, & \mathbf{w}^T\mathbf{x}_i - b <= -1 \end{cases}$$

This however requires the classes to be linearly separable, which in practise might not be the case. To solve this issue, a different optimization objective can be used. Instead of maximizing the gap between the classes, the distance from samples that are on the wrong side of the plane is minimized. This allows the model to still give its best effort classification where it is sometimes wrong, instead of not being able to give a classification at all. This different update method is called the soft-margin approach and has multiple functions that can be used to calculate the distance between the plane and the sample point.

SVMs are commonly used in combination with what has been called the “kernel trick”. With this method, instead of trying to find which side of the hyperplane the sample is on using the linear mapping $\mathbf{w}^T\mathbf{x}_i - b$, the SVM first maps the sample onto a higher dimension using a non-linear kernel function. Classification is however still done with a hyperplane, but now acting on a higher dimensional feature space. SVMs have been used in [18, 27] to predict the generated power of PV-panels.

2.3.1.3 Decision trees

Decision trees are hierarchical rule based classifiers. The decision-making is done by asking questions (rules) about the data and based on the answers follow an associated path (branch) to a next question or arrive at the classification result (leaf). An example is shown in Figure 2.2.

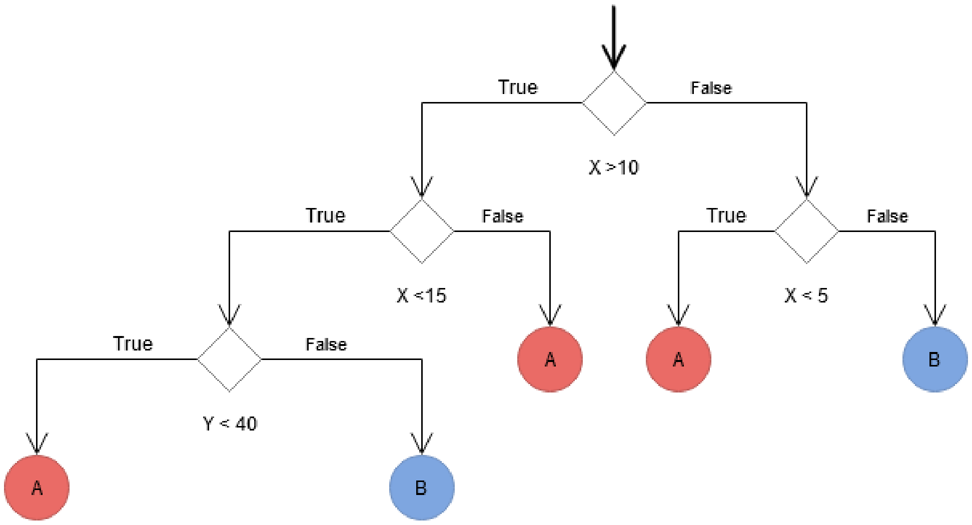


Figure 2.2: An example of a decision tree. Here the process starts at the arrow at the top, where depending on the sample a path along the tree is followed until it ends up in a leaf where the sample is classified.

Random (Decision) Forests are like decision trees. But here multiple trees are trained on different parts of the training dataset or on different features. The results of each tree are then combined to give, for example, for classification the probability for each class based on the classification of each tree, or for regression a (weighted) average of each tree. This method was used in [26] to estimate the IV-characteristics of the solar panel in the near future. Here it showed a mean absolute error percentage (MAPE) of 0.13%, performing better than Neural networks and with lower training and execution time. This method can be used as a last step in predicting the actual PV-output, as the expected irradiance values can be used as an input to this model. This step has shown high correlation with the actual environment of the PV-panel as described in [25], and can improve the accuracy of the prediction over the use of a static PV-model.

2.3.1.4 Artificial Neural Networks (ANN)

One of the most commonly known methods in the field of AI are Artificial Neural Networks (ANN). These networks have been applied in various fields and purposes. The main advantage of an ANN is its ability to approximate any continuous function, this is known as being a universal function approximator. The most basic neural network is a dense-connected network. A visual representation of a dense network is shown in Figure 2.3. Each node in a layer, also called a neuron, computes the weighted sum of its inputs and a bias. This sum is known as the activation. The output is the activation value after it is applied to an activation function. This function is often non-linear, as it allows the whole network to model non-linear behaviour. This process for one layer can also be represented as follows: $\mathbf{y} = f(\mathbf{W} \cdot \mathbf{x} + \mathbf{b})$.

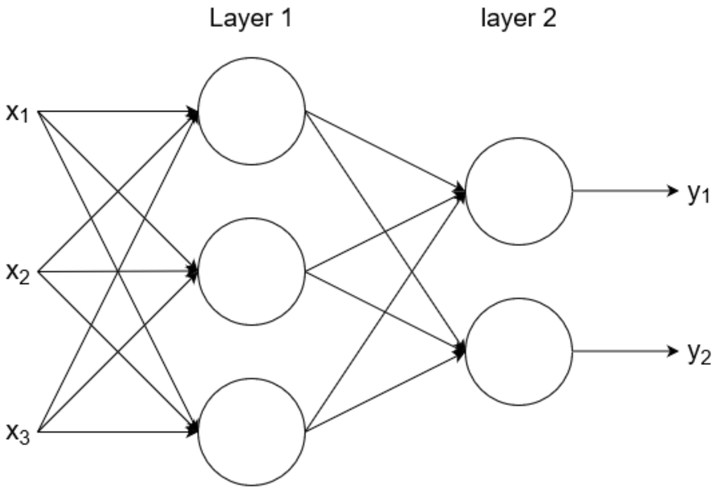


Figure 2.3: An example of a dense neural network consisting of two layers. Each line represents a multiplication of the input with the weight associated to that input and the target node. The circle represents the node where the sum of all arrows is taken, and the activation function is applied. This process is then repeated for each layer after that until the final answer is obtained.

An ANN is trained by the use of a loss function, which describes how well the network performed. This decision is based on the output of the network and its expected output. The training algorithm then updates the network weights and biases or other trainable parameters to find the solution for which the loss function is minimized. An example of such a training algorithm is the gradient descent. Here, the weights are updated based on the derivative of the loss function with respect to the output of the network. The new weights are updated in the direction where the gradient of the loss function is lower, like a ball rolling down a hill. This update rule can be described as follows where η represent how far it moves down the slope:

$$w_{i+1} := w_i - \eta \cdot \nabla f(w_i)$$

A downside of this algorithm is that it is possible to get stuck in a local minimum, as it always moves down the slope. This can be helped by adding momentum to the update rule, or use a more modern update rule like ADAM [41]. The problem of finding the most optimal solution is however not only restricted to getting stuck in local minima. Another significant problem is the solution space, which grows in dimension for each parameter that is learned. In a dense network where each weight and bias are learned parameters, the solution space can grow quite quickly. This often means more local minima in the solution space and more iterations that have to be taken for the update rule to find a minimum. In addition, it is also possible that when the training data is not sufficient, larger networks due to their large solution space can show overfitting. This is a phenomenon where the output of the network matches the training set closely, but under performs when presented with new unseen data. This can also be seen as the network not being able to generalize the data well enough due to noise in the data or missing data where the network is filling in the gaps incorrectly.

In order to solve these issues, there have been new methods proposed to reduce the amount of parameters that have to be learned. In general, these solutions can be summarized as weight sharing or better information extraction/representation. Two common examples are Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

Convolutional Neural Networks (CNN)
Normally each node in a dense network layer takes all previous results into account, as can be seen in Figure 2.3, this leads to many connections in the network and thus also many weights. A node in a CNN only takes a small subset or window of the input into account when calculating its result. This reduces the amount of connections that need to be calculated and the number of weights in the model. Next to that, each node shares the same weights with all nodes in the layer, reducing the number of weights further. The resulting sliding window with shared weights closely resembles the convolution operation, hence the name. The layer has a few parameters, the kernel/window size describes how many inputs should be taken into account for each node. The next parameter is the stride interval, or how many inputs the sliding window should shift over by between nodes. The number of filters describes the amount of different convolutions the network should perform, each filter has its own set of weights.

This network structure is often used when images are involved as the relationships within nearby pixels often hold more information compared to distant pixels, but this structure can be used whenever it is assumed that nearby data points hold the most information. An example in the field of solar energy forecasting is the research done in [32] where CNNs are used to process images made by a total-sky-imager to predict the GHI. A total-sky-imager gives a 180° dome view of the sky above, this image is used together with lagged measurements of the GHI to model the effect of cloud cover on GHI.

Recurrent Neural Networks (RNN)
Recurrent neural networks are employed when time series data is used as an input. These networks try to find patterns in the past data which can help predict the (near)future, the most well-known RNN is the Long Short-Term Memory (LSTM). LSTM rose to prominence as most RNN suffer from the “vanishing/exploding gradient problem”, where the back propagation gradient would become too small or infinite such that the weights would not change or be undefined due to its (infinite) recursion over past time series data. LSTM limits the amount of past samples that are included in the backpropagation gradient, while still retaining information outside this time window in its internal state. It does this by deciding based on the current input and its internal state if it should store/remove/output information. The network is still recurrent, as for each time step in the data the LSTM cell receives the hidden state and activation state from the previous time step as an input. Although the data passed to each LSTM cell is not the same for each cell, the cells do share their weights across time. Some notable uses of recurrent networks in PV power prediction can be found in [21, 42].

2.3.2 Special Architectures

The previously mentioned methods can be modified and/or combined to make the model better equipped for these predictions. These modifications are based on assumptions about the environment which can help generalise the model, reduce the complexity, and filter out noise from the input. Some architectures of note are described here with their applications.

2.3.2.1 Autoencoders (AE)

Autoencoders represent the input data with as little variables as possible. Autoencoders are neural networks that consist of two parts, an encoder network and a decoder network. The encoder outputs a transformed version of the input data with as little variables as possible, whilst the decoder tries to reconstruct the original input data as accurately as possible based on the output of the encoder. This structure ensures that the most relevant data is kept from the input and the redundant information is discarded, giving a more generic description of the data. This strategy allows for the network to be used as a pre-processing step to reduce the number of input dimensions and perform feature extraction on the input data for other models. The encoder network is used in these situations as an input for other models to make their predictions better. Or for noise filtering, as the output from the decoder is also more generic and thus less affected by noise on the input. Some examples of these autoencoders can be seen in [43, 44, 45]

2.3.2.2 Clustering

Not all data is unique, and often shares similar behaviour with other samples from the same data. As such, it may be worthwhile to find these samples and group them together, this is called clustering. For weather data these types could be for example sunny, rainy, or cloudy. There are many methods and techniques to try to find these clusters in the data, some of which are discussed here.

A simple form of clustering can be done manually, as is done in [22] where the data is divided into 4 clusters based on manually defined thresholds for DHI and GHI. In that paper, a model is trained on each cluster and later combined into one single output.

Another method is called K-Means clustering, where it is assumed that the data can be divided into K clusters and the data is assigned to the nearest cluster based on a distance function. This method learns where the centres of each cluster are by iteratively moving the centres to the mean value of the samples that belong to that cluster. This method is used in [18] to divide the data and train a model on each group, which is later combined. K-Means clustering is combined in [45] with an autoencoder to find these clusters on an embedded representation instead of directly in the dataset.

Another approach, as is done in [44], is to modify the output of the encoder stage from an autoencoder to only output its k-highest activations and set the others to 0. By gradually decreasing k from the full width of the output to 1 during training, the output of the encoder will no longer represent features but a cluster. The actual value of k does not have to be 1, an optimal value for k can be small. This would mean that combinations or a superposition of clusters would give a better representation of the data. A similar technique is used in [43] where the sparse activation technique is applied to the filter activations of a convolutional layer and should result in more unique filters found by the network. This would mean one filter is selected to describe that part of an image.

A disadvantage of manual clustering and K-Means clustering is the inherent assumption that the number of clusters can be known beforehand, or how they can be separated. This becomes difficult when describing the weather, as it changes continuously between situations. For example, the weather shifts from sunny to cloudy and somewhere in that time the classification of the weather also has to change. For manual clustering, this transition point would be well-defined, but the values near the transition point would not be well represented as it is not completely cloudy or sunny. For K-Means clustering, the transition is influenced by the distances of the data point to the cluster means, and the classification can transition gradually. But this all depends on how the cluster means are distributed, and not by the characteristics themselves. [45] improves on K-Means clustering by learning an embedded representation which can better shape these transition boundaries. K-Sparse [44] learns an embedded representation as well, but on top of that allows for multiple independent classifications to exist at the same time, compared to a competitive classification using K-Means clustering.

2.3.2.3 Combinations

A general recommendation from the survey [11] is to combine multiple deep learning strategies to achieve better performance. The idea here is to play into the strengths of each method or to compensate for its shortcomings. These combinations can be described by three characteristics:

1. **Series** models use the output of one method as the input of another, the autoencoder examples described before would fall in this category. These are commonly used with pre-processing steps like feature extraction, or in the case of [22] to separate climate and weather to reduce location and seasonal influences.
2. **Parallel** models or Ensemble models use multiple methods in parallel on the same input and are later combined to produce one output, as is used in [29] for irradiance prediction. A more formal description of ensemble methodologies and their respective combination strategies are listed in [46].
3. **Meta** models are used in the development of other models but are not used in the final model. For example, in [18] the dataset is divided based on k-means clustering to train different methods for different types of weather. Another example is feature selection where the model tries to select the best features in the input to be used by the other model as is done in [47] and shows promising results as it can learn complex non-linear relations in the data compared to standard feature selection based on correlation or linear regression.

2.3.3 Metrics

Metrics give a score of how well a model performs compared to another, therefore making it possible to quantify how much better one model performs over the other. These metrics are designed with an attribute in mind. Such an attribute could be how far off a model's prediction is compared to the truth, how consistent its accuracy is over time, or how computationally complex a model is. The most commonly used metrics found by the survey papers are explained below.

- **Mean Absolute Error (MAE)** measures how far the prediction was off from the observed value. When taken over all samples, it represents the average error from the true value. A smaller value means a better prediction. The value has the same unit as the variable. The MAE is also often scaled to the observed value, in that case it would be called the Mean Absolute Percentage Error(MAPE).

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_{i,forecast} - x_{i,observed}|$$

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{x_{i,forecast} - x_{i,observed}}{x_{i,observed}}\right|$$

- **Mean Square Error (MSE)** measures how far the prediction was off from the observed value, but penalizes extreme errors more and small errors less compared to the MAE. If the square root is taken of the MSE it is called the Root Mean Square Error(RMSE). The RMSE does share the same unit as the variable itself, like the MAE.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_{i,forecast} - x_{i,observed})^2$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{i,forecast} - x_{i,observed})^2}$$

- **Mean Bias Error (MBE)** gives a measure of how much the model, on average, is above or below the observed value. Here a positive value would mean that the model is overestimating the actual value and negative means underestimating, 0 would mean no bias is present. The MBE indicates how much the prediction should be trusted, as over or underestimating the actual value can have different consequences depending on how the model is used.

$$MBE = \frac{1}{N}\sum_{i=1}^{N}(x_{i,forecast} - x_{i,observed})$$

These scores are often reduced to one number for all samples in the dataset, giving one score to summarize the performance of the model. Most methods only present this score as the performance metric of the models, without looking into uncommon situations that may be present in the data. As these scores give a summary over all situations and time, meaning that there can be situations where the prediction errors are significant, but are not reflected in the results and thus relying on these scores alone might be misleading to the actual performance of the model in all situations. To remedy this problem, some papers divide the predictions based on these types of pitfalls and give individual performance metrics for each of them. These divisions could be based for example on the time of day/year, the Clear Sky Index, the prediction time window, or weather type.

2.3.4 Benchmarks

Often the performance of a model is compared to other models, but this would mean that the performance increase claims only hold relative to these models. In order to move from the relative performance claims away to a more grounded claim, benchmark models are employed. Benchmark models are used to show that the proposed model performs better than a model which is based on simple and often more intuitive assumptions. If the proposed model performs better than the benchmark model, it would mean that the proposed design does have a better representation of the inner workings of the environment. There are two main benchmark models used in this field:

2.3.4.1 Climatology

The climatology benchmark model is based on the assumption that the energy production at a certain point in time is similar to previous measurements at similar times of the day and year from the past. Meaning that for a prediction for a day in June, the energy production would be the similar to that from the same date a year or multiple years ago. The model can be the actual measurement from a year ago, an average over the last few days at a certain point in time, or a model fitted to data from the past with time of year and time of day as its input parameters. A common model used here is a linear regression model with the time of day and time of year as its inputs.

2.3.4.2 Persistence

The persistence benchmark model is based on the assumption that the future energy production will not change much from the current energy production. As such, this model predicts that the energy production for all time windows in the future will remain the same as the current measurement. This model performs well for time windows in the near future, but start to degrade for more distant windows in the future. Some papers use a modification to this model where the last few samples are also included in the model, this way the progression in the production can be extrapolated from the past measurements.

2.3.5 Summary Methods

Based on the methods discussed in this section, a focus on generalization and classification/specialization for weather data seems promising, as a few papers show their models improving when grouping their weather data. Therefore, the feature selection architecture called LassoNet proposed by [47] is looked at further for feature selection and the sparse clustering approach used in [44] is considered. The proposal by the survey papers [11, 14] recommending the combination of different methods is followed. The need for better evaluation of the performance of models using the aforementioned metrics is followed by additionally searching for possible pitfalls and conditions where the model might perform worse. For example, by comparing dependencies of the models of rainy days to other days, or looking for a dependency on the time of day, time of year, the clear sky index, or the magnitude of the power production.

2.4 Probability

By their nature, weather forecasts have a systemic uncertainty in their predictions, as weather itself is too complex to accurately model within current practical limitations. This uncertainty in turn also influences the performance of the solar energy forecasts, which depend on accurate weather forecasts. Currently, most solar energy forecasts use point forecasts, these are predictions where the future energy production is given as a single value. This value is the best guess of the model based on the data that is available, but on its own cannot provide insight into the likelihood of that actually happening.
Research has been done to move from these point forecasts to probability forecasts, which by design contains information about the possible spread of future values and the likelihood of those happening. In this section, an overview is given on how point forecast models can be replaced with probabilistic ones. First, some necessary background and theory on probability is introduced. Secondly, what methods can be used to produce probabilistic forecasts and how they compare to their point forecast counterparts. After that, the possible performance metrics and benchmark models are introduced. Finally, in the conclusion, the advantages and possible pitfalls of probabilistic forecasting are discussed.

2.4.1 Types of Uncertainty

Probability
For point forecasts, the output is assumed to be a deterministic value, meaning that the value can be determined beforehand. In contrast, the probabilistic forecast provides the possible values that can happen from this point and how likely they are to happen. An example would be rolling dice, where the deterministic forecasts will predict what value the dice will land on, while the probabilistic forecasts would predict the chance of landing on that side as well as the chances of landing on any of the other values. In the following paragraph, the rolling of the dice is used as an example for all the mathematical facets of probability.

To start of with, a probability of something happening is given in the (0-1) range, where 0 means it will never happen and 1 means it will always happen. For a standard 6-sided dice the chance of rolling a 1 would be $\frac{1}{6}$, rolling a 2 would also have a $\frac{1}{6}$ chance, etc... . Let's say that we've been rolling the dice a lot and have noted down these outcomes. The outcomes of these rolls are grouped under a random variable, for example the random variable X. The random variable X would map in this case each roll that we've done to a single number x, the side that the dice landed on. So x is a particular value out of the random variable X. This allows us to specify constraints to what we consider to be part of the experiments and its influences. Moreover, it gives us a notation for describing probability: $Pr(X = x)$ would describe the probability of rolling x given the random variable X. The example of rolling a 1 can be written as $Pr(X = 1) = \frac{1}{6}$ and the chance of rolling 4 or less can be written as $Pr(X \leq 4) = \frac{4}{6}$, these examples are also represented in Figure 2.4 for the more visually inclined.

Rolling a 7 cannot happen, so the probability of $Pr(X = 7) = 0$, the chance of the dice landing on any of its sides is $Pr(X = anyside) = 1$ as the dice must always land on one of its sides. This last example can be generalised to a simple rule: the probabilities of all possible outcomes added together must equal 1: $Pr(X = anyside) = \sum Pr(X = side) = \frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6} = 1$. Consequently, one can use this rule to calculate the probability of the opposite happening. If there is a $\frac{4}{6}$ chance of rolling a 4 or less, there is also a $\frac{2}{6}$ chance of rolling above a 4: $Pr(X > 4) = 1 - Pr(X \leq 4) = 1 - \frac{4}{6} = \frac{2}{6}$. These examples shown here are based on discrete events and outcomes, but these rules also apply in the continuous case, for which examples are given in Section 2.4.1. These examples are meant to give a basic understanding of what probability entails and its terminology. This is built upon further in the following sections.

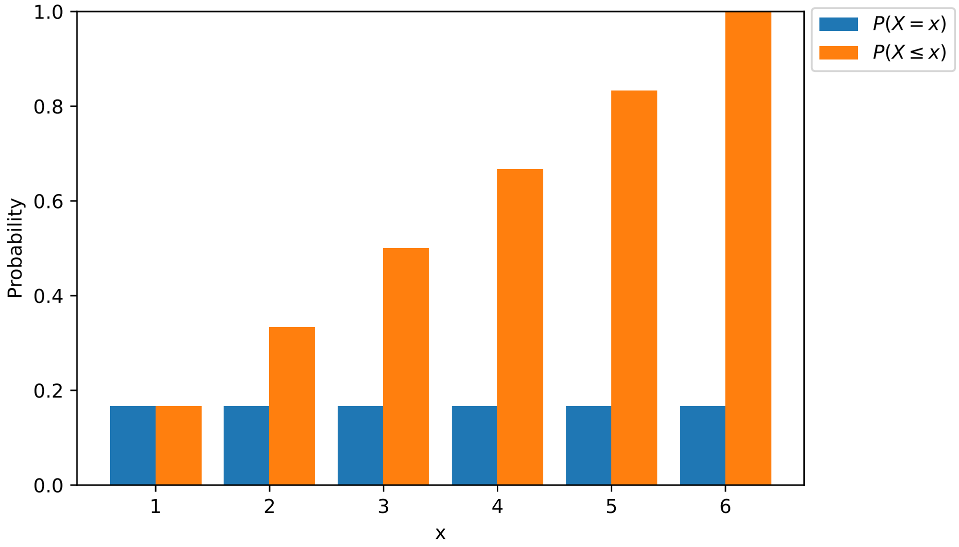


Figure 2.4: A visual example of what the probabilities are for each possible outcome on a standard 6-sided dice (X), the chance of landing on each number $Pr(X = x)$ and the chance of rolling less or equal to that number $Pr(X \leq x)$.

Aleatoric and Epistemic Uncertainty

When talking about probability the term uncertainty is also often brought up, like in the dice example the outcome of the role is not known beforehand and as such its outcome is uncertain. To make it easier to talk about uncertainty, the term can be split into two definitions: the aleatoric uncertainty and epistemic uncertainty. The aleatoric uncertainty refers to the uncertainty of the outcome, for example the uncertainty of what value the dice will land on or due to noise in the sensor readings. This uncertainty will always be there and can be referred to as a “known-unknown”. This uncertainty can be quantified or described with models, but the actual outcome will never be known until it happens. Mitigating this type of uncertainty is rarely possible, due to its random nature.

The epistemic uncertainty is the uncertainty due to a lack of information. This lack of information could be due to sensor measurements that are not available, or missing information about situations that have not happened yet. For example, a rare weather event might not be in the dataset, so how the model will behave in that case is uncertain. Or there is a possible influence on the outcome that is not known or modelled yet which might be perceived as an aleatoric uncertainty in the model at first, but is actually an epistemic uncertainty. This type of uncertainty can be reduced by gathering more data, adding more input types to the model, or gain more insight into the inner workings of what is modelled. But reducing this is not always a practical approach due to for example time or monetary constraints.

Probability Distributions

Probability distributions describe the likelihood of the possible outcomes to happen, an example for discrete random variables has already been discussed in Section 2.4. For continuous random variables, a similar visualization can be made as for the discrete random variable and is shown in Figure 2.5. The probability density function (PDF) $f(x)$ describes how a random variable is distributed over its range and how frequent a value in that range is observed, similar to the $Pr(X = x)$ histogram shown in Figure 2.4 as they are each-other’s discrete and continuous counterparts. However, for the PDF, the height of the graph is not the probability of that value happening. This is because the range of possible values for $f(x)$ is infinite as it is a continuous random variable, in contrast to the discrete random value whose range is bounded. This, together with the rule stating that the probabilities of all possible outcomes added together must equal 1, results in the fact that $Pr(X = x)$ will always be 0. In order to get a real probability from the PDF an integral over an interval can be taken, this would be written as $Pr(a < X < b)$. Contrary to the PDF, the cumulative distribution function (CDF) or $F(x)$ does describe a probability. The CDF describes the probability of the continuous random variable X being smaller or equal to x, as is done for $Pr(X \leq x)$ in Figure 2.4 for the discrete case. To put it more formally, the CDF can be obtained from the PDF by taking the integral of $f(x)$ up to z as done in Equation 2.5 and describes the probability of X being smaller or equal to z as shown in Equation 2.6. The observed value y, also called the realization of X, can be represented as a CDF as well. It is a step function with the transition at the observed value, or more generally as an indicator function as shown in Equation 2.7 where $expr = x \geq y$.

$$F(z) = \int_{-\infty}^{z} f(x)dx \tag{2.5}$$

$$Pr(X = z) = F(z) \tag{2.6}$$

$$\mathbb{1}\{expr\} = \begin{cases} 1 & \text{if } expr \text{ evaluates to true} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

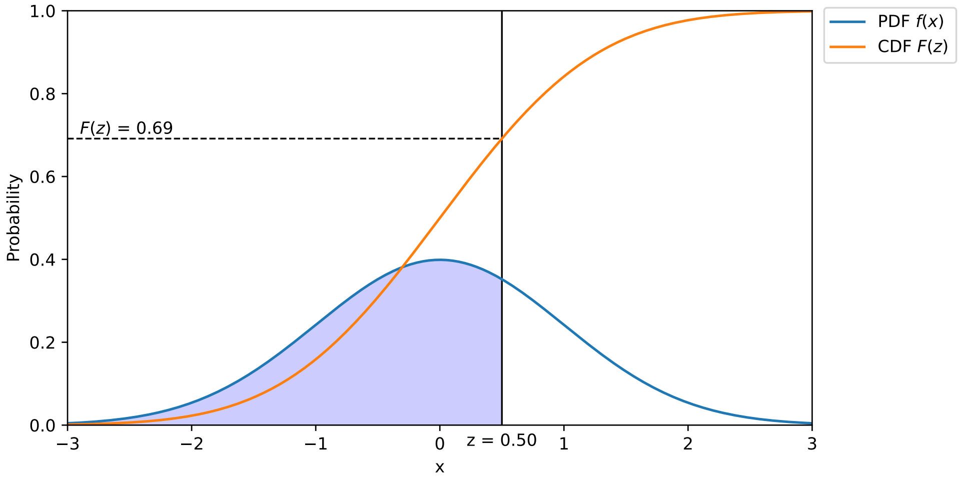


Figure 2.5: An example of a standard normal distribution showing the relation of the probability density function $f(x)$ with its cumulative density function $F(z)$

Parametric Distributions

Parametric distributions use a limited set of parameters to describe how the PDF and CDF could look like. A common parametric distribution is the normal distribution, which requires only two parameters (μ, σ) to define its shape.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{2.8}$$

Here μ represents the mean of the distribution and σ represents the standard deviation or more intuitively how spread out the random variable is around μ. The PDF and CDF with parameters $(\mu = 0, \sigma = 1)$ is shown in Figure 2.5. An extension of the normal distribution is the truncated normal distribution. where the range of possible values for the random variable is limited to the interval (a, b). This can be useful for PV power prediction, as the power can never be negative or more than its rated power. For the standard normal distribution predictions around these edges would lead to inaccurate predictions as values outside this range would still be possible and likely according to the standard normal distribution, because of this the truncated normal is favoured over the standard normal distribution in [20] to model GHI.

Non-Parametric Distributions

Using parametric distributions to model CDF and PDF often rely on assumptions of how the observed power PDF is shaped. But choosing a parametric distribution might not hold with reality, as is proven in [28] where the observed PDF and CDF did not follow the normal, t-location-scale or logistic distributions the data were fitted to. It therefore proposed to use quantiles to model the CDF. Quantiles can be seen as the inverse of the CDF, instead of giving the probability for certain values in X it estimates the values q_τ which lie in X where $Pr(X \leq q_\tau) = \tau$ holds. Or more formally defined as the quantile function as shown in Equation 2.9 and its density forecast $\hat{Q}_t$ as shown in Equation 2.10 for M quantiles at time t.

$$q_\tau = F^{-1}(\tau) \tag{2.9}$$

$$Q_t = \{q_{t,\tau}; 0 \leq \tau_1 \leq \tau_2 \leq .. \leq \tau_M \leq 1\} \tag{2.10}$$

This mapping of τ to q_τ is visualised in Figure 2.6 where the values for q_τ are estimated for a normal distribution. If the probabilities τ are evenly distributed, as is done in this example, the values for q_τ represent the cut-off points to divide the random variable X in equally likely chunks. In practise, when estimating values for q_τ, it might happen that the predicted values are not monotonically increasing with τ. By the definition of the CDF this must however always be true, as $q_{0.1}$ should be less than $q_{0.2}$ for the statement $Pr(X \leq q_\tau) = \tau$ to hold. A naive solution for this problem is by sorting q_τ first and reassigning τ based on the new ordering before estimating the CDF. But when quantiles cross, it is a good indication that the model is under performing due to unusual data at the input or the model cannot represent the actual distribution. A common case when this happens is near or during nighttime when the density is very slim and near zero, here the chance of quantiles crossing is more likely due to the limited numerical precision and accuracy of the model.

By using quantiles any distribution can be approximated, but choosing the appropriate number of quantiles is a trade-off between accuracy, computational complexity, and the size of the dataset. The size of the dataset matters, as increasing the number of quantiles also increases the chance of quantiles crossing due to the decreasing amount of samples that fall in the interval of two neighbouring quantiles. In [28] 18 quantiles are used to estimate the probability density, in [42] only 3 quantiles are used $\{0.1, 0.5, 0.9\}$, in [20] only 5 quantiles are used to estimate the probability density and in [48] two quantile estimators are used in series with 9 and 51 quantiles respectively. In these studies there is no justification is given for why this number of quantiles are chosen except for [28, 48] where the quantiles were chosen for form, readability, or convenience but none discussed trying multiple quantile configurations or a trade-off in performance.

Another way to create a non-parametric distribution is to let a point forecast model run multiple predictions, with each time a slightly different input to the model based on the uncertainty of the input data. With these different predictions, an empirical cumulative probability density can be modelled by sorting the predictions by value and assign each prediction a probability by the order of the sorting. For example, when a random ensemble of 10 predictions is made, the lowest predicted value is given the probability $\frac{1}{10}$ and the second $\frac{2}{10}$. From then on it behaves the same as the quantile ensemble with a regular spacing τ for the probability and q_τ being the sorted predictions. This method is also used by ECMWF to forecast the many ways that the weather can evolve over time. There 50 perturbations are made based on measurements done by weather stations and satellite readings taking their uncertainty into account.

The difference between quantile and random ensemble densities is that ensemble predictions are assumed to be from a regular spacing in the probability range when constructing the CDF, whereas the quantile function generates the predictions in X for these specific probability thresholds. This means that when the input data is not sampled properly or not enough predictions are made, the resulting ensemble does not represent the distribution of the random variable well and can lead to inaccurate forecasts. This issue has been studied in [49] where the amount of predictions that have to be made for the density estimator to be accurate needs to be above 1000 predictions before the random ensemble method becomes stable enough to be compared with the standard normal distribution, whereas for quantile ensembles 30 quantiles seems to be good enough.

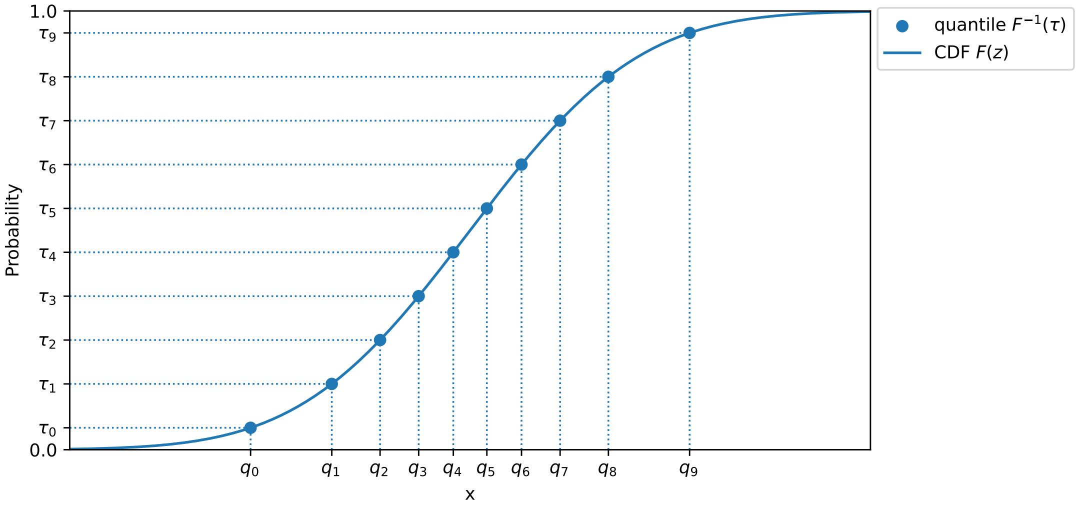


Figure 2.6: An example of how a CDF $F(z)$ of a standard normal distribution relates to its inverse, the quantile function $q_\tau = F^{-1}(\tau)$

2.4.2 Non-Parametric Methods

In this section, the most popular non-parametric methods as found by [13] are discussed. Only non-parametric methods are discussed, as shown in [13] the more recent studies moved from parametric to non-parametric as other studies have shown that the observed power distributions don't match parametric models well. The non-parametric methods listed here are all derivations of the methods previously mentioned in Section 2.3.1. As such, only the major differences is discussed here.

- **Quantile Regression (QR)** works similar to standard linear regression and shares the same matrix structure. But instead of training just one linear regression model, multiple models are trained in parallel such that each model predicts a unique q_τ using the quantile score listed in Section 2.4.3 as the loss function. The output of the model is the combined set of q_τ from which the CDF can be reconstructed. QR can suffer from quantile crossing as the predictions q_τ are made independent of each-other and its effects should be considered before using the output of the model.

- **Quantile Neural Networks (QNN)** are an extension of standard neural networks and similarly to QR the point predictions are replaced by a vector of q_τ predictions. QNN can also suffer from quantile crossing due to the possible complexity of the model. The effects of quantile crossing can be decreased by adding a penalty during training to enforce correct quantile ordering, this has been done in [28] along with a penalty to predictions made outside the range of possible values that the random variable can be in.

- **Monotone Composite Quantile Regression Neural Network (MCQRNN) [50]** takes the quantile crossing problem one step further by making it impossible for to occur by using a special activation function and constructing the network layers in such a way that each q_τ is guaranteed to increase with τ. This method is used in [20] to predict solar radiation.

- **Quantile Regression Forests (QRF) [51]** is similar to a standard regression forest, but it uses the returned observations O, which each tree thinks belongs to the same set as the input data differently. Instead of calculating the weighted mean over the observations, it calculates the empirical chance that the output is smaller than some threshold y. It does this by comparing each value in the returned observations with this threshold, and then takes a weighted average of the number of times the threshold y was smaller. This can be done for any number of y and when complete it can be processed like an ensemble forecast. The QRF is also used in [20] to predict solar radiation.

2.4.3 Metrics

- **Reliability diagrams [52]** give an indication of how often the predicted quantiles line up with the observed frequency, such that the premise of $Pr(X \leq q_\tau) = \tau$ indeed is correct. This means that in 10% of cases the observed values of y_t must fall below the predicted quantile $\hat{q}_{t,0.1}$ associated with $\tau = 0.1$, the observed probability can be calculated using Equation 2.12. The reliability can be visually assessed by plotting the nominal probability τ against the observed probability $\hat{\tau}$, this is called the reliability diagram. The reliability can also be represented with a score by taking the absolute distance of the two probabilities, as is shown in Equation 2.13.

$$\xi_{t,\tau} = \mathbb{1}\{y_t \leq \hat{q}_{t,\tau}\} \tag{2.11}$$

$$\hat{\tau} = \frac{1}{N}\sum_{t=1}^{N} \xi_{t,\tau} \tag{2.12}$$

$$Dev = |\tau - \hat{\tau}| \tag{2.13}$$

- **Continuous Ranked Probability Score (CRPS) [53]** is the generalised representation of the mean absolute error discussed in Section 2.3.3 and allows two density functions to be compared to one another. The definition of the CRPS is shown in Equation 2.14 for comparing two density functions, when comparing the density to a scalar observation the indicator function defined in Equation 2.7 can be used to represent the empirical CDF of the observation and is shown in Equation 2.15. The integral form of the CRPS is not practical in its application, as such it has been proven that for distributions with a finite first moment the CRPS can be written as Equation 2.16. For quantile or random ensembles, the CRPS can be estimated by the discretized version shown in Equation 2.17 with M representing the number of ensemble members. It is used in [20, 48] for the validation of their forecasts. As the discretized version of the CRPS is an estimation of the CRPS, the estimation will not always be correct. To remedy this, the research done in [49] gives recommendations on how to select the number of quantiles and the minimum amount of samples needed before the discretized CRPS becomes accurate.

$$CRPS(F,G) = \int_{-\infty}^{\infty} (F(x) - G(x))^2 dx \tag{2.14}$$

$$CRPS(X,y) = \int_{-\infty}^{\infty} (F(x) - \mathbb{1}\{y \leq x\})^2 dx \tag{2.15}$$

$$CRPS(F,y) = \mathbb{E}_X|X - y| - \frac{1}{2}\mathbb{E}_{X,X'}|X - X'| \tag{2.16}$$

$$CRPS(\mathbf{x},y) \approx \frac{1}{M}\sum_{i=1}^{M}|x_i - y| - \frac{1}{2M^2}\sum_{i,j=1}^{M}|x_i - x_j| \tag{2.17}$$

- **Quantile Loss (QL)/Pinball Loss [53]** is a weighted median estimation loss function where the error between the quantile and the observed value is weighted differently based on the value of τ the quantile represents and the sign of the error, its two most used definitions are shown in Equations 2.18 2.19. For quantile $\tau = 0.5$ both sides of the error would be weighted equally and is the same as the MAE in Section 2.3.3, whereas for $\tau = 0.2$ the contribution of the negative error would be weighted with 80% of the max weight and the positive side with 20% of the max weight. This shifts the minimum of the loss function to the point where roughly 20% of the values would fall on the left side and 80% on the right side. This function is used in [28, 42] to train their neural networks. This loss function is known under many names, “quantile loss”, “pinball loss”, “tick loss”, “linlin”, in this thesis the loss is only referred to as the quantile loss.

$$QS(q_\tau, y) = (q_\tau - y)(\mathbb{1}\{y < q_\tau\} - \tau) \tag{2.18}$$

$$QS(q_\tau, y) = max(\tau(q_\tau - y), (1 - \tau)(y - q_\tau)) \tag{2.19}$$

2.4.4 Benchmarks

In the literature study for this thesis, no probabilistic forecast specific benchmark model has been found, other than altered versions of the Climatology and Persistence models such that their output is a probabilistic forecast. These adjusted benchmark models are discussed here. But if the proposed models are compared using the CRPS for validation, the MAE error can be used for the point forecast benchmark models as they describe the same score without having to use these adjusted models.

Climatology
The Climatology benchmark described in Section 2.3.4 can be extended upon to produce either parametric or non-parametric probabilistic forecasts. For the parametric forecast one can use the linear regression model to predict the mean forecast and from that train another linear regression model to forecast the standard deviation such that both models combined can predict a (truncated) normal distribution. For the non-parametric forecast, one can swap the linear regression for quantile regression to model the distribution.

Persistence
For the persistence model, a similar technique to the climatology model can be used to go from a point forecast to probabilistic. For the parametric forecast, one can calculate the standard deviation of the sample data compared to the current power production for each of the future power windows. If a (truncated) normal is used, the current power can be used for μ and the calculated standard deviation for its σ. For non-parametric forecasts, the same distribution of the difference between the current power and the power production for each time window can

be used with the quantile function to determine the values for q_τ. These values for q_τ can then be added to the current power to give a density forecast.

2.4.5 Summary Probability

As recommended by [28, 20] the best choice for modelling probabilistic forecasts of PV-power installations seems to be using quantiles to model the probabilistic densities. As for the number of quantiles to be used, there is no clear selection criteria other than the recommendations given in [49] to use 30 or more quantiles to accurately model the CRPS when the observed values follow a normal distribution. CRPS seems to be a commonly used and easily interpretable score to use when validating the performance of the proposed models. For training a model with an ensemble, the Quantile loss is to be preferred over CRPS as the CRPS has no relation with how τ is distributed and results in learned ensembles without an associated τ. This would make the learned ensemble randomly distributed and from the results of [49] would mean a worse performance compared to regularly spaced quantiles. The QR and QRF methods are recommended by [20] over standard QNN or MCQRNN methods and are considered for this thesis. But as was shown in the deep learning surveys [11, 14] the use of neural networks give better results when special structures and constraints are used, and is considered the main focus for this thesis with QR and QRF as reference models.

CHAPTER 3

METHODOLOGY

Here, the developed model is discussed together with its dependencies and configurations. But first the available data is discussed, as their characteristics dictate the shape of the inputs and what methods are applicable. After which the general structure of the model is discussed and how each subcomponent fits into the model. Then each subcomponent is discussed individually on how and why the model is designed, along with its hyperparameters that need to be fine-tuned.

3.1 Data Selection

In this section, the choices for each data source are discussed. These are the PV-panel measurements, weather forecasts, satellite images used by the model. The data sources used for validating the performance of the model are also discussed here. Each source has its data dimensions listed, as well as the variables that are part of it.

3.1.1 Solar Power Logs

As summarized in Section 2.2.3, there are not many publicly available datasets of power measurements or standardized data collection guidelines for PV-panels. Next to that, power measurements made with a temporal resolution of less than 1 minute are preferred, as is discussed in Section 2.2.2.5. Therefore, a custom dataset is used that does have this high temporal resolution data and comes from the SlimPark car park introduced in Section 1.2. The data has a temporal resolution of 10 seconds and spans the full year of 2022, an overview of the PV measurement data and PV specification can be seen in Table 3.1. The power recorded in this dataset is measured after the solar power inverter, which converts the electricity generated by the solar panels and feeds it to the other connected systems. This dataset therefore describes the usable solar power available to the other systems and not the exact solar power generated by the PV-panels. This definition of solar power might not necessarily align with other datasets that are used in the literature, as specified in Section 2.2.2.5. It is however favourable for the EMS application described in this thesis, as no further transformations of the power or solar inverter models are required to estimate the available solar energy produced by the PV-panels.

3.1.2 NWP data

The Numerical Weather Prediction models described in Section 2.2.2.3 were all considered for this thesis, but due to practical limitations, only the CAMS option remained. The ECMWF HRES and ensemble models were considered, as the ensemble model inherently describe some level of uncertainty and would be interesting for this study. But for practical reasons the time required to download all predictions for the year 2022 would take 2 years for both the ensemble and the HRES dataset due to the high strain on the ECMWF public servers, as such these models were not feasible options for this study. The archive storing the predictions made

Dimension	Values
Measurement start	01/01/2022
Measurement end	31/12/2022
Temporal resolution	10 seconds
Total samples	3153600
Max power	27 kWp
Latitude	52.239891
Longitude	6.852906

Table 3.1: Data specifications of the SlimPark PV-installation located at the University of Twente [6].

by the HARMONIE-AROME model [37] does not go back far enough in time to get all the data for 2022, also removing it as an option. Therefore, only one possible weather model option was remaining, the CAMS model. The CAMS model [39] is implemented by ECMWF on behalf of the Copernicus programme of the European Commission with the same model behind it as the ECMWF HRES and ENS models, but with a reduced spacial resolution as well as less frequent forecasts.

Although the CAMS dataset [39] has a total of 498 variables available for download, the selection has been reduced to 29 variables based on the most often used variables, as discussed in Section 2.2.1.1. The complete list of variables and their identifiers in the CAMS dataset are listed in Table 3.3. The auxiliary data dimensions are listed in Table 3.2. As is listed in Table 2.1, every 12 hours a new forecast is made, each taking a total of 10 hours to complete. The forecast time steps are therefore taken for the hours 10 through 29, by doing this it is guaranteed that there will be at least a 7-hour window available for all power forecast times that fall in the 12 hours between forecasts.

Dimension	Values
Time window	01/01/2022 - 31/12/2022
Forecast interval	12 hours
Temporal resolution	1 hour
Forecast time steps	10–29
Latitude	53.0, 52.6, 52.2, 51.8, 51.4, 51.0
Longitude	6.0, 6.4, 6.8, 7.2, 7.6, 8.0

Table 3.2: CAMS data dimensions used.

3.1.3 Satellite

As discussed in Section 2.2.2.2 a proven source for irradiance forecasting using satellite imagery is done in [19] where the output of the SEVIRI satellite is used directly to model cloud movement. From the review paper [33] the use of SEVIRI seems like a good choice as well. As such, the possible derived products of the SEVIRI satellite were researched. There, two options were deemed useful, the filtered visible and infrared channels images, and a derived product made by EUMETSAT that produces a cloud coverage mask from these channels. Both options were considered until it became clear that the storage and processing time needed for the visible and infrared channels would prove unworkable. This is due to its 10 times larger memory footprint compared to the cloud mask, necessitating the processing of the images to be done by the shared and busy EUMETSAT server instead of locally before the preprocessed data can be stored and used. This would consume a large part of the time available for this thesis, therefore only the cloud mask remains as a viable option. This cloud mask is called the

Identifier	Description	Unit
u10	10 metre U wind component	m/s^2
v10	10 metre V wind component	m/s^2
t2m	2 metre temperature	K
cdir	Clear-sky direct solar radiation at surface	J/m^2
cp	Convective precipitation	m
dsrp	Direct solar radiation	J/m^2
fal	Forecast albedo	$(0-1)$
hcc	High cloud cover	$(0-1)$
lsp	Large-scale precipitation	m
lcc	Low cloud cover	$(0-1)$
mcc	Medium cloud cover	$(0-1)$
sund	Sunshine duration	s
ssr	Surface net short-wave (solar) radiation	J/m^2
ssrc	Surface net short-wave (solar) radiation, clear sky	J/m^2
sp	Surface pressure	Pa
ssrdc	Surface solar radiation downward clear-sky	J/m^2
ssrd	Surface short-wave (solar) radiation downwards	J/m^2
tisr	TOA incident solar radiation	J/m^2
tsr	Top net short-wave (solar) radiation	J/m^2
tsrc	Top net solar radiation, clear sky	J/m^2
tcc	Total cloud cover	$(0-1)$
tciw	Total column cloud ice water	kg/m^2
tclw	Total column cloud liquid water	kg/m^2
tcrw	Total column rain water	kg/m^2
tcsw	Total column snow water	kg/m^2
tcslw	Total column supercooled liquid water	kg/m^2
tcw	Total column water	kg/m^2
tp	Total precipitation	m
fdir	Total sky direct solar radiation at surface	J/m^2

Table 3.3: All downloaded parameters and their units from the CAMS global atmospheric composition forecasts available at [39]

“Cloud Mask—MSG - 0 degree” product [54] by EUMETSAT, an example output of this cloud mask can be seen in Figure 3.1. The data specifications used to download the data are shown in Table 3.4.

Dimension	Values
Time window	01/01/2022 - 31/12/2022
Temporal resolution	15 minutes
Latitude	56 - 48
Longitude	-1 - 11
Projection	geographic
Horizontal resolution	258 pixels
Vertical resolution	172 pixels

Table 3.4: ”Cloud Mask - MSG - 0 degree” [54] data dimensions and preprocess settings used.

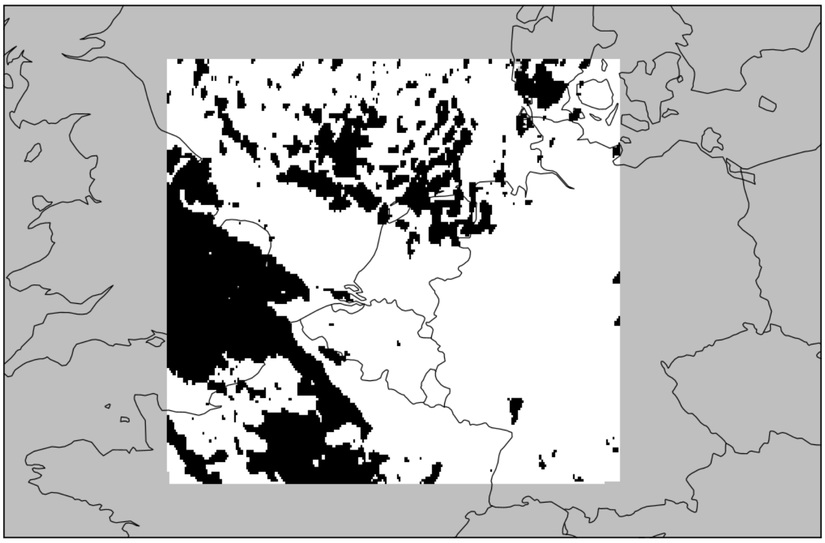


Figure 3.1: An image taken from the cloud mask dataset taken on January 5th at 11:00 with country outlines overlaid on top. Here the black shaded area represents the cloud cover and white no cloud cover, the greyed out areas are outside the longitude and latitude range of the dataset.

3.1.4 Validation Datasets

To validate the performance of the model, two additional datasets were downloaded. Their purpose is to give context to the observed power measurements. The first dataset [55] consists of statistics derived from measurements made by a nearby weather station called “Twenthe” and is maintained by the KNMI. As researched in [15, 40], the correlation of meteorological measurements from weather stations with PV-panel power generation is negatively impacted by the distance between them. But in this case the distance between the nearest weather station and the solar installation is 4.5 km, so well within significance according to their results. Information about the weather station and data can be found in Table 3.5, its weather variables can be found in Table 3.6. The data in the dataset are hourly statistics, as such the data is not highly correlated in time with the power measurements and not useful for correlation analysis. It can however be used to divide the power forecasts into groups with similar circumstances, as the data does describe the general weather type observed at that time.

Next to meteorological measurements, the CAMS solar radiation time-series made by Copernicus [56] is useful to analyse a model’s performance based on irradiance information. The CAMS solar radiation dataset is derived from local weather station data and satellite observations with a temporal resolution of 1 minute, meaning correlation analysis is useful according to [40]. The irradiation variables and their specification can be seen in Table 3.7.

Dimension	**Values**
Time window	01/01/2022 - 31/12/2022
Temporal resolution	1 hour
Latitude	52.274
Longitude	6.891

Table 3.5: KNMI automatic weather station ”Twenthe” data specification

Variable	Description
DD	Wind direction
FH	Hourly mean wind speed
FF	Mean wind speed past 10 minutes
FX	Maximum wind gust
T	Temperature
T10N	minimum temperature past 6 hours
TD	dew point temperature
SQ	Sunshine duration
Q	Global radiation
DR	Rain duration
RH	Rain amount
N	Cloud coverage
U	Relative Humidity
M	Mist occurrence (Yes/No)
R	Rain occurrence (Yes/No)
S	Snow occurrence (Yes/No)
O	Thunder occurrence (Yes/No)
Y	Ice formation occurrence (Yes/No)

Table 3.6: KNMI automatic weather station measurement variables

Dimension	Values
Time window	01/01/2022 - 31/12/2022
Temporal resolution	1 minute
Latitude	52.239891
Longitude	6.852906
Variables	TOA, (Clear sky) GHI, (Clear sky) BHI, (Clear sky) DHI, (Clear sky) BNI

Table 3.7: CAMS all-sky irradiation

3.2 Model

The model framework to create the models used in this thesis are built from five possible components that can be connected together to form the complete model. There are four input components that each process a unique input, and one output component that produces a quantile forecast from the results of these input components. The structure of this framework can be seen in Figure 3.2 together with the input and output shapes of the model. Each of the four input components create a list of features from their input. The list of features that each component produces are discussed separately in their own sections. These lists of features made by the components are then joined together and are processed by the last component such that a quantile forecast can be made.

With this framework, it is possible to remove certain input components from the complete model. Now, multiple input combinations can be made to see how much information each input component contributes to the overall forecast, without many changes that need to be made. Each component consists of a neural network and are all described in detail in the following sections. Each component has their own section in which their design considerations and overall structure are discussed, together with the hyperparameters that need to be tuned for this component.

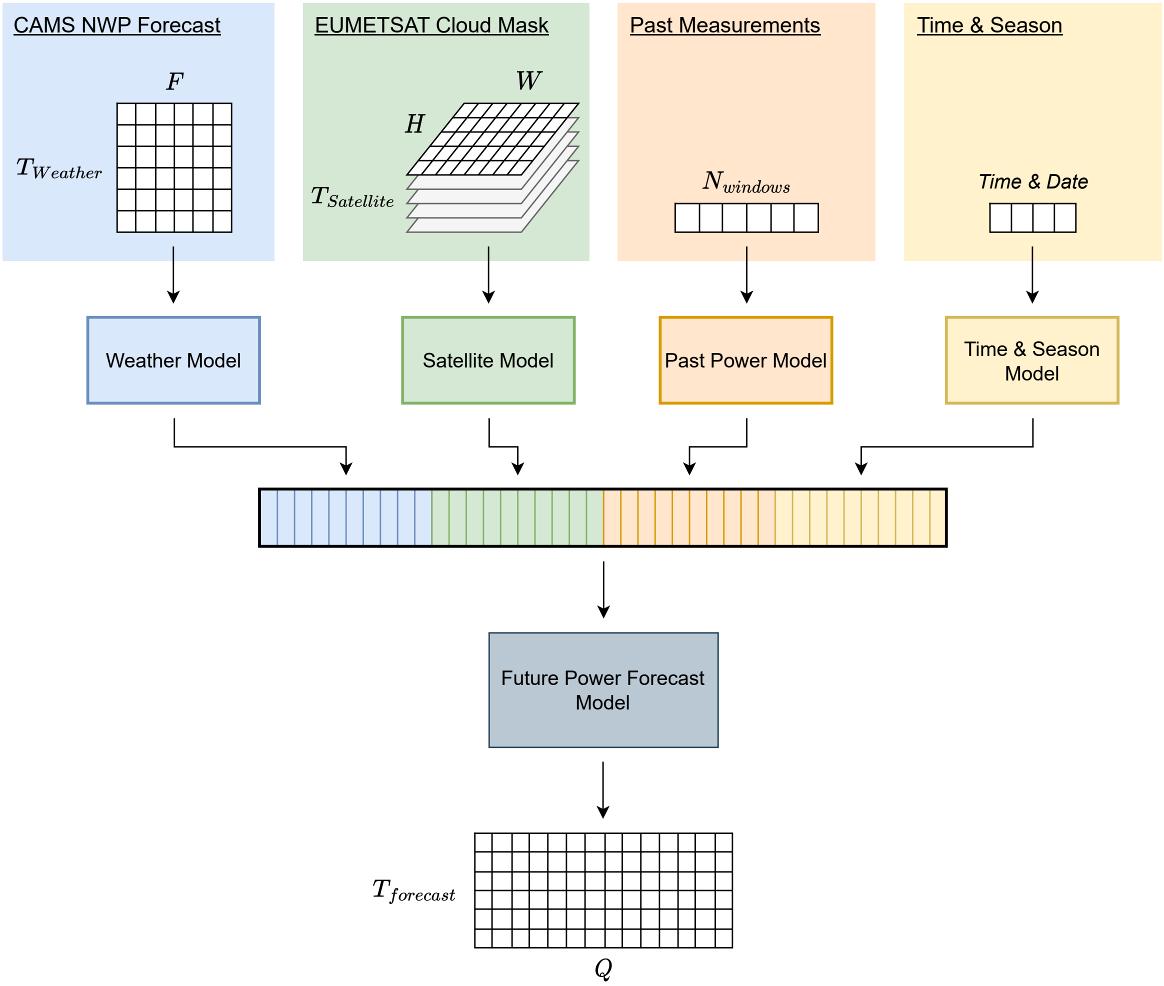


Figure 3.2: Graphical overview of the model's subcomponents, the model's inputs and outputs, and how they are connected.

3.2.1 Past Power Model

For this model, the architecture and its performance is largely dependent on how the past power measurements are measured and used. It depends on the sampling interval T_{sample} and the time period T_{span} that describes the power measurements p_t that are taken into account and spans the samples $\{p_i \mid t - T \leq i \leq t\}$ at time t. Using these samples directly as an input is however not efficient, as not every sample on its own carries much information. Therefore, it is more useful to reduce the samples to statistical descriptions, like the average power of past samples. However, defining the most effective number of elements and how they are distributed over time is not trivial. There are many ways to compute statistics from a time interval T_{span} into N unique elements, so to limit the amount of combinations to try out three windowing strategies and two time distribution algorithms were thought of to find a resource efficient model with computational complexity and performance in mind.

The two time distribution algorithms used in this thesis distribute at most N_{max} points in time in the time interval T. This is done by either linearly spacing the points such that each point is $\frac{T}{N_{max}}$ away from the other points in time, or by exponentially decreasing the distance from the current time t, starting with its largest possible distance T_{span}. The two algorithms are described in Algorithm 1 and 2 respectively. As the power measurement dataset has a limited temporal resolution of 10 seconds, the value for point t_i needs to be rounded to the nearest 10 second sampling interval. Any of the rounded points that are now duplicates are removed from the set. This leaves a set of $\{t_1, ..., t_N \mid t - T \leq t_i \leq t\}$ points in time with $0 < N \leq N_{max}$ points that can be used.

Algorithm 1 Constant decay

```
$i \leftarrow N_{max}$
while $i \geq 1$ do
    $t_i \leftarrow \frac{i}{N_{max}} T_{span}$
    $i \leftarrow i - 1$
end while
```

Algorithm 2 Exponential decay

```
$i \leftarrow N_{max}$
$t_i \leftarrow T$
while $i > 1$ do
    $t_{i-1} \leftarrow \frac{2}{3} t_i$
    $i \leftarrow i - 1$
end while
```

Now, the question of how to summarize the data covered by these points in time remains. Here, three options are defined. The first option reduces the samples between the current time t and $t - t_i$ to its average power $\bar{p}_i$ and is called the *'overlap'* option as its time window overlaps with the time windows of the other points. Whereas the *'separate'* option reduces the samples between two successive points in time $t - t_i$ and $t - t_{i-1}$ to its average power $\bar{p}_i$ and does not overlap with the time windows of the other points. The last option, *'single'* takes a single measurement p_{t-t_i} as its representative value $\bar{p}_i$. The combination of these three options with the two interval spacing algorithms can be seen in Figures 3.3 and 3.4 for constant spacing and exponential spacing respectively.

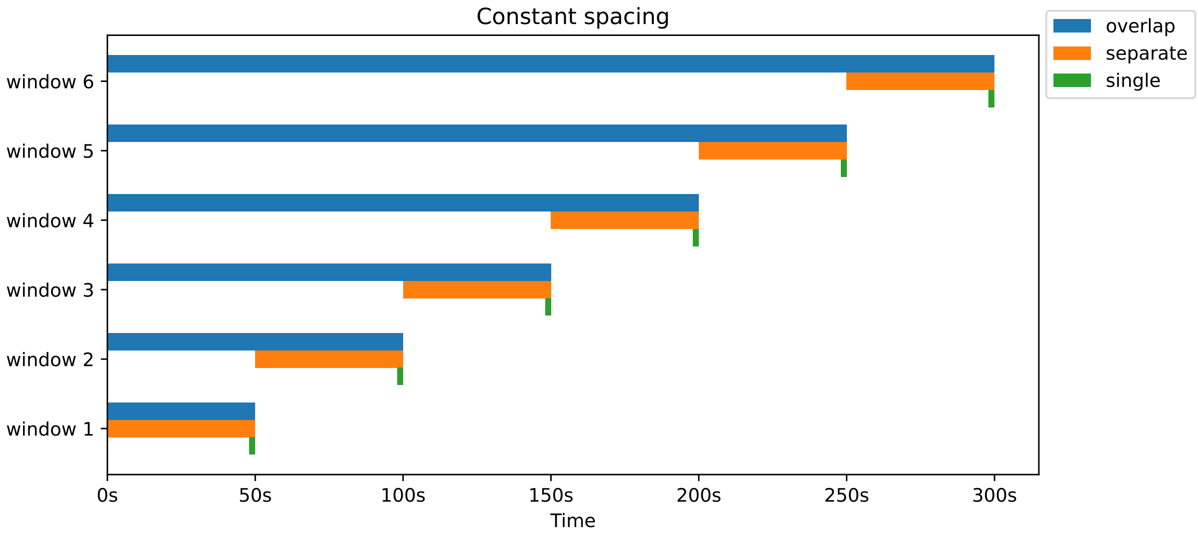


Figure 3.3: Distribution of windows with the constant decay algorithm for a time period T_{span} of 300 seconds and the number of elements N_{max} set to 6.

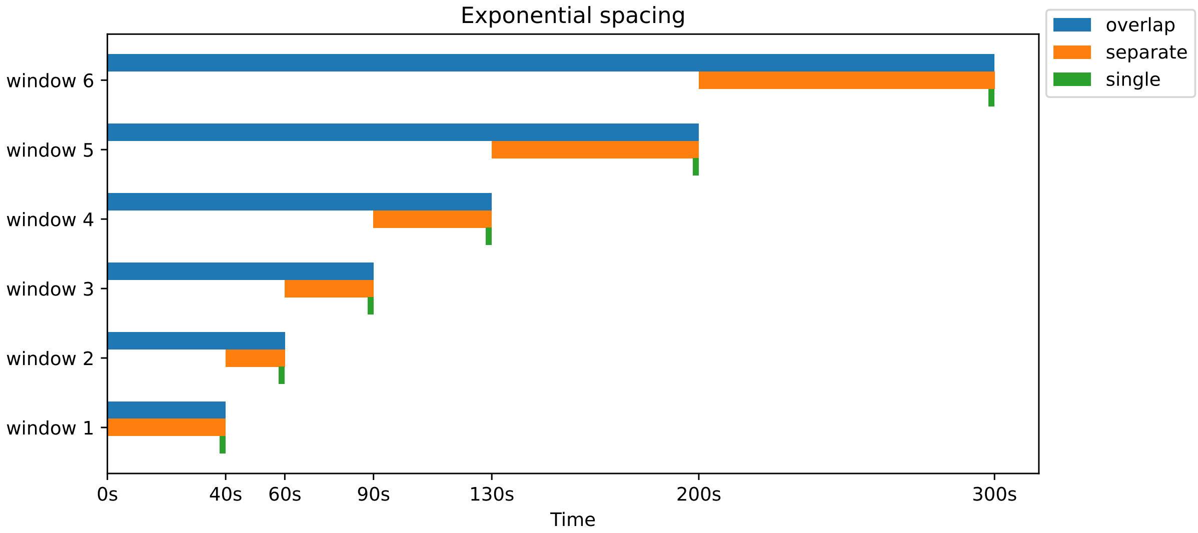


Figure 3.4: Distribution of windows with the exponential decay algorithm for a time period T_{span} of 300 seconds and the number of elements N_{max} set to 6.

The obtained values are the inputs to a neural network with two hidden layers and N units with sigmoid activations, this network can be seen in Figure 3.5. The output of this model then gets used by the main dense network for further processing. The complete list of hyperparameters that need to be tuned for the Past Power model are listed in Table 3.8.

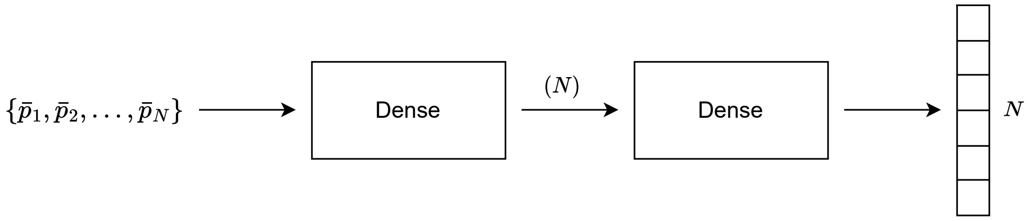


Figure 3.5: Graphical representation of the Past Power model component.

Hyperparameter	**values**
Window type	*'overlap'*, *'separate'*, *'single'*
Window step	*'constant'*, *'exponential'*
Max number of elements	N_{max}
Time period	T_{span}
Number of output features N_P	N

Table 3.8: Hyperparameters values under consideration for the past power model.

3.2.2 Time & Seasons Model

The goal of the time and seasons component is to allow the model to learn a naive representation of the expected irradiance given the time of day or year at that point time. The path that the irradiance takes over a day is similar for cloudless days and is the inspiration for the clear sky model. To give the model a similar input without providing actual clear sky estimates, the time of day and year can be used instead. In order to make a time and date suitable for machine learning, the dates and time have to be mapped to scalar values while keeping the cyclical relation of irradiance with respect to time. To do this, the time of day is mapped along a unit circle, where one full rotation represents the passing of 24 hours. The time of day can then be represented by the sine and cosine components associated by that point in time. The same strategy can be employed for the time of year, where one full rotation along the unit circle represents the passing of a year. These 4 components together describe the date and time of day, with each component's values within a $[-1, 1]$ range. A graphical illustration of this process is shown in Figure 3.6.

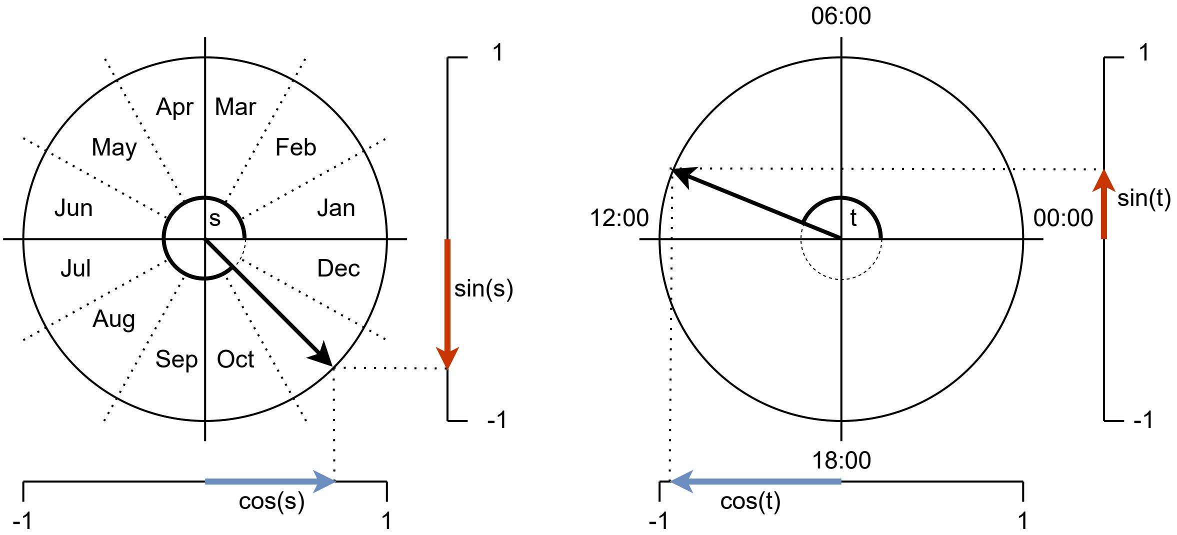


Figure 3.6: Decomposition of time and date used in the Time & Seasons model.

These 4 components are the inputs of a neural network with two hidden layers and 10 units each and is shown in Figure 3.7. Both layers have the hyperbolic tangent function as its activation function to keep the values in a $[-1, 1]$ range and allow the model to learn specialised representations of time. This model has no hyperparameters to optimize for and will be the same for all model configurations.

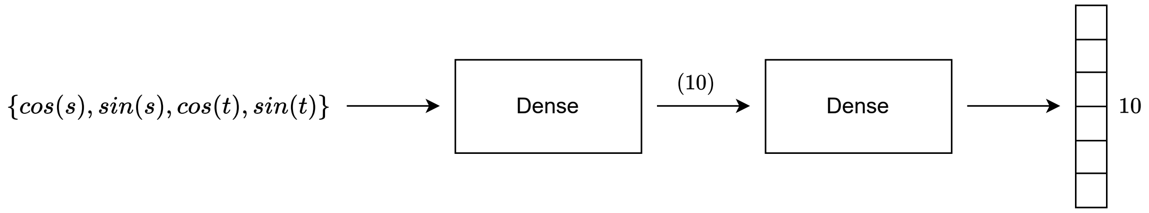


Figure 3.7: Graphical representation of the Time and Season model component.

3.2.3 Weather Model

The steps needed to find a good performing weather model with as little resource usage as possible are two-fold. First, it would be preferred if the number of features F used by the model could be reduced with as little impact to performance as possible. To do this, the LassoNet architecture and algorithm described in [47] is implemented and analysed to find an appropriate feature subset, the details on how LassoNet is implemented is discussed in Section 3.2.3.1. Next to this, a clustering algorithm proposed in [44] is implemented to see if the weather information can be generalised further into clusters types. The clustering of weather data was recommended by papers discussed in Section 2.3.2.2 to generalise the data and prevent overfitting. This algorithm is explained in Section 3.2.3.2. The architecture of the model and its hyperparameters are discussed in Section 3.2.3.3.

3.2.3.1 LassoNet

LassoNet [47] is designed to bring lasso regularization to non-linear neural network models. Lasso regularization reduces the magnitude of a features weights after each training epoch with a value of λ, such that eventually most irrelevant features contribute nothing to the output of the model and only useful features remain. The neural network required by LassoNet is a residual network. A residual network is a network consisting of two parts, a network with any number of hidden layers and a residual layer which bypasses the hidden layers and directly connects the input of the network to the output of the network. This architecture can be seen in Figure 3.8, where LassoNet is used on the weather forecast data directly.

The training process is comparable to the standard gradient descent algorithm used for training neural networks, but with a few extra steps. This process is described in Algorithm 3 and in essence constrains the magnitude of the first layer's weights W to the magnitude of the residual layer's weights θ, whilst consistently reducing the magnitude of the residual layer's weights after each epoch. The algorithm needs a few parameters to function: λ_0 is the lasso penalty coefficient the process starts with and increases exponentially with a factor ϵ each training epoch. The hierarchy multiplier M balances the influence of the linear residual layer's weights θ and the hidden network's weights W on the significance of a feature, $M = 0$ would mean only the residual layer will contribute to the output of the model, and $M \rightarrow +\inf$ would mean the hidden network is unconstrained by the residual network. The weights of the residual and the first hidden layer are constrained using the *Hier-Prox* algorithm, which is described in Algorithm 4.

Algorithm 3 LassoNet training

```
$f \leftarrow N_{Features}$
$i \leftarrow N_{epochs}$
$\lambda \leftarrow \lambda_0$
while $i > 1$ and $f > 0$ do
    Increase regularization factor:
    $\lambda \leftarrow (1 + \epsilon)\lambda$

    Perform back-propogation:
    $\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta, W)$
    $W \leftarrow W - \alpha \nabla_W L(\theta, W)$

    Perform regularization using Hier-Prox:
    $(\theta, W) \leftarrow$ Hier-Prox$(\theta, W, \alpha\lambda, M)$

    Count number of non-zero features:
    $f \leftarrow \sum_{j \in \theta} \mathbb{1}\{\mid \theta_j \mid > 0\}$
    $i \leftarrow i - 1$
end while
```

Algorithm 4 Hier-Prox

```
procedure HIER-PROX$(\theta, W, \lambda, M)$
    Iterate over every feature in the input:
    for $j \in \{1, ..., F\}$ do
        $W_j^{sorted} \leftarrow$ sort$(\mid W_j \mid)$        ▷ Sort values in $W_j$ from largest to smallest
        for $m \in \{1, ..., F\}$ do
            $s_m \leftarrow \mid \theta_j \mid + M \cdot \sum_{i=1}^{m} W_{j,i}^{sorted}$        ▷ Cumulative sum of weights up until $m$
            $w_m \leftarrow \frac{M}{1+m \cdot M^2} \cdot$ sign$(s_m) \cdot$ max$(\mid s_m \mid - \lambda, 0)$        ▷ Reduce magnitude of weights
        end for
        Find the first index where $w_m$ becomes greater than $W_{j,m}^{sorted}$:
        $\tilde{m} \leftarrow \sum_{m \in \{1,...,F\}} \mathbb{1}\{W_{j,m}^{sorted} \geq w_m\}$

        Scale and clamp $\theta_j$ and $W_j$ to $w_{\tilde{m}}$:
        $\tilde{\theta}_j \leftarrow \frac{1}{M} \cdot$ sign$(\theta_j) \cdot w_{\tilde{m}}$
        $\tilde{W}_j \leftarrow$ sign$(W_j) \cdot$ min$(w_{\tilde{m}}, \mid W_j \mid)$
    end for
    return $(\tilde{\theta}_j, \tilde{W}_j)$
end procedure
```

3.2.3.2 K-Sparse Clustering

K-sparse clustering [44] works as an extension to a standard dense network and modifies the output produced by the dense network to only let the highest k activations through and sets the others to 0. This procedure is shown in Algorithm 5, where K corresponds to the width of the output and is the number of clusters or features that can be found, k is the number of activations that are allowed to progress. During training, only the selected activations are used when back-propagating the error, the other weights are left untouched. This allows the network to only adjust for the errors caused by this cluster, but can mean that some activations in the vector will never be used and trained if they never reach the activation threshold required to be in the top k activations. To circumvent this, the value for k is set to K at the start of training, such that each activation will be a part of training, k will then slowly count down during training till at the end of training k will be 0. By saving snapshots of the model during training whenever k changes, the best value for k can be chosen as a trade-off between accuracy while limiting the number of activations.

Algorithm 5 K-Sparse

```
$\boldsymbol{z} \leftarrow \boldsymbol{W}^T\boldsymbol{x} + \boldsymbol{b}$
$\hat{\boldsymbol{z}} \leftarrow$ sort($\boldsymbol{z}$)
for $i \in \{1, ..., K\}$ do
    if $z_i \geq \hat{z}_k$ then
        $\tilde{z}_i \leftarrow z_i$
    else
        $\tilde{z}_i \leftarrow 0$
    end if
end for
```

3.2.3.3 Model Architecture

The Weather model used for this thesis is show in Figure 3.8. The input to the model is a two-dimensional matrix with shape $F \times T$, where F is the number of features used by the model from the weather forecast for T time steps into the future. First, a LassoNet network is used, which is described in Section 3.2.3.1. This network is applied T times for each forecast time step in the input weather forecast, that way each time step shares the same set of found features. The LassoNet network outputs F transformed features for each time step, a total of $F \times T$ transformed features. After the LassoNet an LSTM layer is used to find patterns in the data, the number of LSTM cells in this layer is equal to the number of features used and will keep the same shape as its input. For clustering, the output of the LSTM can then be fed into a dense network that can be used by the k-sparse algorithm discussed in Section 3.2.3.2 to select the k highest activations in the output at each time step. This is then flattened to a one-dimensional array to align with the outputs of the other models, totalling $K \times T$ features in the case of clustering, or $F \times T$ features without clustering.

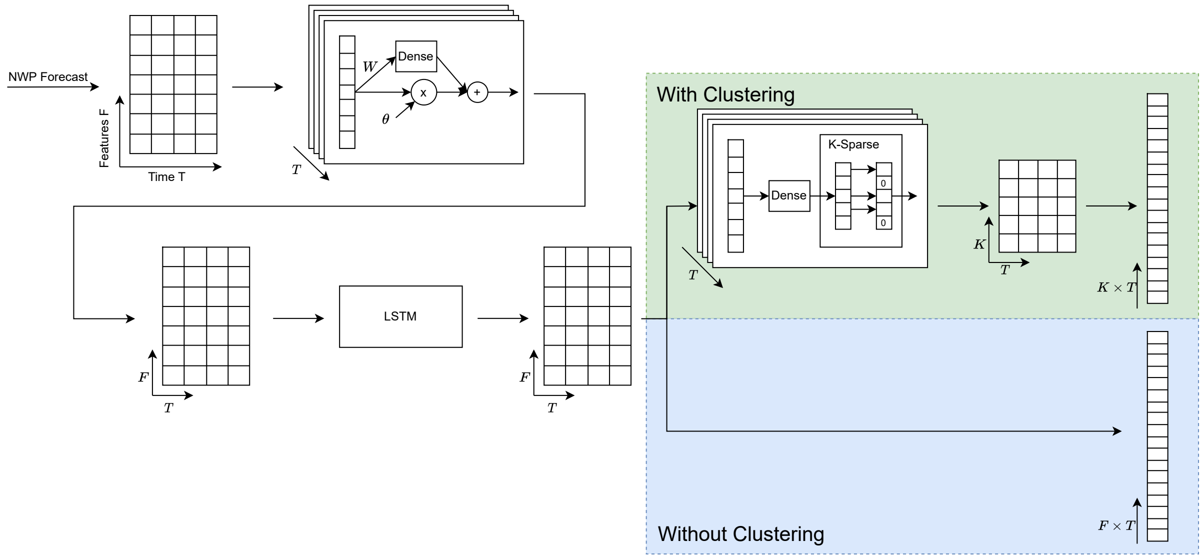


Figure 3.8: Graphical representation of the Weather model component.

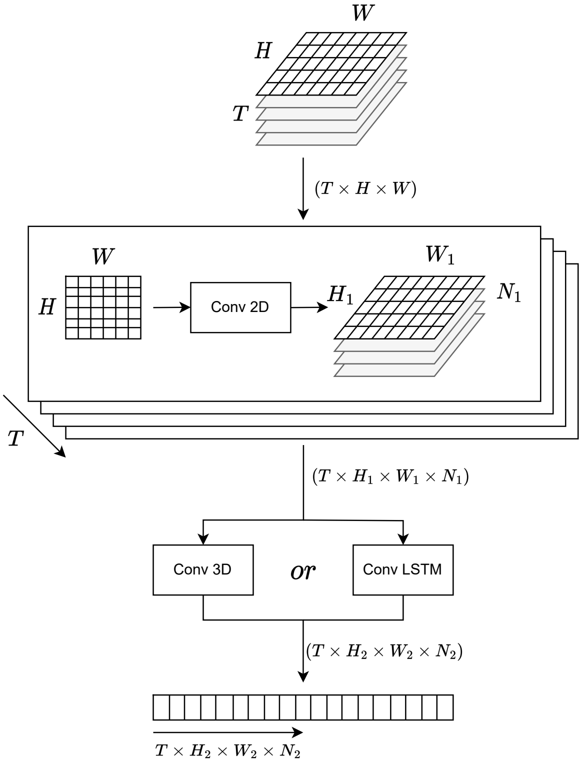
The Weather model has quite a lot of hyperparameters to optimise. Therefore, the model is first optimised without k-sparse clustering to find the most significant features. After which the model is downsized to only use these features and trained again with k-sparse clustering but without the LassoNet weight reduction process active. The complete list of hyperparameters is shown in Table 3.9.

Hyperparameter	**values**
Hierarchy coefficient	M
Lasso penalty coefficient	λ_0
Lasso penalty multiplier	ϵ
K-sparse width	K
K-activations	$0 < k \leq K$
Number of output features N_W	$K \times T$ or $F \times T$

Table 3.9: Hyperparameters values under consideration for the Weather model.

3.2.4 Satellite Model

The satellite model is inspired by [57], which focuses on spatial-temporal problems with quantile regression output. This is similar to the problem tackled here, as the goal of this component is to forecast the influence of cloud movement on PV-panels power generation, based on past satellite images. In [57] they use two Convolutional LSTM layers in series with a dense network to forecast quantiles for a single time step for each pixel in the input. That is different from what is needed for this model, as only one PV-panel is used and the model needs to predict for multiple time steps in the future. This model has been adapted to fit the needs of this thesis by replacing the first Convolutional LSTM by a standard convolutional layer which is applied to each time step and a Convolutional LSTM or a 3D Convolutional layer which returns its state for all time steps. This architecture is shown in Figure 3.9. The hyperparameters used in the Satellite model are all depended on the dimensions of the satellite images used and are listed in Table 3.10. Image zoom represents the downsampling factor, allowing the model to cover a larger area with fewer pixels as its input, reducing its resource footprint.



W
H
T
$(T \times H \times W)$
W
H
Conv 2D
W_1
H_1
N_1
T
$(T \times H_1 \times W_1 \times N_1)$
Conv 3D
or
Conv LSTM
$(T \times H_2 \times W_2 \times N_2)$
$T \times H_2 \times W_2 \times N_2$

Figure 3.9: Graphical representation of the Satellite model component.

Hyperparameter	**values**
Image zoom level	z
Image width/height	$0 \leq W/H \leq \frac{W_{max}/H_{max}}{z}$
Number of past images	T
Number of filters	N_1, N_2
Kernel size	k_1, k_2
Strides	s_1, s_2
W_1/H_1	$\left\lfloor \frac{W/H - k_1}{s_1} \right\rfloor + 1$
W_2/H_2	$\left\lfloor \frac{W_1/H_1 - k_2}{s_2} \right\rfloor + 1$
Number of output features N_S	$T \times H_2 \times W_2 \times N_2$

Table 3.10: Hyperparameters and architecture definitions for the Satellite model.

3.2.5 Future Power Forecast Model

For the Future Power model, a neural network is used with a quantile forecast as its output. The shape of the model is defined by the number of quantiles to predict for (Q) and the number of time windows it should predict (T). The architecture of this model component is shown in Figure 3.10. First, the concatenated outputs $(N_W + N_S + N_P + N_T)$ of the previously listed input components are fed through two dense layers with 60 units each and function as intermediate layers to the quantile forecast. This allows the model to learn complex relationships in the combined date. The quantile forecast network consists of two layers, the first layer is a dense network that outputs $T \times Q$ values. This output is then sliced up into T parts, which the next layer then uses to predict Q quantiles. This way, the quantile forecasts are created the same way for all time windows in the forecast and encourages the model to find a generic representation to produce quantiles with, this reduces the chance for the model to over-fit and is less likely for the quantiles to cross. The architecture is inspired by the MCQRNN architecture [50] discussed in Section 2.4.2, which tries to establish a connection between quantiles to ensure monotonically increasing quantiles. Predicting a multi-horizon forecast in one go with shared weights has other benefits, compared to forecasting each time step separately or iteratively. As [58] notes that the direct multi-horizon forecasting approach is overall less biased, more robust, and does not suffer from error accumulation.

The Quantile loss described in Section 2.4.3 is used as the main loss function of the model in combination with two additional loss functions to constrain the model. The first additional loss function penalises forecasts with quantile crossing, incentivising monotonically increasing quantiles over improper forecasts. Checking the relations between quantiles separately is required, because Quantile Loss on its own scores each quantile independently and does not penalise quantile crossing. The second additional loss function limits the range the prediction can be in to the (0-1) range, as it is impractical to allow a PV-panel to consume power and the PV-panels cannot produce more power than its maximum.

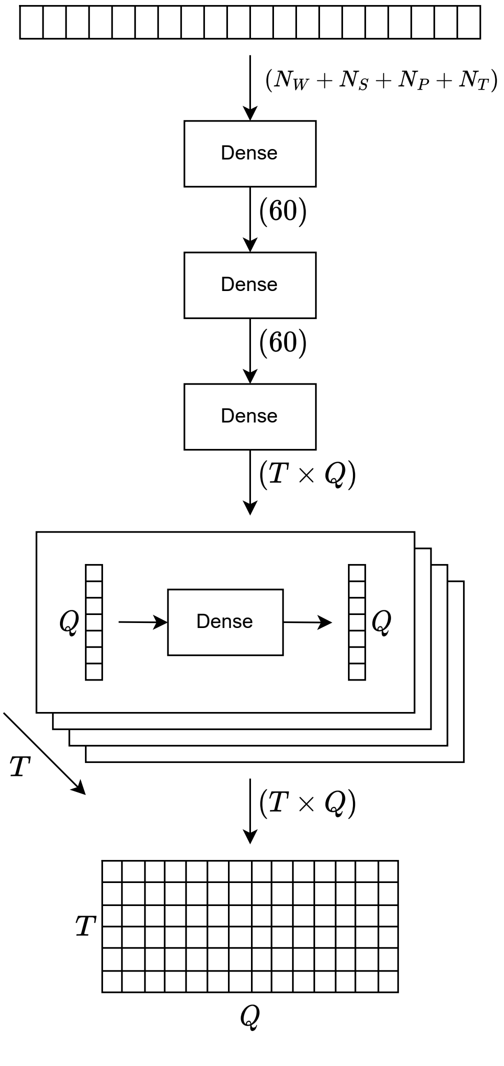


Figure 3.10: Graphical representation of the Future Power Forecast model component.

CHAPTER 4

EXPERIMENTS & RESULTS

In this chapter, the models previously defined in the methodology are implemented, and their hyperparameters are chosen. First, the datasets used for training, testing, and validation are prepared for these experiments, this process is detailed in Section 4.1. After which, the hyperparameters of all the model's subcomponents are explored separately for each component in Section 4.2 together with their final values, which are used by the final models. With these hyperparameters selected, the final models are configured and trained. These models are then compared against each other in Section 4.3 where their overall performance as well as situation specific dependencies are investigated. The models are then implemented on an embedded device to determine their practical feasibility. Next, the relation to energy management systems are explored in Section 4.4. Finally, the results are discussed in Section 4.5.

4.1 Test Setup

Before the models can be trained, multiple datasets need to be prepared to train, test, and validate the models with. To do this, the data sources described in Section 3.1 are prepared and aligned in time to form one complete dataset. The complete dataset is divided into 6 parts, 5 training datasets and one validation set, following the k-fold cross validation strategy. K-fold cross validation is used to ensure that the model that is trained is not overfitting or biased to a particular type in the dataset. By dividing the training set into k parts, or 5 parts used here, the model can be trained 5 times on 4 of the training sets and validated on the other. This procedure is illustrated in Figure 4.1 where for each iteration a new test set to validate the trained model on the other 4 training sets, the test scores can be then used to analyse the validity of the model and be used to choose the optimal hyperparameters. The performance scores should be similar for all iterations in the training process for the model to be considered a valid forecasting model which represents the overall behaviour of the data well. The validation set is left untouched until all models have been trained and is used to compare the performance of all the models.

To make the distribution of samples across the different datasets as fair as possible the dataset is partitioned by month and divided up by day, the validation set is then given 5 random days of each month and the other days per month are divided equally among the training sets. To make sure each dataset represents the different types of weather equally, the dataset is made with another fairness constraint in mind. By using the weather types provided by the weather station dataset referenced in Section 3.1.4, the shuffled datasets are compared to make sure each dataset has a somewhat equal share per type. But as these weather types are non-exclusive, and the dataset is divided by day, the datasets are not completely equal in their distribution. The resulting distribution of the datasets are shown for the weather type and cloud coverage levels in Figures 4.2 and 4.3 respectively.

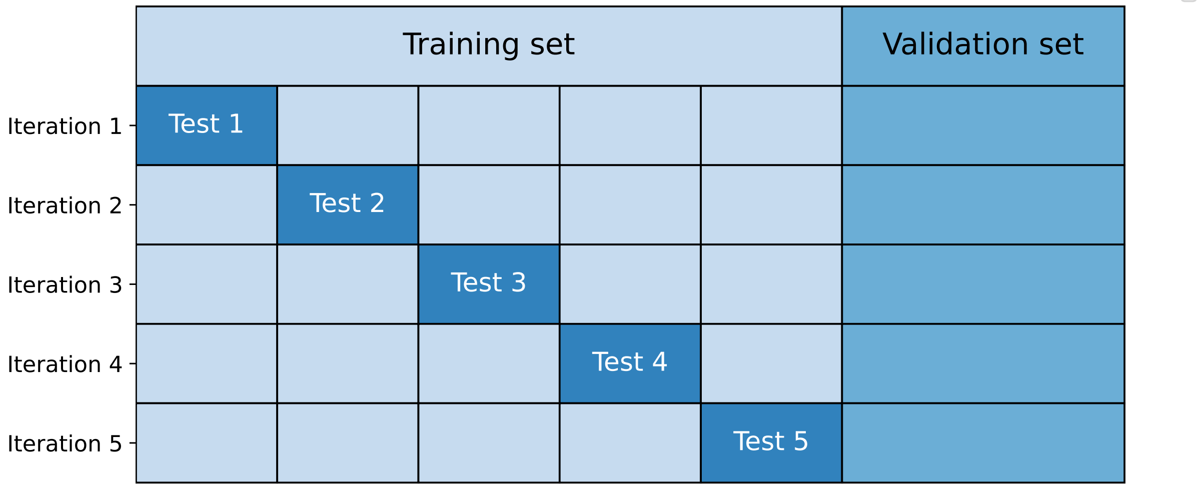


Figure 4.1: Illustration of how the dataset is split and used for each iteration in the k-fold cross validation procedure.

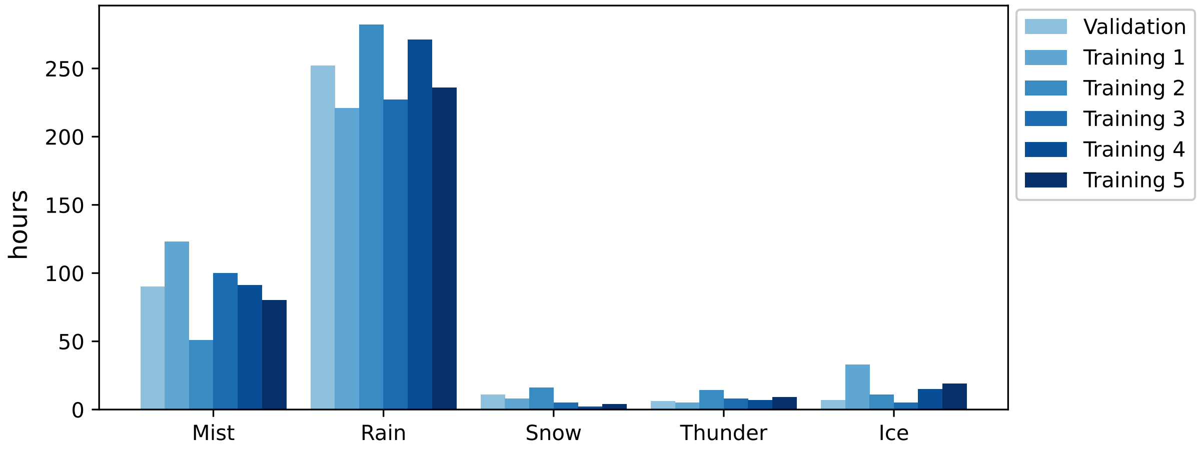


Figure 4.2: Weather types and their frequency in 2022 distribution across all datasets

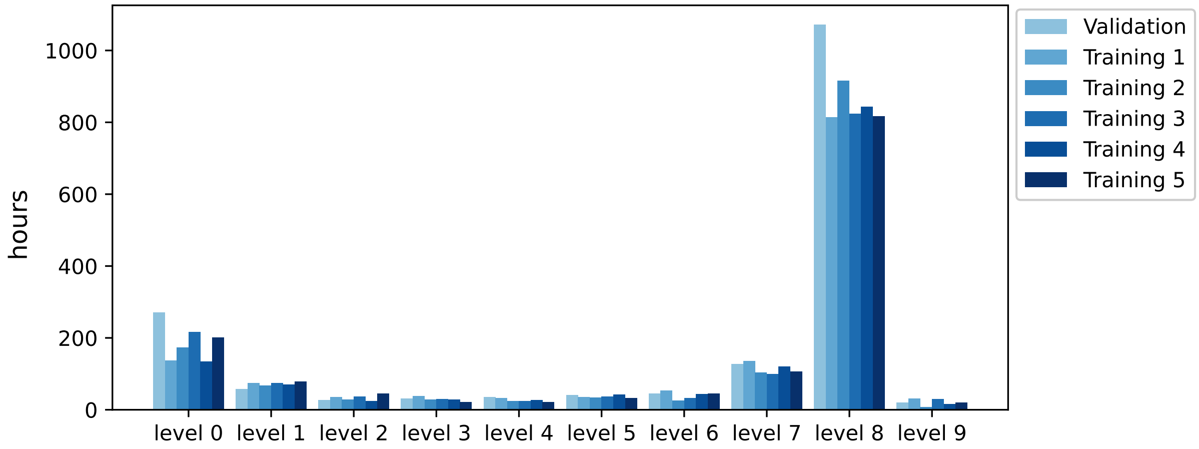


Figure 4.3: Cloud types and their frequency in 2022 distribution across all datasets. Level 0 refers to almost no cloud cover, and level 9 means total cloud cover.

4.2 Parameter Selection

In this section, the best model configurations are found by optimizing for the hyperparameters for each component discussed in Section 3.2. To limit the search space, each subcomponent is optimized separately and combined later using these found hyperparameters, the optimization processes are discussed in their own sections. The Future Power Forecast component is optimized in Section 4.2.1, the Past Power component is optimized in Section 4.2.2. In Section 4.2.3 the Weather component is optimized, where feature selection and weather clustering is performed. Lastly, in Section 4.2.5 the Satellite component is optimised.

4.2.1 Future Power

First the output needs to be defined before any model parameters can be selected as the output specification directly influences what hyperparameters are chosen. To determine the future power output windows, the power needs to be analysed first. As a start, the power has been averaged over multiple time windows to see how the power is distributed over time, the results of this analysis can be seen in Figure 4.4. Here it can be seen that the power is distributed mostly on the left side, with long tails stretching far to the right. The tails do reduce as the time window grows, which makes sense, as the outliers will average out more and take on less extreme values. But the effect of the night samples on the distribution can still be seen at the large time windows, where the mass density has shifted more to the left compared to the smaller time windows. This imbalance in power is expected, but should be noted when validating the models as the models are optimized relative to the average error, which will favour the lower power predictions.

Another analysis has been performed to see how much the power will change over time can be seen in Figure 4.4. For small time intervals the power will not change significantly from the current power production, but there are outliers which can still deviate significantly from the current power, likely due to cloud cover. The distribution becomes more spread out from the 10-minute interval onward, this is also why the persistence benchmark model is recommended by [33] for below the 5-minute interval. From then on, the tails of the distribution become more populated and the deviation becomes more significant and is where the most useful windows to predict lie. It is however more difficult to predict as the further into the future you predict the weather is more likely to deviate from its prediction. But as the time window increases, the average power becomes more stable due to the averaging and less likely to change. This does reduce the temporal resolution of the prediction, but like for the EV charging problem, the average power is still useful as long as it is accurate enough to determine its schedule on.

For EV charging based on solar energy, it is good to know if and when the forecasted power does not line up with the previous forecasts anymore. That way, a new charging schedule can be made preemptively, hopefully reducing the chance that the charging schedule will fail its promised deadlines. As such, the following time windows were selected which strikes a balance between providing enough information about the near future such that small changes in the schedule are possible and giving enough information about the farther future such that a complete schedule can be made in the first place. The windows used to predict can be seen in Figure 4.6.

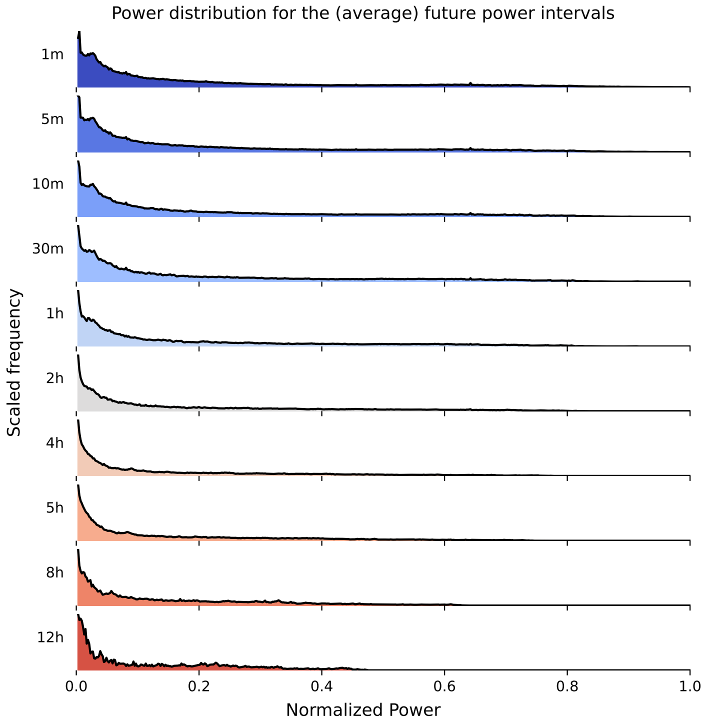


Figure 4.4: The distribution of the power production in 2022 for different time windows using the 'overlap' averaging method. Only the results where there is a non-zero power are shown here to filter out the nighttime samples, which would otherwise outweigh the daytime samples.

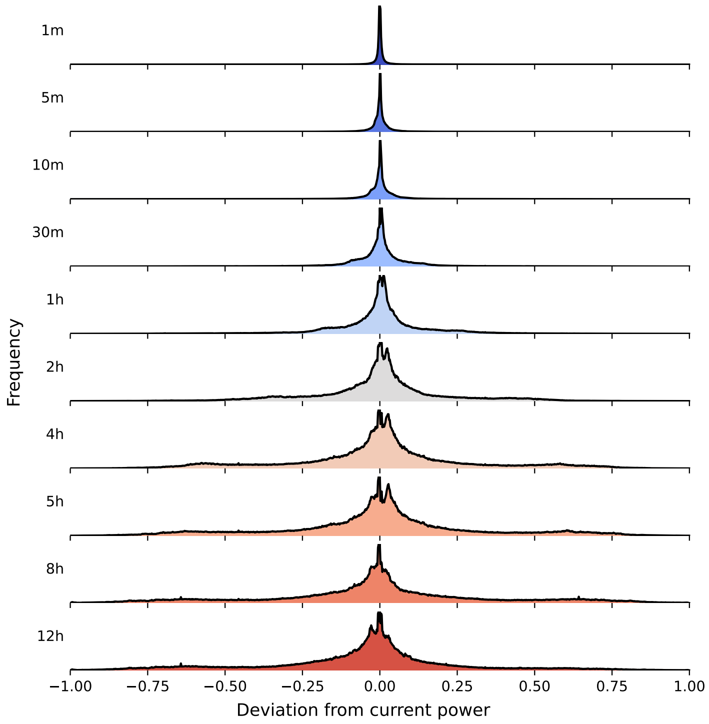


Figure 4.5: The distribution of how the power can change from its current production at time t to certain points in time of its future. The x-axis represents the distance of the current power p_t with its point in the future $p_{t+\Delta_{window}}$ using the 'single' windowing method. Similar to Figure 4.4, the samples within $|p_t - p_{t+\Delta_{window}}| < 0.002$ are excluded, as they are nighttime samples where no change happens and would otherwise overshadow the daytime samples contribution.

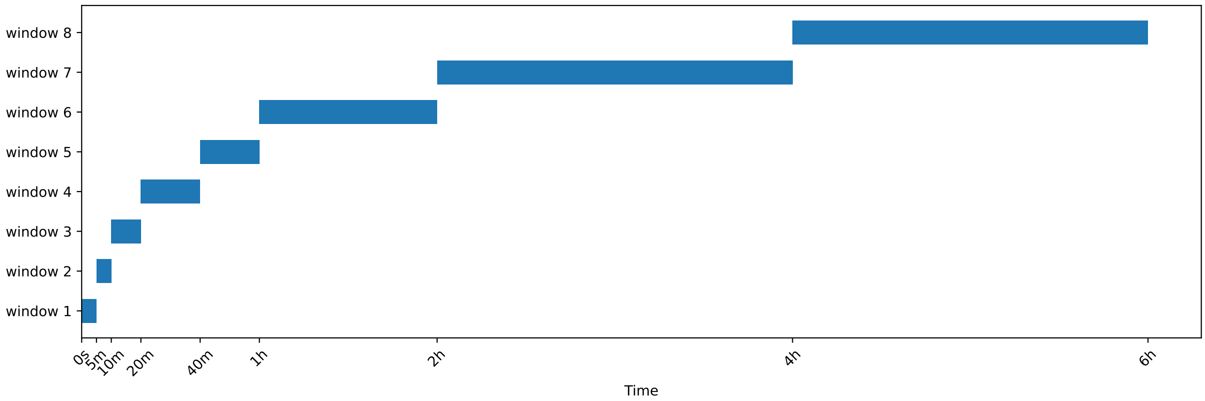


Figure 4.6: Window definitions for predicting the future average power using on the 'separate' windowing strategy. For each time window, a prediction is made that estimates the average power that will be generated in that time window. For example, the prediction corresponding to window 7 will predict the average power that will be generated from 2 up to 4 hours from now.

4.2.2 Past Power Production

As discussed in Section 3.2.1 there are 4 hyperparameters that need to be chosen for the Past Power model. The parameter search list is shown in Table 4.1. For this experiment, the Satellite and Weather models are removed so that the model is only influenced by the past power measurements and the current time of day and year. The results of this experiment are shown in Figure 4.7 where the relations between the maximum window length and the number of windows are shown. A clear relation can be seen between the maximum window length and the best performance that a model can achieve with that window length. Increasing the largest window length increases the performance of the model up until the 1-hour mark where it almost flattens off, but for resource efficiency the 1-hour window length is preferred as fewer samples need to be saved and processed.

For the number of windows used there is a less noticeable dependency relative to the best performing models, but between 8 and 12 windows there seems to be a slight improvement. From the models in $T_{span} = 600s$ and $8 \leq N_{max} \leq 12$ there is no window type or window step that dominates this list and all combinations perform approximately equal. The window length does matter as the models with the number of windows equal to 10 are in the top 5 with 4 of its 6 models. Within those 4 models the *'exponential'* and *'separate'* are most significant, but with the differences in CRPS scores the choice does not really matter. These found hyperparameters were however used to determine the final hyperparameters for the Past model and are listed in Table 4.1.

Hyperparameter	**Values**	**Final value**
Window type	*'overlap'*, *'separate'*, *'single'*	*'separate'*
Window step	*'constant'*, *'exponential'*	*'exponential'*
N_{max}	4,8,12,16	12
T_{span}	30, 60, 300, 600, 1800, 3600 seconds	600

Table 4.1: Hyperparameter values under consideration for the past power model, giving a total of $3 \times 2 \times 4 \times 6 = 144$ model configurations.

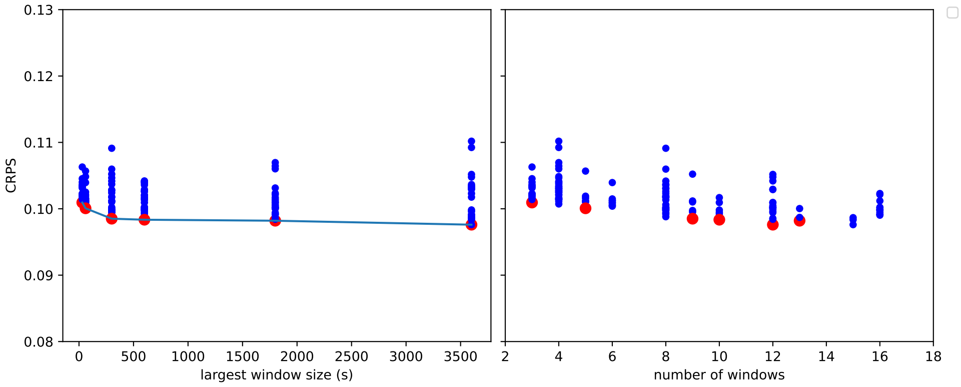


Figure 4.7: Performance scores for all permutations in the past power production hyperparameter selection experiment. On the left, the CRPS is plotted against the largest window size used for that model. On the right, the CRPS is plotted against the number of windows used for that model. The red dots represent the best performing model for each window size considered and match the red dots in the plot on the left.

4.2.3 Weather Forecast Parameter Selection

As described in Section 3.2.3 the weather model has many hyperparameters to optimize for, therefore first a feature selection process is done to find the most significant weather variables. The hyperparameters used to find these variables are listed in Table 4.2 with the values that are used. During training, the weight reduction process described in [47] will gradually decrease the contributions of the individual weights. During training the reduction becomes significant enough that a variable will no longer contribute to the model, this is monitored during the training process and after each training epoch the model is checked to see how many variables are still active and saves a snapshot of the best performing model with that number of variables for each iteration and configuration. As it is a dynamic process, there is no guarantee to how many variables will be removed during a training run or how many variables will be removed at a time.

Hyper Parameter	Values
M	2, 6, 10
λ_0	0.005, 6.67, 13.33, 20.0
ϵ	0.005, 0.01

Table 4.2: Hyperparameter values under consideration for weather forecast parameters selection. This gives $3 \times 4 \times 2 = 24$ permutations to test.

The found models are plotted against the CRPS scores on the test set in Figure 4.8. In that scatter plot, a clear trend can be seen between the number of features remaining and the models with the lower and consequently better CRPS scores, which increases as the number of features used decreases.

The varying CRPS scores per number of features used also shows a trend, with the spread in CRPS scores reducing as the number of features used also reduces. This can be explained in two ways, the first is that the training process uses the same training iteration to find multiple feature sets and save a model for each number of features used. As a result, the model found with fewer features has been trained for longer than the model with more features, as the process ensures that the number of features will always decrease and never increase. The second reason is that the ADAM optimizer can find the best performing model easier as the problem becomes less complex with each feature removed, thus reducing the chance that the model gets stuck in a local minimum. The second explanation is more likely, as most models with a large amount of features used are distributed more closely to its best performing models and the worse models are less frequent.

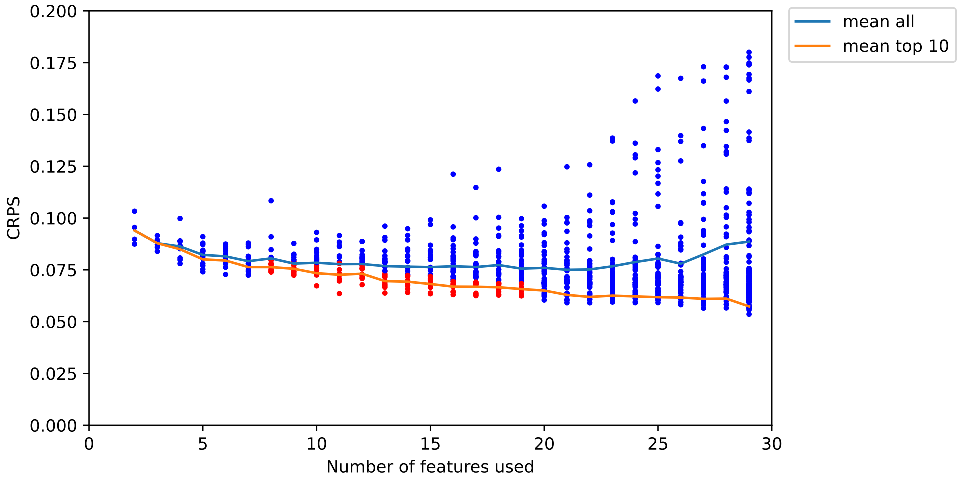


Figure 4.8: The CRPS scores of all considered weather models relative to the amount of features needed to make their prediction. Each dot represents the performance of a single model, the red dots are the best performing models that fall within the selection criteria and are used to select the best features in the weather model, the blue dots will not be taken into account.

These models each learned their own set of features that they deem significant. To come to one set of features, the best performing models are compared to see what features they chose to explain the future power generation of PV-panels. The amount of times that a feature was used in these models was counted and can be seen in Table 4.3 where the feature is sorted by how frequently it was chosen by these models, a graphical representation of this list is shown in Figure 4.9. Here, the most prominent features found in the models are the variables that are related to clouds and the water suspended in the atmosphere, both reducing the amount of sunlight received by the PV-panel.

Although irradiance variables are the most prominent variables used in solar power predictions, as discussed in Section 2.2, the only significant variable regarding irradiance found in this experiment is the DHI at place 8. This is due to the fact that most irradiance information can be reconstructed from other weather variables in combination with the Time and Seasons information, which has a higher temporal resolution compared to the weather forecast, and thus the irradiance predictions are less informative to the model as a whole. To demonstrate this distinction, the same experiment has been replicated with a weather only model where the Time & Seasons model component has been removed, everything else has been left unchanged. The results of this experiment can be found in Appendix A, where irradiance variables are more prominently featured.

The convective precipitation and total column cloud variables remain among the most important variables in both experiments. These variables were surpassed by the surface net radiation (*ssr*) variable in the weather only experiment as the most important variable, this is a stark difference to the first experiment where the *ssr* variable is ranked 19th and only used by $1/5$ of the models. Convective precipitation is regarded as the most useful variable among all precipitation variables, exceeding large-scale precipitation and total precipitation. Convective precipitation is the chaotic type of precipitation where large amounts of water can fall in a short amount of time and indicates sporadic or fast changing cloud cover. Large-scale precipitation would be more consistent over time and is also recorded in the cloud cover variables, making its overall information contribution less significant. There is no variable in the CAMS weather forecasts that describes how irregular the cloud cover can be, in the same vein that convective precipitation is related to large-scale precipitation. Convective precipitation is therefore the only variable that can give an indication on the irregularity in cloud cover.

From the results of both experiments, the variables wind, temperature, and surface pressure were all deemed negligible in their contribution. However, based on the studies listed in [13], these variables are often included as inputs to these types of models. As was discussed in Section 2.2, these meteorological variables are directly related to solar power production, but in Section 2.2.2.4 it was noted that these variables need to be measured locally as the temporal and spatial resolution of these forecasts are not representative of the actual situation.

The subset of weather features to be used by the final model are chosen based on a trade-off of how often a feature is chosen and the amount of features that would remain, as the total amount of features influences the computational complexity of the model. Therefore, the top 9 features were chosen for the final weather features used to strike a balance between accuracy and efficiency.

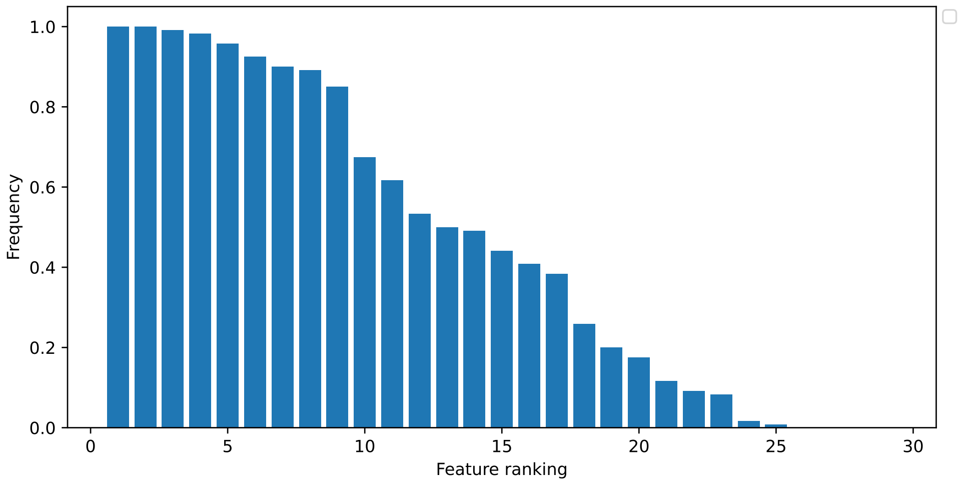


Figure 4.9: Graphical representation of the data shown in Table 4.3 as a sorted frequency graph showing how often a feature remained significant in the best performing weather models. A value of 1 means the feature was present in all of these models, and a 0 means no model kept that feature.

4.2.4 Weather Clustering

Now the model of the model found in the previous Section 4.2.3 are extended with a dense network at its output as described in Section 3.2.3. The K-sparse algorithm was implemented in the model and configured with the hyperparameters described in Table 4.4. During training the value of k is slowly reduced to 1 and for each value of k the best performing model is saved, this is recommended by the paper [44] itself to find the k-sparse activations instead of individually training a network with a fixed value for k.

The results of this experiment can be seen in Figure 4.10, in this experiment most models could not find a good representation of the weather variables to cluster the data with. This is evident as most models are distributed above a CRPS of 0.08 which is worse than the original performance of these models without clustering as can be seen in Figure 4.8. There are a few models that do perform better than their non-clustered counterparts, these are all made by a few runs where the initial models already found a balanced representation of the data which made clustering more performant. Each initial model specification had a run where a k-fold iteration found such a representation whereas the other iterations did not, it should be noted that the iteration ID is not the same for these runs and is not biased to a training or test set. This performance can be explained by looking at how each node in the output layer is used. The bar graph in Figure 4.11 shows how the best performing iteration also has learned a more balanced activation frequency over all output nodes compared to the models that did not find a similar representation. Due to the unstable nature of the model, this approach was not considered for the final model and the model defined in Section 4.2.3 is used instead.

top	Identifier	Description	Unit	Frequency
1	tciw	Total column cloud ice water	kg/m^2	1.000
2	tclw	Total column cloud liquid water	kg/m^2	1.000
3	cp	Convective precipitation	m	0.992
4	tcrw	Total column rain water	kg/m^2	0.983
5	tcslw	Total column supercooled liquid water	kg/m^2	0.958
6	mcc	Medium cloud cover	$(0-1)$	0.925
7	lcc	Low cloud cover	$(0-1)$	0.900
8	fdir	Total sky direct solar radiation at surface	J/m^2	0.892
9	hcc	High cloud cover	$(0-1)$	0.850
10	tcsw	Total column snow water	kg/m^2	0.675
11	tcw	Total column water	kg/m^2	0.617
12	cdir	Clear-sky direct solar radiation at surface	J/m^2	0.533
13	lsp	Large-scale precipitation	m	0.500
14	ssrc	Surface net short-wave (solar) radiation, clear sky	J/m^2	0.492
15	dsrp	Direct solar radiation	J/m^2	0.442
16	ssrdc	Surface solar radiation downward clear-sky	J/m^2	0.408
17	tsrc	Top net solar radiation, clear sky	J/m^2	0.383
18	tcc	Total cloud cover	$(0-1)$	0.258
19	ssr	Surface net short-wave (solar) radiation	J/m^2	0.200
20	tisr	TOA incident solar radiation	J/m^2	0.175
21	sund	Sunshine duration	s	0.117
22	ssrd	Surface short-wave (solar) radiation downwards	J/m^2	0.092
23	tp	Total precipitation	m	0.083
24	u10	10 metre U wind component	m/s^2	0.017
25	v10	10 metre V wind component	m/s^2	0.008
26	t2m	2 metre temperature	K	0.000
27	fal	Forecast albedo	$(0-1)$	0.000
28	sp	Surface pressure	Pa	0.000
29	tsr	Top net short-wave (solar) radiation	J/m^2	0.000

Table 4.3: Sorted list of weather features by how often the feature was present in the best models.

Width K	k features
5	1, 2, 3, 4, 5
10	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
15	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Table 4.4: Hyperparameter values under consideration for weather clustering. Width specifies the width of the clustering layer, and k-features the amount of sparse activations that can be active at the same time. The training loop is run 1 time for each width, while reducing the value for k steadily during training to obtain the models for each k-features. This leads to a total of $5 + 10 + 15 = 30$ model configurations.

4.2.5 Satellite

The satellite model described in Section 3.2.4 has been implemented and run with the hyperparameters listed in Table 4.5. The results as shown in Figure 4.12 shows similar behaviour to what was found when using the k-sparse clustering algorithm in Section 4.2.4 where most models would not converge to a passable performing model. This is likely due to insufficient information represented in the model on how to steer the gradient towards optimal values be-

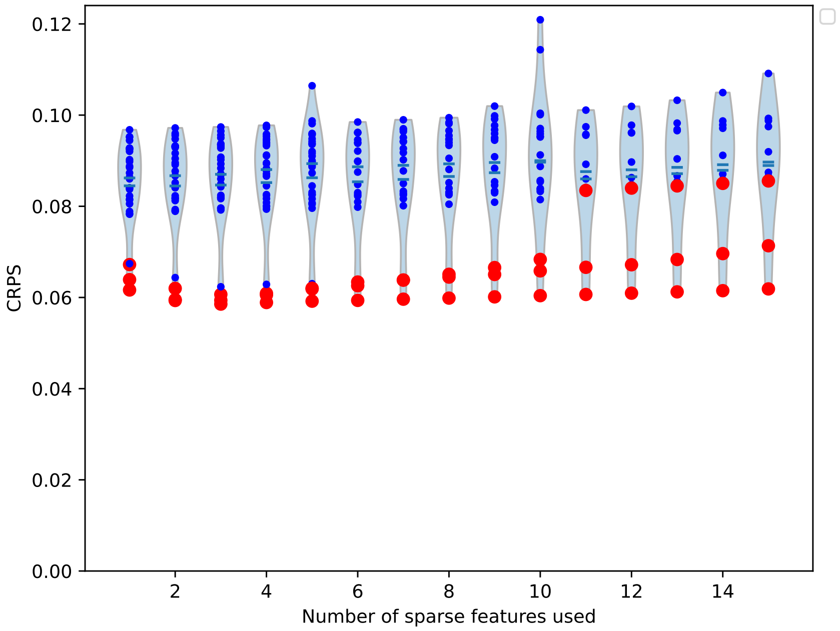


Figure 4.10: Performance of each trained cluster model with the best 3 models per number of sparse features used shown in red

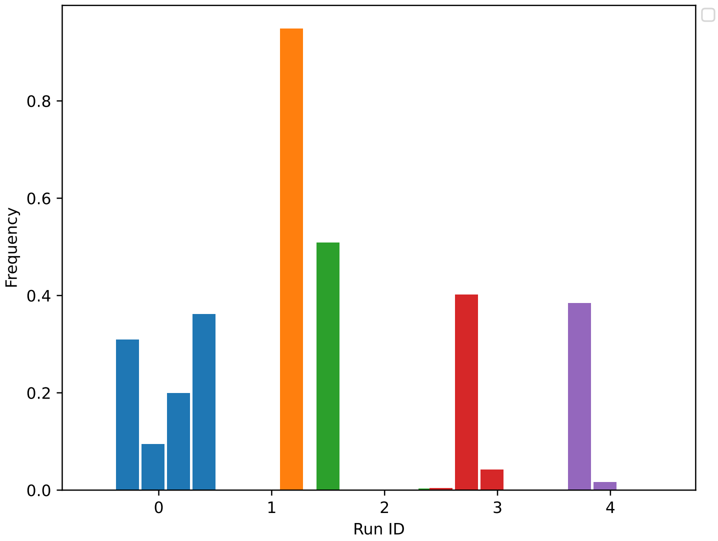


Figure 4.11: Activation frequency of the $K = 5$ wide network with $k = 2$ activations showing how the best performing model, Run ID 0, also has the most diverse activations.

cause of the sheer number of variables in the Satellite model. This can be seen in Figure 4.12 comparing the number of past images to the CRPS score, where fewer images increases the likelihood of the model converging to decent representations of the output. But there are some useful patterns that can be found in the data, as the models that cover a larger area generally perform better than a more local view. This implies that the models that were found do indeed have some understanding of how the clouds impact the power generated.

As has been shown in [33, 19] the use of satellite images is only useful in the near future. This is also the domain of the Past Power model, which converges more consistently to performant models and generally performs better with a less complex network. Therefore, the Satellite model is not considered a viable option and is removed from consideration in the final model.

Hyperparameter	**Values**
Zoom z	1, 2, 4
Image width W/ height H	32
Number of past images T	2, 3, 4
Number of filters	$N_1 = 4$, $N_2 = 10$
Kernel size	$k_1 = 5$, $k_2 = 4$
Strides s_1, s_2	3
W_1/H_1	10
W_2/H_2	3

Table 4.5: Hyperparameters used for generating the satellite model, giving a total of $3 \times 3 \times 2 = 18$ *model specifications to be tested.*

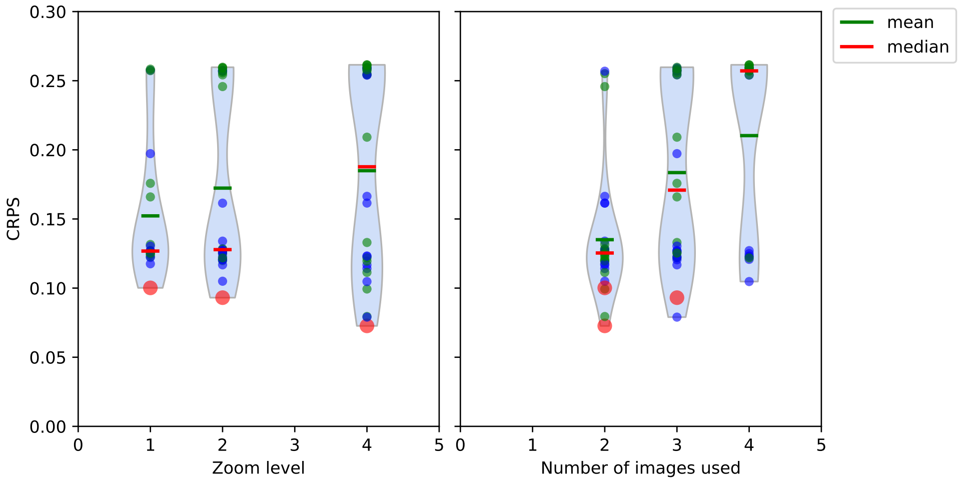


Figure 4.12: CRPS performance scores of each trained satellite model plotted against the zoom level and number of past images used. The red dots are the best performing models per zoom level, the green dots are the LSTM models and the blue dots the Conv3D models.

4.3 Results

In this section, different model combinations are evaluated to test their reliability using the validation set that has not been used so far. These models use the hyperparameters found in Section 4.2 for their respective components. Next to the neural network models, a QR model is made and will function as a baseline model to compare these models with. The QR model is trained on the same data and data dimensions used by the Time + Past + Weather model. Similar to the neural network models, the QR model is trained with the Quantile Loss function as described in Section 2.4.3. The complete list of models and model combinations that are used are listed in Table 4.6. Before evaluating the reliability of these models, they are first evaluated based on their overall performance and accuracy in Section 4.3.1. After which in Section 4.3.2, the models' reliability and accuracy are presented by evaluating their performance, these are illustrated from the perspective of multiple environmental variables that can influence the energy production of solar panels.

Combination
QR
Time
Time + Past
Time + Past + Weather
Time + Weather

Table 4.6: The final 5 models that are considered.

4.3.1 Overall Performance

4.3.1.1 Day Forecast

In this analysis, the power forecasts of the models are used to show how their output changes over a day and how the models react to different environmental changes. The models are used to generate a new forecast every 10 seconds using the real life data that would be available at that moment in time. These forecasts together with the actual average power generated at that time in the future can be seen in Figure 4.13. Here the models predict their power forecasts for June 3rd, 2022. Three of the eight time windows are shown to illustrate how these predictions change over time and how the input data influences the time windows differently. A large blue area coupled with a narrow red area means the prediction is very certain to be in the red, but can take extreme values in rare occasions. A large area with a slow transition from red to blue corresponds to a less clear view of the future.

Here it can be seen that for all models using the weather forecast as an input that their forecast changes significantly when a new weather forecast is used, this is evident from the jagged transitions between forecast on the hour mark. The influence of the weather is most notable on the 4h time window, which is further into the future, indicating the significance of the weather forecast in this time interval. For the models which use the past power measurements as an input, the forecasts follow the actual power production more closely. This makes sense as the past power measurements are the only source of data with a short and frequent update cycle.

The QR model has trouble producing forecasts with a good fit around the actual power, producing forecasts with large uncertainties in the outliers as well as significant uncertainty around the median. The same can be said for the Time model, which can only rely on seasonality and time of day. The Time + Past model also predicts large lower bounds for the outliers, but has more realistic expectations of what the upper bound would look like. The upper and lower bounds are shaped similar to the Time model, but its quantiles around the median are more densely distributed around the actual average power production for time windows in the near future. The neural networks with a weather input produce better estimates of the upper and lower bounds of values the outliers are expected to take. But for the Time + Weather model, the network has a very dense distribution of quantiles around the median quantile. Meaning that the model is confident of the expected power being near that value, but as can be seen its prediction can be significantly off the mark in the shorter time windows.

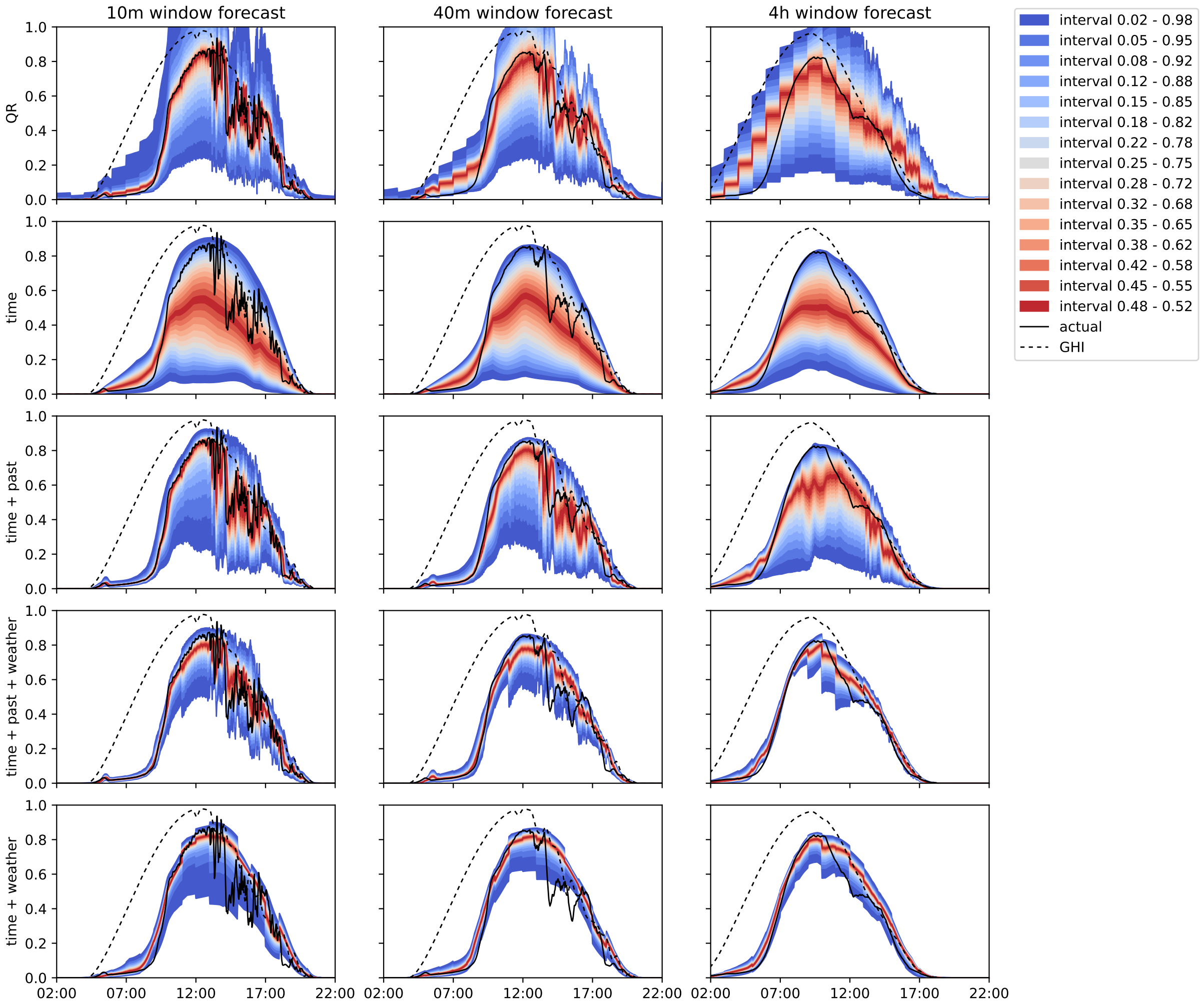


Figure 4.13: Power forecast of June 3rd, 2022 as described in Section 4.3.1.1. The predictions shown here are aligned to when the predictions were made, not the time the predicted window covers. The evolution of the actual power average for that time window is shown in black. For the predictions, the outermost quantiles are shown in blue and transitions to red for quantiles closer to the predicted median with $\tau = 0.5$.

4.3.1.2 Reliability Diagram

As discussed in Section 2.4.3, reliability diagrams are used to visualise how accurate the forecasted distributions match with their observations. For this analysis a reliability diagram is made for every model as well as every time window that that model forecasts, these diagrams can be seen in Figure 4.14. The ideal distribution is shown as a dashed black line, meaning the observed frequency matches with the probability associated with that quantile. If a line is above the dashed line, the model is generally overestimating the thresholds for that quantile and covers more samples than expected. If the line is under the dashed line, the model is generally underestimating the threshold for that quantile.

It should be noted that the reliability diagrams were made using most of the forecasts generated by the models, but not all. The samples from the evaluation needed to be filtered first, as the data included samples at nighttime which skewed the distributions in the diagram. This is because the power produced at night is almost zero, making numerical instabilities and quantiles overlapping each other due to the very dense probability distribution a significant issue. Next to that, the observed quantile frequency is counted using Equation 2.12, which relies on a less than equal comparison. This all disproportionally affects the observed frequency, as such the samples where the average future power is less than 0.01 are not considered.

In these diagrams, it can be seen that both neural networks using weather data as an input deviate significantly from their ideal distribution. Both models are overestimating the lower quantiles and underestimating the upper quantiles, meaning the distribution around the median quantile is denser than it should be. This can also be seen in Figure 4.14 where these two models forecast distributions that are very compact around the median. For the model combing weather data and the past measurements, the reliability diagram becomes more accurate with a smaller time window. At these time windows, the forecast is using mostly the information from the past measurements and less from the weather forecast. The QR model also uses weather data, but does not show the same effects.

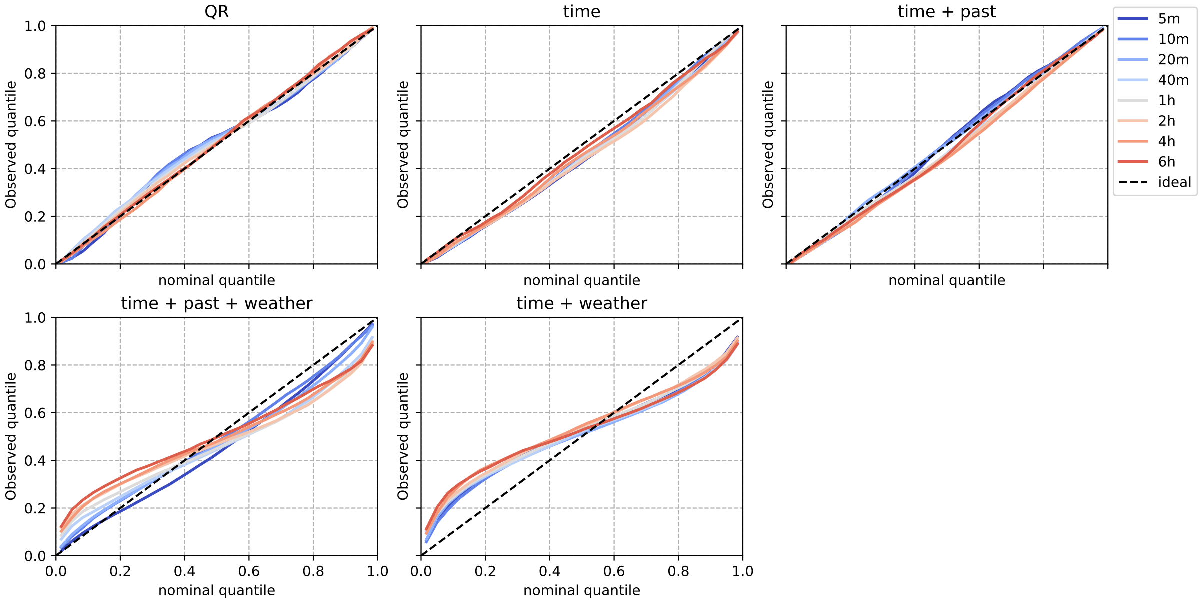


Figure 4.14: Reliability diagram for all models under consideration. The ideal distribution is shown as a dotted black line. The distribution for the different time windows are shown as a range of colours transitioning from blue to red, blue meaning windows in the near future and red windows further away.

4.3.1.3 Time Interval Performance

The next performance analysis is done by evaluating the forecast of a model using the CRPS metric defined in Section 2.4.3. The performance of each model is grouped by their CRPS for each time window in the forecast. In this analysis only daytime data is used, as otherwise the CRPS distributions would be heavily skewed to 0 as these samples will have almost no prediction error and would provide no useful information. In Figure 4.15, the CRPS is shown based on the relative frequency of that score happening. For the models with past measurements, their CRPS is significantly lower at small time windows compared to the models without. However, the neural network models with weather data perform significantly better at the longer time windows. For QR, the model performs well at small time windows, but performs significantly worse the further away the prediction is from the time the forecast is made. The neural network model combing both past measurements and weather data performs better across all time windows.

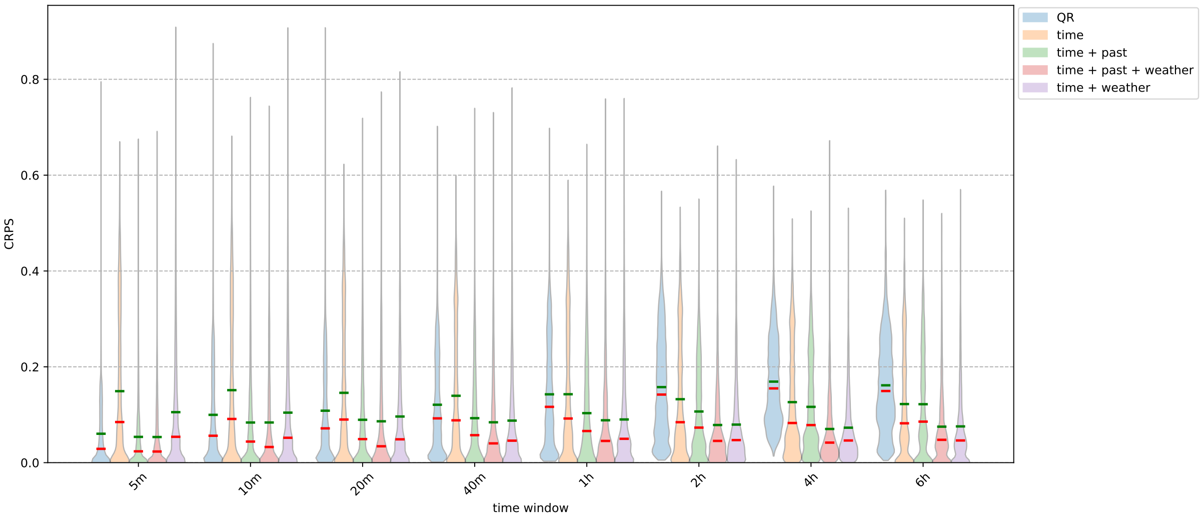


Figure 4.15: Distribution of CRPS scores based on future windows. The shapes' width indicates the relative frequency that a score is present in the dataset, a wide shape means frequent and a small width infrequent. The mean and median of these distributions are shown in green and red, respectively.

4.3.1.4 Relative Performance

For the last performance analysis, the neural networks are compared with the QR model to measure the relative performance of each model. For this two metrics are used, the first expresses how often a model was better than another and the second metric gives a score by how much a model, on average, is better. The first metric shown in Equation 4.1 is the relative frequency a model had a lower CRPS compared to the QR model for the same sample, the relative frequency is expressed as a percentage. The second metric is using the Skill Score, which calculates the relative performance increase of a model compared to another model, its definition is shown in Equation 4.2 using QR as the reference model. A Skill Score below 0 means the QR performs on average better, above 0 means the other model performs better. A score of 1.0 would mean the model made forecasts that match the actual observations perfectly.

$$RF_{CRPS} = \frac{100\%}{N} \sum_{t=1}^{N} \mathbb{1}\{CRPS(X_t^{model}, y_t) < CRPS(X_t^{QR}, y_t)\} \tag{4.1}$$

$$SS_{CRPS} = 1 - \frac{\frac{1}{N}\sum_{t=1}^{N} CRPS(X_t^{model}, y_t)}{\frac{1}{N}\sum_{t=1}^{N} CRPS(X_t^{QR}, y_t)} \tag{4.2}$$

In Figure 4.16 the results of the first metric are shown. Here it can be seen that the QR model is only competitive in the small time windows, but is outperformed everywhere else. The neural networks using the past power measurements always outperform the QR model. As for the models with weather data as input, the models perform significantly better at longer time windows.

In Figure 4.17 the results of the second metric are shown. With the results from the first metric, it can be seen that the QR model is outperformed by the neural networks using past measurements at small time windows, but not by much. For the Time and Time + Weather models the performance at these small time windows is significantly worse than QR, but do improve at longer time windows. The neural network using both past measurements and weather data either meets or improves on the neural network models missing either input source, meaning the model retains the same information from both past and weather sources as its specialised models without compromises.

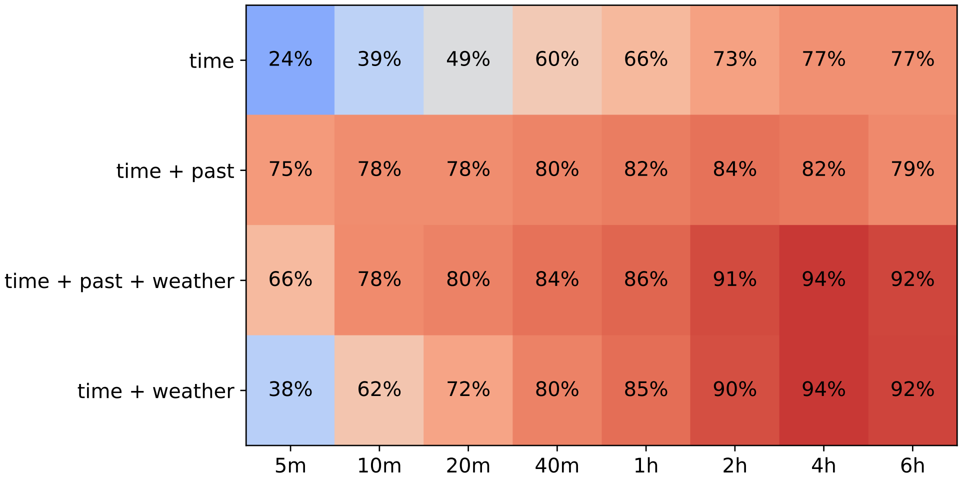


Figure 4.16: Comparison between the QR model and the other models based on how often the other model gives a better CRPS score as defined in Equation 4.1. 100% *means the other model always outperforms the QR model and* 0% *means the QR model always outperforms the listed model.*

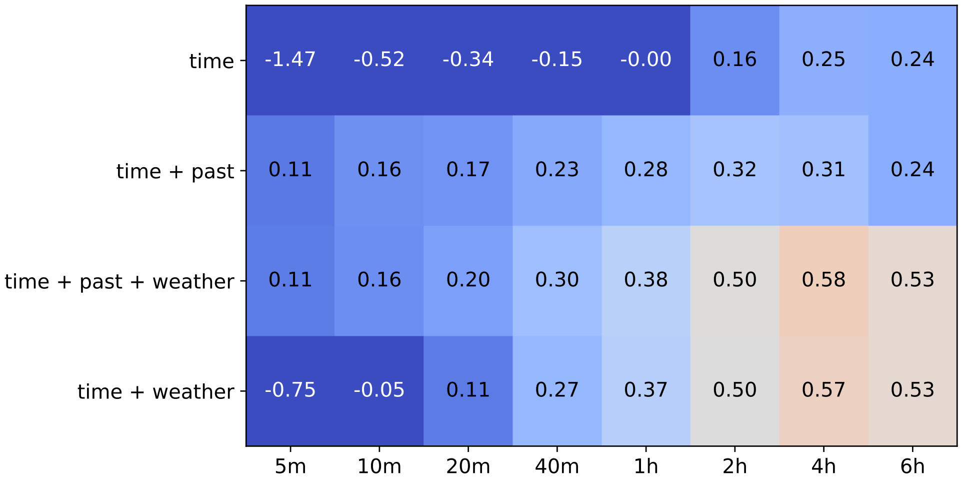


Figure 4.17: Comparison between the QR model and the other models using the Skill Score metric as defined in Equation 4.2.

4.3.2 Data Dependencies

From the background study performed for this thesis, it was concluded in Section 2.3.3 that the overall performance of a model does not accurately reflect the models'performance for all situations. Therefore, in this section, the performance of the models are looked at from different angles to discern whether a model's performance decreases in particular situations.

These perspectives should give insight into how robust and reliable a model is in real-life situations.

4.3.2.1 Seasonality

To see if the models have a seasonal dependency the models' performance is partitioned by the month the forecast was made in, this gives insight into possible seasonal influences which are reflected in the CRPS. The results of this analysis are shown in Figure 4.18. In this figure, a clear relation between sunny summer months and the more dim winter months can be seen. In the summer months, when the sun is more intense, differences in forecast and reality become more pronounced in the CRPS. This is because the CRPS is related to the absolute error, which is more sensitive to differences in scale.

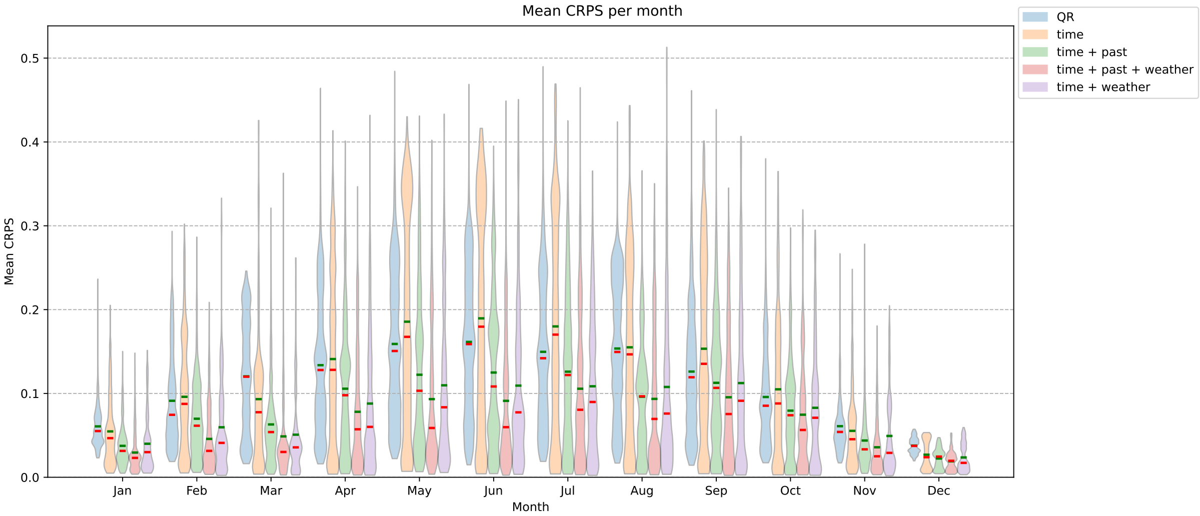


Figure 4.18: Averaged CRPS score of all forecast windows categorized by month. The mean and median of these distributions are shown in green and red, respectively.

4.3.2.2 GHI

From the seasonality analysis done in Section 4.3.2.1 it can be seen that CRPS is sensitive to solar intensity. As such, another analysis has been done to categorize the CRPS by GHI. The GHI measurements used for this analysis are taken from the CAMS solar radiation time-series discussed in Section 3.1.4. The CRPS is averaged over all time windows to give a general performance indication of a model. The GHI is also averaged over time, which aligns with the largest prediction window lengths.

The results of this analysis are shown in Figure 4.19. Here it can be seen that the CRPS increases with GHI for all models. But for the neural network models with weather forecasts as an input, the CRPS goes down again as GHI goes to its maximum. The GHI is averaged over the largest prediction window length, meaning that a high GHI correlates to an almost clear sky. For clear skies, the forecast of the weather models can predict the expected power production more accurately, as there will be no interference from outside sources on the solar radiation received at the PV panel.

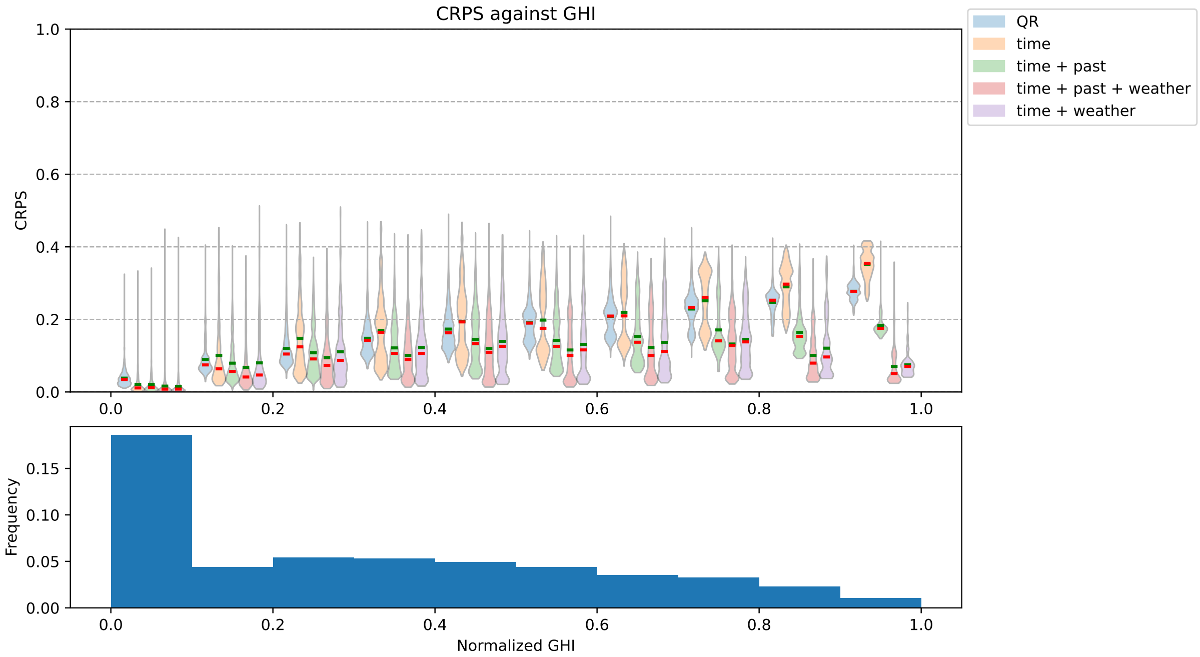


Figure 4.19: CRPS categorized by GHI. The histogram at the bottom of the figure corresponds to the frequency of samples whose GHI fall in that data range, the figure on top shows how the CRPS scores are distributed for samples belonging to that GHI range.

4.3.2.3 Cloud Coverage

From the background study summarised in Section 2.2.3 the influence of clouds on solar radiation was stated as a significant contributor to loss in PV power. Therefore, an analysis on how cloud coverage influences the CRPS has been performed. For this analysis the measurements from a nearby weather station have been used, this weather station has been detailed in Section 3.1.4. From this dataset, the cloud coverage variable has been used. These measurements are however for every hour, so the results have been linearly interpolated to match the 10-second prediction interval. The CRPS scores have been averaged over all predicted time windows, and the cloudiness index has been averaged to match the largest prediction window length.

The result of this analysis are shown in Figure 4.20. The cloudiness score shown here is an average over a span of time, meaning scores at the extremes reflect the consistently cloudy or clear sky samples and the samples in the middle reflect both medium cloudiness and transitions from cloudy to clear sky, or vice versa. The QR and Time model overall perform equally over all cloudiness levels, whereas the other models perform better at consistently cloudy or clear sky environments.

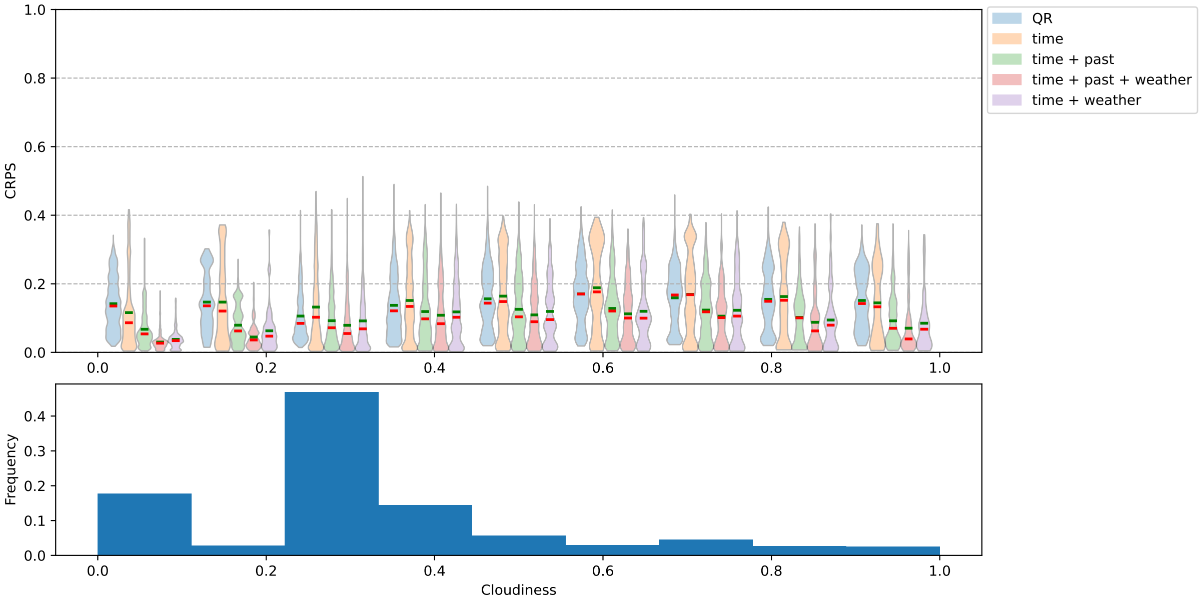


Figure 4.20: Average CRPS score separated by the cloudiness index observed by a nearby weather station, the weather station and its weather types were discussed in Section 3.1.4. The cloudiness is measured from 0, meaning no clouds, to 1, meaning full cloud coverage. The histogram at the bottom of the figure corresponds to the frequency of samples whose cloudiness fall in that data range, the figure on top shows how the CRPS scores are distributed for samples belonging to that cloudiness range.

4.3.2.4 Weather Type

The last analysis is aimed at categorising the CRPS by different weather phenomena. The different weather types and labels are taken from the KNMI weather station dataset described in Section 3.1.4. The weather types used for this evaluation are Mist, Rain, Thunder, and Ice. These weather type occurrences are indicated in the dataset by yes/no values and correspond to 1 and 0 respectively. The Snow weather type could not be used as too few samples were in the dataset that were measured during the day, so it is missing from this evaluation. Similar to the previous analysis, the CRPS is averaged over all time windows and the weather types have been interpolated to match the 10-second prediction interval of the power forecast. The dataset was then filtered using the weather type indicator, if a sample's interpolated weather type indicator was above 0.5, it is included in the distribution. The weather type distributions are shown in Figure 4.21. For the thunder setting, the models with past measurements perform significantly better. Ice and Mist weather types are better handled by the neural network models. Overall, Rain is significantly more difficult for these models to handle.

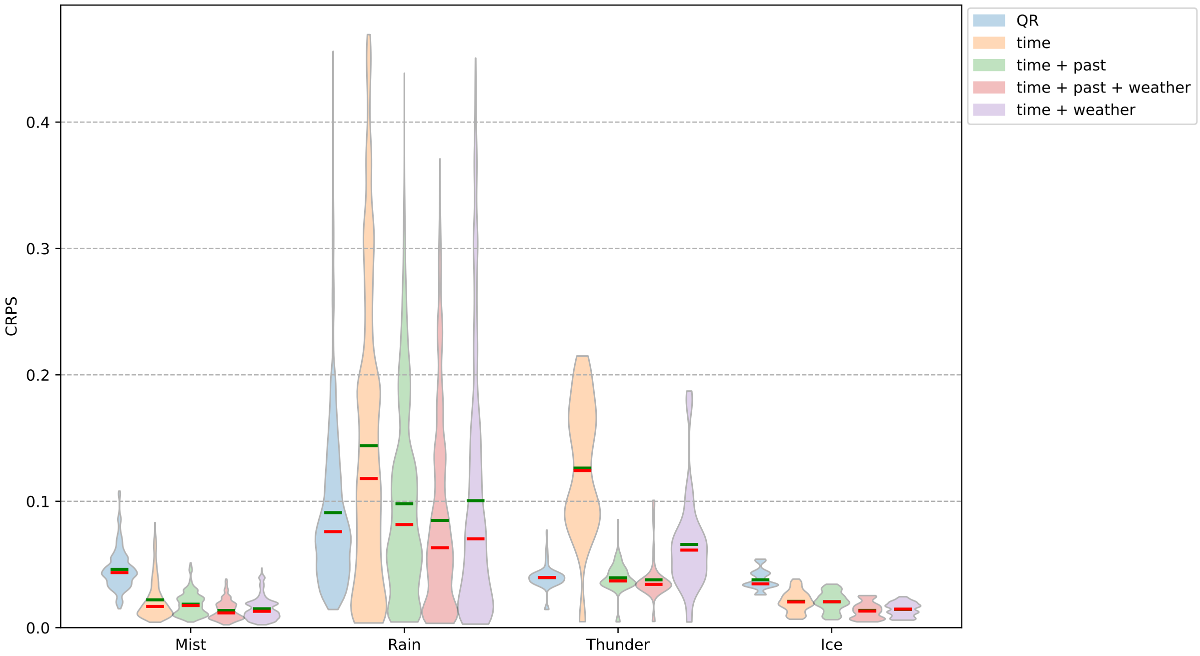


Figure 4.21: Average CRPS score separated by the different weather types observed by a nearby weather station, the weather station and its weather types were discussed in Section 3.1.4.

4.3.3 Embedded Implementation

As a last experiment, the 5 models have been implemented on an embedded device to determine their resource usage characteristics and operational feasibility of a real life application. For this experiment a NodeMCU-ESP32 DEVKITV1 is used as the embedded device, this development board houses a ESP32-D0WDQ6-V3 microcontroller running at 240MHz with 512KB of RAM and 2MB of flash memory.

In order to be able to run the 4 neural networks on this device the models have been converted to TensorFlow Lite model descriptions, these descriptions hold all the practical information needed to run the neural networks on an embedded device, such as the model's weights, a list of what operations need to be executed and in what order, memory buffer locations, and buffer sizes. An interpreter then uses these descriptions to execute the models on the embedded device. For the QR model, a program has been written that calculates the dot product between the weight matrix and the input feature vector.

In Table 4.7 it can be seen that all models are able to produce their output well before the next input sample is available 10 seconds later, all below 1% of the allotted time. For the neural networks, the inclusion of weather data increases the execution time of the model significantly compared to including past power measurements in the model. This can be attributed to the use of multiple dense layers as well as the use of an LSTM layer, which is computationally complex.

Overall, the neural networks take decidedly longer to process compared to the QR model, even the more simple Time model takes significantly longer. This has two causes, the first is the complexity of the quantile forecast part of the model, which is the same for all neural network models. The second cause is the additional work done by the interpreter loading, storing, and moving data around in memory as all operations are evaluated at runtime per instruction and not optimized beforehand, resulting in work being done that could have been avoided. The QR model with its dot product is less demanding on memory and computational

Model	Model size (bytes)	execution time (ms)	iterations/second (1/s)	Time active (%)
QR	65292	8.843	113.085	0.088
Time	84156	13.044	76.654	0.130
Time + Past	88524	13.342	74.935	0.133
Time + Weather	110540	15.769	63.400	0.158
Time + Past + Weather	114964	16.260	61.333	0.163

Table 4.7: Model characteristics when implemented on an NodeMCU-ESP32 DEVKITV1, the shown execution time and iterations/second are averaged over a 1000 model executions. The model size represents the amount of memory needed to fully describe the model, not program size. Execution time measures the time needed for the model to execute from input to output. Time active is the ratio between the time spent calculating and the time available between samples.

complexity, requiring only multiply accumulate operations iterating over the variables, which is easier for the compiler to optimize for.

4.4 Example EMS Use Cases

The use of probabilistic forecast in EMS requires a different approach compared to the currently used point forecasts, as the information in the probabilistic forecast is spread out over all quantiles instead of one single value. The probabilistic forecasts therefore have to be processed further before they are used, this allows for the information relevant to the use case to be represented more efficiently. However, using probabilistic forecasts instead of point forecasts in EMS would also enable unique opportunities that are not possible with point forecasts. These new and unique use cases are discussed in this section, together with options on how to use probabilistic forecasts where point forecasts are currently used.

Studies like [42, 58] use only a small subset of the produced quantiles, these are often the median quantile and two quantiles further apart from the median, for example the $q_{0.1}$ and $q_{0.9}$ quantiles. The median quantile is used in this case to represent the expected value of the future power, just like a standard point forecast would, and the two outer quantiles represent the lower and upper bounds the future power is expected to be in. This allows an EMS to take these worst case scenarios into account when making decisions.

Other studies like [20, 10] use the full set of quantiles available to perform cost-risk analysis to evaluate potential economic costs associated with these presumed future realities, or steer the systems using these forecasts to minimize potential costs. This can be for example to investigate how likely it is that a certain energy production quota is realized, given a certain power forecast. This example use case is demonstrated in Figure 4.22, where the Time + Past + Weather model forecasts how likely it is that the average power generated will lie within its allowed power production range, indicated by two red lines. The probabilities of each category are calculated by sampling or interpolating the quantiles near the point of interest. The allowed production ranges are kept constant in this example for simplicity, but dynamic ranges could also be applied. The boundaries of the allowed power production could for example represent a preferred power profile for charging electric vehicles. The analysis would then indicate the feasibility of that profile and possibly signal the need to update its charging schedule based on the likelihood that the current charging profile would not be sufficient. Additionally, the probabilistic boundary could also be employed to indicate the available headroom on the local power grid. The probability would then signify the chance that the local energy grid would be overloaded, allowing preventive actions to be taken.

In [20] this cost-risk analysis is given a score by comparing the derived chance to exceed a set value with a variable indicating a probability threshold from what point action should be taken. Taking action is then associated with a cost, missing an event is then associated with a loss, and correct rejection is not associated with a cost or loss. The total cost-loss ratio is then evaluated such that the balance between cost and loss can be optimized.

The last three examples all use previously established reference schedules or conditions to compare the forecast with, but this can be turned around. For example, the forecast can be used to generate the electric vehicle charging schedule with, instead of validating a charging schedule. The quantiles are then used to guide or inform the search for an optimal charging profile, balancing costs and service reliability. This setup can also be used to inform an EMS what the optimal state-of-charge should be for its batteries. This allows an EMS to better handle the unpredictability of the future and reduce the number of cases where PV power is used inefficiently due to unexpected overproduction or underproduction of PV power and not enough headroom on the state-of-charge of the batteries to make use of this.

Replacing the *'overlap'* or *'separate'* windowing methods with the *'single'* method would open up new possibilities as well. The models would in that case forecast a conditional CDF of the instantaneous power production at a certain time in the future, instead of the average power during a certain time period, this would give an indication of the volatility in the power generation. The forecast would then describe the expected range and likelihood of the instantaneous power that could be observed at that point in the future, similar to the observed power frequency shown in Figure 4.4 or power deviation frequency shown in Figure 4.5. In [50] this is used to forecast rainfall intensity and duration to predict extreme weather events. This can be applied to solar power forecasting to warn for sharp changes in the expected power generation. Another use case would be to estimate the volatility of the instantaneous power and its impact on the local power grid to model the degradation of grid infrastructure, as advocated by [3].

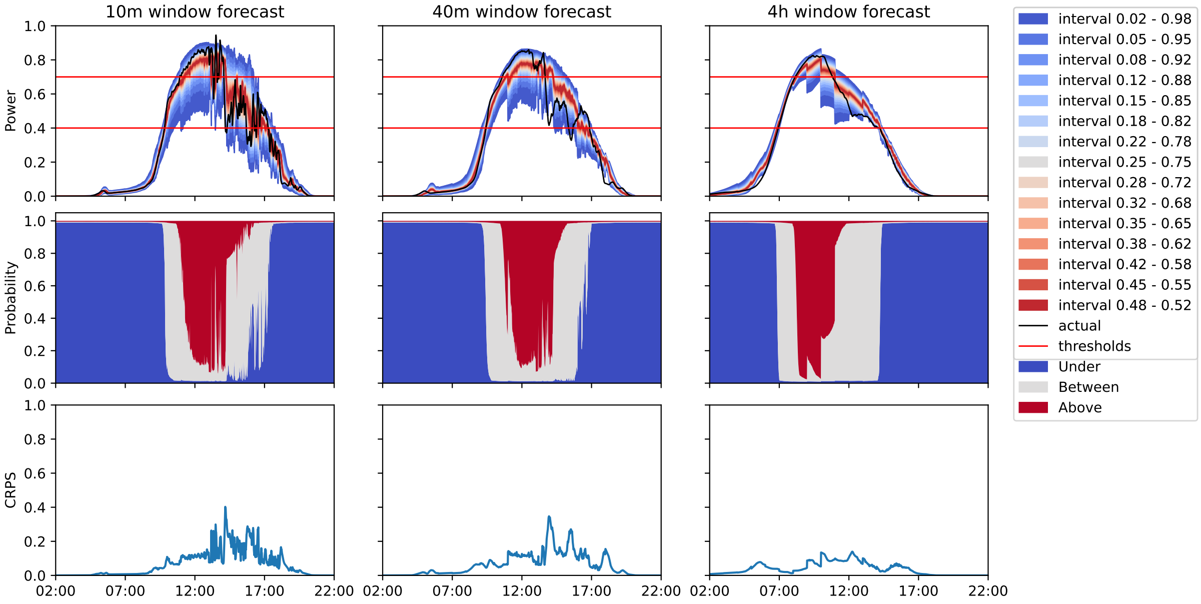


Figure 4.22: Power and probability forecast of June 3rd, 2022 made by the Time + Past + Weather model. In the top figures, the power forecast is shown, similar to Figure 4.13. In the middle figures the predicted probability is plotted of the actual power being below, between, or above the two threshold indicators as predicted by the model. In the bottom figures, the instantaneous CRPS of the power forecast and actual power is plotted as an indication of forecast accuracy.

4.5 Discussion

The use-case of this thesis drove the decisions for the future window method and the model parameters, but its selection depends on the use-case and should be researched independently per application. The results shown in this chapter are to introduce probabilistic forecasting with a specific focus on EV charging. As was found in the literature study the choice of quantiles is not fixed and the number of quantiles can be adjusted to the use-case as well, if for example only estimates of quantiles are good enough for practical applications without needing to process the quantiles for comparing performance one might reduce the number of quantiles significantly. There is likely a relation between the number of quantiles and the value that the added quantile brings, but this needs to be investigated further.

In Figure 4.21 it can be seen that when thunderclouds are expected, the models with past power measurement data included perform significantly better than their counterparts. This is likely because the thunderclouds are more regular in their shape and can be adjusted for more easily, as they are less likely to change much over time. The neural network with only weather data cannot accurately predict when exactly the thundercloud will arrive and thus produces larger errors.

The CRPS used in this thesis is not always an accurate representation of the actual performance of a model. The definition of the CRPS equates two continuous probability density functions and scores the similarity between these two densities, this has been discussed in Section 2.4.3. The metric used here however approximates the CRPS using a discrete density function in the form of quantiles together with singular observations. These approximations however need to be averaged before it converges to the true CRPS and can be compared to the MAE. In most comparisons, the CRPS scores have been averaged before use or are shown together with an average CRPS in the case of CRPS density plots like Figure 4.6. The CRPS densities plotted in these types of figures do not represent an interpretable value like the MAE does, and are only shown to illustrate how the CRPS is distributed in these situations.

The CRPS is related to the MAE as discussed in Section 2.4.3, this means that errors at high solar intensity will contribute more to the CRPS than errors in the lower solar intensity spectrum. This can give a wrong insight when interpreting CRPS scores, as the CRPS represents the absolute error, but the relative error is more similar to how an outside influence affects the generated solar power. An example of this can be seen in Figure 4.18 where the CRPS is significantly larger during summer months. This however does not necessarily mean that the models perform better in winter months. The scaled CRPS proposed in [59] would reduce the influence of scale when assessing the averaged performance of a model. But this method requires an approximation of the conditional probability density function to which the observed value belongs to in its calculation, this is not trivial to approximate and not feasible for this study.

CRPS scores both sharpness and accuracy and cannot be presented independently using only the CRPS. A wide probability might be more accurate, but a smaller probability distribution with less accuracy might perform better. As an example, QR has more reliable quantiles as shown in Figure 4.14 compared to the neural network weather models, but performs worse in all performance analysis using CRPS as it forecasts wider probability distributions. Different metrics are required such that accuracy and sharpness can be analysed independently.

Weather models forecast the average power of that hour and do not interpolate nicely from hour to hour, as can be seen in Figure 4.13. The power forecasts show the average expectation for that hour, but is actually higher or lower near the these transitions between weather forecasts. This also influences the reliability diagram shown in Figure 4.14 where this behaviour results in over or underestimations in the quantile distributions.

The low resolution weather forecast might however not be the only cause for the poor results observed in the reliability diagrams. This unreliability might be caused by the network architecture used, as QR does not have the same issues. This could be caused by the complexity of the neural network models using weather data, thus increasing the chance of the model to overfit. This overfitting can then be explained as an overconfidence in the accuracy of the weather forecast, likely due to a deficit in training data where the true uncertainty cannot be estimated and only a handful of examples are available. This overconfidence has been observed on certain days in the validation set, where the quantile forecast indicate low uncertainty, but all quantiles differ from the observed value significantly. In these situations, the forecasts also change significantly when a more recent weather forecasts is available, indicating an overreliance on the weather forecast. This goes together with the type of weather forecast used, a deterministic forecast, which cannot indicate the range of values the future can hold. Thus, when the weather forecast is off, the models' forecasts will also be off.

CHAPTER 5

CONCLUSION

In this thesis, the reliability of energy prediction for solar panels has been investigated and how it can be improved upon. The most compelling strategy found in the literature study is to transition from point-forecast methods to probabilistic forecasting methods. Two of these probabilistic forecasting methods have been used in this thesis to investigate their reliability, accuracy, and feasibility in practical applications. Next to that, examples have been provided on how these probabilistic models can be used by existing systems to improve their operation. Based on the research and studies performed in this study, the research questions stated in Section 1.3 can now be answered fully. These answers and future research possibilities are discussed in this chapter.

5.1 Research Questions

In order to answer the main research question of this thesis, how the reliability of energy prediction for solar panels can be improved, the four related research questions described in Section 1.3 are answered first. This gives the context needed to give an answer to this overarching question.

Question 1: What data can have a meaningful contribution to predicting the expected solar output power of solar panels and its uncertainty?
Four data sources were discussed as an input for future power predictions, time & seasonality, past power measurements, weather forecasts, and satellite imagery. Each source added value to their predictions, but as discussed in Section 4.2.5 the Satellite model was more resource intensive compared to the Past model and did not perform better. Next to that, the Past model does not require external information to function and can rely on a more reliable stream of data.

All models using time & seasonality information alongside weather forecasts in their inputs were found to rely more on variables representing atmospheric water content, compared to irradiance variables, as can be seen in Table 4.3. In literature, irradiance variables are often chosen as inputs to forecasting models, as discussed in Section 2.2.3. These irradiance variables were less preferred to other weather variables that complemented the information provided by the time and seasons variables, removing redundant or unimportant variables and decreasing the total number of variables used by the model. This feature selection process using the algorithm presented in [47] made this selection process more adaptive and specific to the model that is used, as can be seen in Section 4.2.3.

The information from the Time, Past, and Weather subcomponents complement each other as can be seen in Figure 4.17, where these three subcomponents together produce the best performing model. This combined model keeps the short interval performance of the Past model and the long interval performance of the Weather model, taking the best of both worlds and improving upon the CRPS in the medium interval windows. The combination of these three sources is therefore a clear recommendation.

Question 2: How can methods be used to make these predictions more reliable?
In this thesis, two main methods have been used to make probabilistic forecasts with, namely neural networks and quantile regression methods. The neural network models have been augmented with special algorithms like LassoNet [47] to reduce the feature space and make the models more generic, and K-Sparse layers [44] which try to group weather data into similar clusters to reduce the feature space even more. Reducing the feature space and restricting the possible states the models' layers can take reduces the chance of overfitting and improves stability. For the weather models, the use of LassoNet did improve the reliability overall without sacrificing accuracy too much, as can be seen in Figure 4.8.

Question 3: How can the reliability of the predictions be assessed?
As can be seen from the results discussed in Section 4.3 the reliability of the models implemented for this work can be viewed in multiple ways. The reliability diagrams in Figure 4.14 give an indication on how accurately the expected quantile distributions match with the observed distributions, providing a good visual basis of how much the models predictions should be trusted and when not. The models' performance is further dissected in Section 4.3.2 based on different weather types, time and weather observations to show possible shortcomings in their understanding of these situations. These tests form a good basis to judge the reliability of these models on.

Question 4: Are the found models practical and resource efficient solutions for real-world applications implemented on an embedded device?
As can be seen in Table 4.7 all final models can be run on an embedded device within the time or memory constraints for real-world applications. The time needed to execute the presented neural networks on an embedded device is very small relative to the time that is available to produce its results. Although the QR model is faster and uses less memory, it is trumped by the neural networks based on overall accuracy, as can be seen in Figure 4.17. The absolute difference between execution time and memory usage is less convincing than the difference in accuracy. Therefore, the choice for neural networks on embedded devices is overall a better one, given the results shown here.

How can the reliability of energy prediction of solar panels be improved?
The answers to the four research questions explain how the different aspects of reliability can be defined, measured, influenced, and how it can be improved upon in the field of energy prediction of solar panels.

The foremost improvement to energy prediction of solar panels is the change from point forecasting to probabilistic forecasts. This change grants the end-user information about the aleatoric level of uncertainty in the future energy production, which point forecasts do not. This gives a more realistic representation of the future, improving the overall reliability of the prediction. The input variables used in literature were selected and optimized for point forecasts, but these discarded input variables can still prove useful as they are possibly related to the unpredictability of the future. This can make previously unused variables useful again and improve the forecasts with additional information and reduce the epistemic uncertainty.

By studying the behaviour and limitations of models by assessing the models' performances in different environmental situations, as is done in Section 4.3, the shortcomings and expected range of error can be determined. These assessments give insight into the epistemic uncertainty of the model and gives an indication on how much a model should be trusted, given these specific circumstances. The end-user can then act on this knowledge when these circumstances arise and improve the reliability of the end-user's application.

Multiple different types of data sources used by these models results in more reliable forecasts overall, as is demonstrated in Figure 4.17. Choosing past power measurements over satellite imagery improves performance as well as reliability, as past power measurements do not rely on external sources, improving the availability of the application.

All methods and proposed improvements discussed here are feasible and actionable solutions for real-life applications, as these solutions have been demonstrated to work on a readily available embedded device with modest resource constraints.

5.2 Future Work

During the making of this thesis, many paths and opportunities were found that could prove useful for further studies. There was however not enough time to dive into these topics for this thesis, but some insights gained during the making of this thesis related to these topics can prove useful for others. As such, these improvements or other fruitful methodologies are listed below for inspiration.

Second Order Uncertainty
For future work, the pursuit of second order uncertainty is encouraged, as this can help explain the uncertainty of the model where the probabilistic forecasts are also bound to be erroneous. The second order uncertainty would then be the uncertainty of the model in its predicted output given the set of inputs to the model. This can be done by developing a second model which predicts the error that the base model will have given the same inputs. This way, the uncertainty of the model can be taken into account with the decision-making process of an EMS.

Weather Forecast Interpolation
The Weather models used in this thesis rely on data that is updated only once per hour, with data that spans the average of one hour. As a result, the predictions made by these models show step-like behaviour in their forecasts, as can be seen in Figure 4.13. This step-like behaviour is negatively impacting the forecasts, as the forecast nearer to the edges of these jumps are often over or underestimating the actual future, as can be seen in Figure 4.14. To fix this, the weather forecast can be interpolated before it is used as an input to the model, based on the time between the updates. This would mean that the model needs to calculate the weather subcomponent more often instead of each hour as it is currently, increasing the average number of operations that need to be done per inference. Another option is to interpolate the output of the subcomponent based on the output of the subcomponent for its current forecast and the weather forecast shifted over one hour into the future. This would mean the subcomponent has to be run twice per hour instead of each inference like the previous solution requires, but it might be that the output might not be well suited for interpolating. Based on the performance differences shown in Table 4.7 the additional computations required for the first solution might be negligible compared to the effort required to develop the second solution and should be investigated first.

Weather Forecast Source
Another improvement to the weather model would be finding a more frequent and faster source for weather forecasts. The main reason is that with the current source of data, there can be a difference between 10 to at most 28 hours between the moment the last observation was made and the time the prediction represents. This can lead to differences in the forecast and reality due to disparities in the weather and the weather model. These errors accumulate as the forecast and reality differ more and more over time. These errors are more pronounced in

the model's CRPS because of the long delay between the last measurements and the current time. Therefore, more frequent and recent weather forecasts are encouraged to be used in the future, as these can adjust to changes in the weather more regularly, reducing the drift between reality and the forecast.

Ensemble weather forecasts represent possible realisations of the future weather. Incorporating these possible futures as an input can give an estimate of uncertainty in the weather forecast, thereby improving the reliability of the energy forecast.

Weather Clustering

The experiment done in Section 3.2.3 using K-Sparse clustering to find common clusters in the weather data showed poor performance in most of the trained models. The models trained with that algorithm did not converge to useful models most of the time, as can be seen in Figure 4.10. This does not mean that clustering the weather forecast by type is not a valid pursuit, as the models that did converge correctly perform similar to the unconstrained models. The K-Sparse model would however require more time and effort before its predictions can be used further, with the main focus on ensuring evenly distributed activations before and during the K-Sparse clustering specific training.

Satellite Model

The Satellite model does have a future for power forecasting with its short update interval and its regional information, but would be best suited as a separate forecasting service, similar to how the weather forecast is used in this thesis. In [19] this type of model is proposed, which forecasts solar radiation variables derived from cloud movement observations made by satellites. In that research, the cloud movement model outperformed weather forecasts in the first 2 to 3 hours and could prove to be a worthwhile addition to the models introduced here.

Quantile Regression Forests

QRF was planned to be used as a benchmark model, but had to be dropped due to time constraints and uncertainty about the amount of effort required to integrate QRF. But using QRF for solar power forecasting is still compelling, as [20] recommends using QRF or tree-based methods for forecasting solar radiation, as they claim that tree-based methods are more reliable than QR or QNN. QRF would also inherently find commonality in the data and group similar results together, resulting in sharper and more accurate quantile forecasts. As found by [20], tree-based models perform better in clear sky situations as it can distinguish these unique situations separately instead of approximating the relations as a continuous function as QR and QNN do.

Bibliography

[1] IEA (2023), "World energy outlook 2023." IEA, Paris, [Online] Available: `https://www.iea.org/reports/world-energy-outlook-2023`, License: CC BY 4.0 (report); CC BY NC SA 4.0 (Annex A); (visited on Nov. 28, 2023).

[2] ECMWF, "Quantifying forecast uncertainty," Mar 2023. [Online] Available: `https://www.ecmwf.int/en/research/modelling-and-prediction/quantifying-forecast-uncertainty` (visited on Nov. 28, 2023).

[3] S. Sayeef, S. Heslop, D. Cornforth, T. Moore, S. Percy, J. Ward, A. Berry, and D. Rowe, "Solar intermittency: Australia's clean energy challenge. characterising the effect of high penetration solar intermittency on australian electricity networks.," tech. rep., CSIRO, 2012.

[4] I. Alotaibi, M. A. Abido, M. Khalid, and A. V. Savkin, "A comprehensive review of recent advances in smart grids: A sustainable future with renewable energy resources," *Energies*, vol. 13, no. 23, 2020.

[5] T. Nasir, S. S. H. Bukhari, S. Raza, H. M. Munir, M. Abrar, H. A. u. Muqeet, K. L. Bhatti, J.-S. Ro, and R. Masroor, "Recent challenges and methodologies in smart grid demand side management: State-of-the-art literature review," *Mathematical Problems in Engineering*, vol. 2021, p. 5821301, Aug 2021.

[6] B. Nijenhuis, L. Winschermann, N. B. Arias, G. Hoogsteen, and J. L. Hurink, "Protecting the distribution grid while maximizing ev energy flexibility with transparency and active user engagement," in *CIRED Porto Workshop 2022: E-mobility and power distribution systems*, vol. 2022, pp. 209–213, 2022.

[7] M. Schoot Uiterkamp, M. Gerards, and J. Hurink, "Fill-level prediction in online valley-filling algorithms for electric vehicle charging," in *Proceedings - 2018 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe 2018*, (United States), IEEE, Dec. 2018. 8th IEEE PES Innovative Smart Grid Technologies Conference Europe 2018, ISGT Europe 2018 ; Conference date: 21-10-2018 Through 25-10-2018.

[8] G. Hoogsteen, L. Winschermann, B. Nijenhuis, N. B. Arias, and J. L. Hurink, "Robust and predictive charging of large electric vehicle fleets in grid constrained parking lots," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2023 Glasgow*, IEEE, 2023. Conference date: 31-10-2023 Through 3-11-2023.

[9] J. Vreeman, "University of twente opens slimpark living lab on campus," Apr 2022. [Online] Available: `https://www.utwente.nl/en/news/2022/4/538871/university-of-twente-opens-slimpark-living-lab-on-campus` (visited on Nov. 28, 2023).

[10] A. Bracale, G. Carpinelli, P. De Falco, R. Rizzo, and A. Russo, "New advanced method and cost-based indices applied to probabilistic forecasting of photovoltaic generation," *Journal of Renewable and Sustainable Energy*, vol. 8, p. 023505, 04 2016.

[11] S. Shamshirband, T. Rabczuk, and K. Chau, "A survey of deep learning techniques: Application in wind and solar energy resources," *IEEE Access*, vol. 7, pp. 164650–164666, 2019.

[12] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," *International Journal of Forecasting*, vol. 32, no. 3, pp. 896–913, 2016.

[13] D. van der Meer, J. Widén, and J. Munkhammar, "Review on probabilistic forecasting of photovoltaic power production and electricity consumption," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1484–1512, 2018.

[14] M. Yesilbudak, M. Çolak, and R. Bayindir, "A review of data mining and solar power prediction," in *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 1117–1121, 2016.

[15] E. Lorenz, J. Hurka, D. Heinemann, and H. G. Beyer, "Irradiance forecasting for the power prediction of grid-connected photovoltaic systems," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 2, no. 1, pp. 2–10, 2009.

[16] M. O. Badawy and Y. Sozer, "Power flow management of a grid tied pv-battery system for electric vehicles charging," *IEEE Transactions on Industry Applications*, vol. 53, no. 2, pp. 1347–1357, 2017.

[17] Z. Yang, Y. Ying, and Q. Min, "Online optimization for residential pv-ess energy system scheduling," *Mathematical Foundations of Computing*, vol. 2, no. 1, pp. 55–71, 2019.

[18] K. Y. Bae, H. S. Jang, and D. K. Sung, "Hourly solar irradiance prediction based on support vector machine and its error analysis," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 935–945, 2017.

[19] P. Wang, R. van Westrhenen, J. F. Meirink, S. van der Veen, and W. Knap, "Surface solar radiation forecasts by advecting cloud physical properties derived from meteosat second generation observations," *Solar Energy*, vol. 177, pp. 47–58, 2019.

[20] K. Bakker, K. Whan, W. Knap, and M. Schmeits, "Comparison of statistical post-processing methods for probabilistic nwp forecasts of solar radiation," *Solar Energy*, vol. 191, pp. 138–150, 2019.

[21] C. Li, Y. Zhang, G. Zhao, and Y. Ren, "Hourly solar irradiance prediction using deep bilstm network," *Earth Science Informatics*, pp. 1865–0481, 2020.

[22] L. Liu, Y. Zhao, D. Chang, J. Xie, Z. Ma, Q. Sun, H. Yin, and R. Wennersten, "Prediction of short-term pv power output and uncertainty analysis," *Applied Energy*, vol. 228, pp. 700–711, 10 2018.

[23] M. Monfared, M. Fazeli, R. Lewis, and J. Searle, "Fuzzy predictor with additive learning for very short-term pv power generation," *IEEE Access*, vol. 7, pp. 91183–91192, 2019.

[24] B. G. Potter, K. Simmons-Potter, and W. F. Holmgren, "Broad-time-horizon solar power prediction and pv performance degradation research at the university of arizona," in *2018 IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC) (A Joint Conference of 45th IEEE PVSC, 28th PVSEC 34th EU PVSEC)*, pp. 2346–2350, 2018.

[25] G. G. Kim, J. H. Choi, S. Y. Park, B. G. Bhang, W. J. Nam, H. L. Cha, N. Park, and H. Ahn, "Prediction model for pv performance with correlation analysis of environmental variables," *IEEE Journal of Photovoltaics*, vol. 9, no. 3, pp. 832–841, 2019.

[26] I. A. Ibrahim, M. J. Hossain, and B. C. Duck, "An optimized offline random forests-based model for ultra-short-term prediction of pv characteristics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 202–214, 2020.

[27] W. Buwei, C. Jianfeng, W. Bo, and F. Shuanglei, “A solar power prediction using support vector machines based on multi-source data fusion,” in *2018 International Conference on Power System Technology (POWERCON)*, pp. 4573–4577, 2018.

[28] F. Golestaneh, P. Pinson, and H. B. Gooi, “Very short-term nonparametric probabilistic forecasting of renewable energy generation— with application to solar energy,” *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3850–3863, 2016.

[29] R. Al-Hajj, A. Assi, and M. M. Fouad, “Stacking-based ensemble of support vector regressors for one-day ahead solar irradiance prediction,” in *2019 8th International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 428–433, 2019.

[30] M. Ceci, R. Corizzo, F. Fumarola, D. Malerba, and A. Rashkovska, “Predictive modeling of pv energy production: How to set up the learning task for a better prediction?,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 956–966, 2017.

[31] KNMI, “About knmi.” [Online] Available: `https://www.knmi.nl/over-het-knmi/about` (visited on Nov. 29, 2023).

[32] A. Ryu, M. Ito, H. Ishii, and Y. Hayashi, “Preliminary analysis of short-term solar irradiance forecasting by using total-sky imager and convolutional neural network,” in *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*, pp. 627–631, 2019.

[33] M. Diagne, M. David, P. Lauret, J. Boland, and N. Schmutz, “Review of solar irradiance forecasting methods and a proposition for small-scale insular grids,” *Renewable and Sustainable Energy Reviews*, vol. 27, pp. 65–76, 2013.

[34] ECMWF, “Ecmwf frontpage.” [Online] Available: `https://www.ecmwf.int/` (visited on Nov. 28, 2023).

[35] ECMWF, “Atmospheric model high resolution 10-day forecast (set i - hres).” [Online] Available: `https://www.ecmwf.int/en/forecasts/datasets/set-i` (visited on May. 11, 2023).

[36] ECMWF, “Atmospheric model ensemble 15-day forecast (set iii - ens).” [Online] Available: `https://www.ecmwf.int/en/forecasts/datasets/set-iii` (visited on May. 11, 2023).

[37] KNMI, “Harmonie-arome cy40 forecasts europe.” [Online] Available: `https://dataplatform.knmi.nl/dataset/harmonie-arome-cy40-p3-0-2` (visited on Nov. 29, 2023).

[38] L. Bengtsson, U. Andrae, T. Aspelien, Y. Batrak, J. Calvo, W. de Rooy, E. Gleeson, B. Hansen-Sass, M. Homleid, M. Hortal, K.-I. Ivarsson, G. Lenderink, S. Niemelä, K. P. Nielsen, J. Onvlee, L. Rontu, P. Samuelsson, D. S. Muñoz, A. Subias, S. Tijm, V. Toll, X. Yang, and M. Ødegaard Køltzow, “The harmonie–arome model configuration in the aladin–hirlam nwp system,” *Monthly Weather Review*, vol. 145, no. 5, pp. 1919 – 1935, 2017.

[39] Copernicus, “Cams global atmospheric composition forecasts.” [Online] Available: `https://ads.atmosphere.copernicus.eu/cdsapp#!/dataset/cams-global-atmospheric-composition-forecasts.` (visited on Jun. 7, 2023).

[40] J. Widén, N. Carpman, V. Castellucci, D. Lingfors, J. Olauson, F. Remouit, M. Bergkvist, M. Grabbe, and R. Waters, “Variability assessment and forecasting of renewables: A review for solar, wind, wave and tidal resources,” *Renewable and Sustainable Energy Reviews*, vol. 44, pp. 356–375, 2015.

[41] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 12 2014.

[42] J. F. Jønler, F. Brunø Lottrup, B. Berg, D. Zhang, and K. Chen, “Probabilistic forecasts of global horizontal irradiance for solar systems,” *IEEE Sensors Letters*, vol. 7, no. 1, pp. 1–4, 2023.

[43] A. Makhzani and B. J. Frey, “A winner-take-all method for training sparse convolutional autoencoders,” *CoRR*, vol. abs/1409.2752, 2014.

[44] A. Makhzani and B. Frey, “k-sparse autoencoders,” 2014.

[45] N. Mrabah, N. M. Khan, and R. Ksantini, “Deep clustering with a dynamic autoencoder,” *CoRR*, vol. abs/1901.07752, 2019.

[46] Y. Ren, P. Suganthan, and N. Srikanth, “Ensemble methods for wind and solar power forecasting—a state-of-the-art review,” *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 82–91, 2015.

[47] I. Lemhadri, F. Ruan, L. Abraham, and R. Tibshirani, “Lassonet: A neural network with feature sparsity,” 2021.

[48] L. Massidda and M. Marrocu, “Quantile regression post-processing of weather forecast for short-term solar power probabilistic forecasting,” *Energies*, vol. 11, no. 7, 2018.

[49] M. Zamo and P. Naveau, “Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts,” *Mathematical Geosciences*, 11 2017.

[50] A. J. Cannon, “Non-crossing nonlinear regression quantiles by monotone composite quantile regression neural network, with application to rainfall extremes,” *Stochastic Environmental Research and Risk Assessment*, vol. 32, pp. 3207–3225, Nov 2018.

[51] N. Meinshausen, “Quantile regression forests,” *J. Mach. Learn. Res.*, vol. 7, p. 983–999, dec 2006.

[52] P. Pinson, P. McSharry, and H. Madsen, “Reliability diagrams for non-parametric density forecasts of continuous variables: Accounting for serial correlation,” *Quarterly Journal of the Royal Meteorological Society*, vol. 136, no. 646, pp. 77–90, 2010.

[53] T. Gneiting and A. Raftery, “Strictly proper scoring rules, prediction, and estimation,” *Journal of the American Statistical Association*, vol. 102, pp. 359–378, 03 2007.

[54] EUMETSAT, “Cloud mask - msg - 0 degree.” [Online] Available: `https://navigator.eumetsat.int/product/EO:EUM:DAT:MSG:CLM` (visited on Nov. 30, 2023).

[55] KNMI, “Uurgegevens van het weer in nederland.” [Online] Available: `https://www.knmi.nl/nederland-nu/klimatologie/uurgegevens` (visited on Nov. 30, 2023).

[56] Copernicus, “Cams solar radiation time-series.” [Online] Available: `https://ads.atmosphere.copernicus.eu/cdsapp#!/dataset/cams-solar-radiation-timeseries?tab=overview`. (visited on May. 11, 2023).

[57] F. Rodrigues and F. C. Pereira, “Beyond expectation: Deep joint mean and quantile regression for spatiotemporal problems,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5377–5389, 2020.

[58] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, “A multi-horizon quantile recurrent forecaster,” *arXiv: Machine Learning*, 2017.

[59] D. Bolin and J. Wallin, “Local scale invariance and robustness of proper scoring rules,” *Statistical Science*, vol. 38, no. 1, pp. 140–159, 2023.

Appendix A

Weather Only Model Run

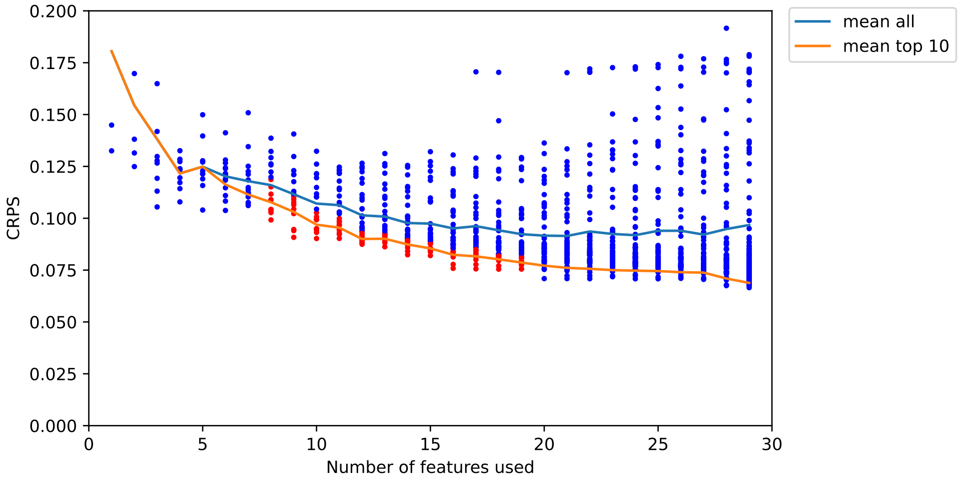


Figure A.1: The CRPS scores of all considered weather models relative to the amount of features needed to make their prediction. Each dot represents the performance of a single model, the red dots are the best performing models that fall within the selection criteria as specified in Section 4.2.3, the blue dots do not.

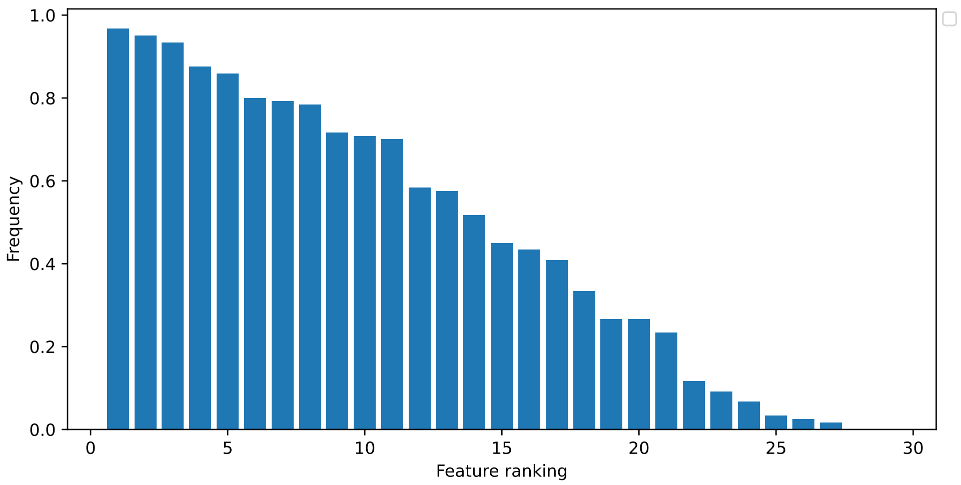


Figure A.2: Sorted frequency graph showing how often a feature remained significant in the best performing weather models. A value of 1 means the feature was present in all of these models, and a 0 means no model kept that feature.

top	Identifier	Description	Unit	Frequency
1	ssr	Surface net short-wave (solar) radiation	J/m^2	0.967
2	tclw	Total column cloud liquid water	kg/m^2	0.950
3	cp	Convective precipitation	m	0.933
4	tciw	Total column cloud ice water	kg/m^2	0.875
5	tcrw	Total column rain water	kg/m^2	0.858
6	tisr	TOA incident solar radiation	J/m^2	0.800
7	sund	Sunshine duration	s	0.792
8	lcc	Low cloud cover	$(0-1)$	0.783
9	tcslw	Total column supercooled liquid water	kg/m^2	0.717
10	tcsw	Total column snow water	kg/m^2	0.708
11	dsrp	Direct solar radiation	J/m^2	0.700
12	lsp	Large-scale precipitation	m	0.583
13	ssrc	Surface net short-wave (solar) radiation, clear sky	J/m^2	0.575
14	tcw	Total column water	kg/m^2	0.517
15	fdir	Total sky direct solar radiation at surface	J/m^2	0.450
16	tsrc	Top net solar radiation, clear sky	J/m^2	0.433
17	mcc	Medium cloud cover	$(0-1)$	0.408
18	hcc	High cloud cover	$(0-1)$	0.333
19	ssrdc	Surface solar radiation downward clear-sky	J/m^2	0.267
20	tcc	Total cloud cover	$(0-1)$	0.267
21	ssrd	Surface short-wave (solar) radiation downwards	J/m^2	0.233
22	cdir	Clear-sky direct solar radiation at surface	J/m^2	0.117
23	tsr	Top net short-wave (solar) radiation	J/m^2	0.092
24	tp	Total precipitation	m	0.067
25	t2m	2 metre temperature	K	0.033
26	fal	Forecast albedo	$(0-1)$	0.025
27	u10	10 metre U wind component	m/s^2	0.017
28	v10	10 metre V wind component	m/s^2	0.000
29	sp	Surface pressure	Pa	0.000

Table A.1: Sorted list of weather features by the amount of times the feature was present in the best models.