

Master in Intelligent Interactive Systems
Universitat Pompeu Fabra

Deep Learning-Based Photovoltaic Energy Forecasting with ground-based Sky Imagery and Atmospheric data

Inés Montoya Espinagosa

Supervisor: Antonio Agudo

July 2025



Contents

1	Introduction	1
1.1	Motivation	3
1.2	Objectives	4
1.3	Structure of the Report	4
2	Dataset	5
2.1	SKIPPD	5
2.1.1	Nowcast	6
2.1.2	Forecast	7
2.2	ERA5	8
2.2.1	Data coverage	8
2.2.2	Meteorological variables	10
2.2.3	Meteorological data pre-processing	12
2.3	Sun Position	14
3	Model	21
3.1	Differences between nowcast and forecast	21
3.2	SUNSET_nowcast model architecture	23
3.3	SUNSET_forecast model architecture	25
3.4	Introduction of variables into the models	27
3.4.1	Single meteorological variable	28
3.4.2	Multiple meteorological variables	28
3.4.3	Sun position	28

3.5	Testing and evaluation metrics (RMSE, MAE)	28
4	Results	30
4.1	Nowcast results	30
4.2	Forecast results	32
5	Conclusion	36
5.1	Future Work	39
	List of Figures	40
	List of Tables	42
	Bibliography	43
A	Creation of Forecast dataset	46

Acknowledgement

I would like to express my sincere gratitude to:

- Antonio Agudo, my supervisor.
- My family and friends.
- I would also like to thank those who provided me valuable advice, help, guidance and recommendations throughout the development of this work, including M^a del Carmen Llasat, Raül Marcos, Carlo Guzzon, Arne Berresheim, Yuhao Nie and Nils Lehmann.

Abstract

Due to the rise in the use of renewable energies as an alternative to traditional energies, and especially solar energy, there is increasing interest in studying how to address photovoltaic forecasting in the face of the challenge of variability in photovoltaic energy production, using different techniques and methodologies. This work develops a hybrid methodology for short and long-term forecasting based on two studies with the same purpose. A multimodal approach is proposed that combines images of the sky and photovoltaic energy history with meteorological data. The main objective is to improve the accuracy of ramp event prediction, increase the robustness of forecasts in cloudy conditions, and extend capabilities beyond nowcasting, in order to support more efficient operation of the power grid and better management of solar variability. Deep convolutional neural network architectures are used for nowcasting and forecasting models, incorporating individual and multiple meteorological variables, as well as solar position. The results demonstrate that the inclusion of meteorological data, particularly the surface long-wave (thermal), radiation downwards (strd), and the combination of wind and solar position, significantly improves predictions in both nowcasting and forecasting tasks, especially on cloudy days. This study highlights the importance of integrating diverse data sources to improve the reliability and interpretability of solar energy prediction models.

Keywords: Neural Networks; Photovoltaic Energy Forecasting; Fish-eye Sky Images, Meteorological Data.

Chapter 1

Introduction

Photovoltaic (PV) solar energy has become a crucial element in the global transition to renewable energy, and the International Energy Agency (IEA) predicts that it will become the largest renewable source by 2030 [1]. However, the variability of solar energy, mainly due to the dynamic nature of clouds, causes rapid fluctuations in irradiation that result in unstable energy production [2], which has a negative impact on grid stability [3, 4]. This unpredictability makes intra-hour solar forecasting (IHSF) and short-term photovoltaic forecasting (PVPF) particularly difficult, but forecasting is needed to maintain grid stability and ensure the system operates safely [5, 6, 7, 8, 9]. To address these challenges, studies are being conducted to improve the accuracy of solar forecasts using sky image-based prediction models, understand the impact of specific atmospheric conditions, and develop prediction models.

One way to improve forecast accuracy is to use models based on sky images. These approaches leverage visual information to anticipate fluctuations in photovoltaic production more accurately than purely numerical or statistical models [7, 9]. For example, [10] developed a two-stream network with photovoltaic energy-guided attention that integrates visual and photovoltaic production information. While this architecture achieved high accuracy, it struggled with limited training data and lacked interpretability, pointing to the importance of both data quantity and explainabil-

ity in complex models. Next, [7] uses a deep-learning model for solar irradiance nowcasting, using a transformer architecture, and then studies the integration of the model into a hybrid system, which uses physics-based models and smart persistence. Their model improves on persistence models and demonstrates better mean absolute error and mean squared error metrics. However, it still has difficulty predicting ramp events (sudden changes in power) and highlights the importance of combining physics-based and data-based approaches to improve robustness. The study also supports the use of nonparametric quantile-based methods to manage prediction uncertainty, indicating a shift toward probabilistic forecasting for operational improvement of the grid.

[11] developed different deep-learning models using static and dynamic images of the sky, and then applied a hybrid model, which showed higher performance. Despite this, they found that their hybrid models under-perform reference models under stable conditions and generate blurry predictions of cloud boundaries. [12] achieved higher accuracy through cloud coverage rates, showing an approximate performance gain of 2%, but their method fails in rainy conditions due to interference from raindrops and limitations inherent in RGB-based coverage calculations. The research emphasized the need for adjustments that take weather conditions into account, especially in Artificial Neural Networks (ANN) models, which showed weak generalization in adverse conditions. Similarly, [13] introduced the aerosol scattering coefficient as an auxiliary feature to improve forecasting in fog-affected scenarios. By combining ground-based sky images with machine-learning approaches such as LSTM, SVM, and XGBoost, their method achieved notable improvements, up to 1.08% and 0.6% in thick and light fog conditions, respectively, demonstrating the importance of including atmospheric composition in prediction frameworks.

Expanding the current landscape, [14] achieved high prediction accuracy by incorporating cloud motion vectors, although the model is overfitted and heavily relies on accurate cloud tracking. Meanwhile, [15] proposed a hybrid mapping model combining CNN, LSTM, and ANN for very short-term solar photovoltaic prediction. Their model integrated sky images and irradiance data through preprocessing, k -

means clustering, and convolutional autoencoders to improve mapping quality. The hybrid framework outperformed standalone convolutional neural networks (CNN) or LSTM models in RMSE, MAE, and correlation metrics, highlighting the value of multi-modal feature extraction. However, the study acknowledged that further improvement in generalization to varied cloud dynamics is needed.

1.1 Motivation

The increasing use of solar energy into power grid has revealed a critical problem: solar energy production varies a lot due to the cloud's motion and changing weather. While artificial-intelligence (AI) methods have helped with short-term solar predictions, current systems still have two key limitations: (1) efficiently predict fluctuations in solar irradiance (called ramp events), and (2) perform accurate forecasts beyond intra-hour timeframes. These challenges have a direct impact on the grid stability and operational efficiency, since they need to perfectly match electricity supply with demand at all times.

This thesis expands on [16] work that used CNNs with Sky Images and Photovoltaic Power Dataset (SKIPPD) [17] ground-based sky images to predict solar power output. While their model showed promise, it faced several key challenges: (1) The neural network could not properly learn all the important patterns, reducing its accuracy, (2) during cloudy weather when solar power changes rapidly, the predictions became less reliable, (3) the dataset did not consistently produce matching results, suggesting data quality issues, (4) problems with image brightness and cloud identification further affected the model's performance.

To address these gaps, this work proposes a multi-modal hybrid methodology approach that combines multiple data sources and methodologies. The approach combines high-resolution ground-based sky images from the SKIPPD dataset [17], along with physical atmospheric data from the ERA5 reanalysis [18] from ECMWF [19]. This strategy aims to use the strengths of each data type: the fine spatial-temporal resolution of sky images for local cloud dynamics and the physical parameters from

ERA5 (including cloud coverage, wind speed, etc.).

1.2 Objectives

This thesis proposes a hybrid methodology to improve short-term solar forecasting by combining ground-based sky images with publicly available meteorological data. The goal is to enhance the prediction of ramp events, improve forecast robustness under cloudy conditions, and extend capabilities beyond nowcasting. By increasing accuracy and reliability, this work aims to support more efficient power grid operation, enable better management of solar variability, as well as provide physical interpretability of the model's predictions.

1.3 Structure of the Report

The work begins with a description of the study area and the database that have been used in Chapter 2. Chapter 3 describes the different models, how meteorological variables are processed and fed to the models, and the evaluation metrics. Results are shown in Chapter 4, and the work ends with the conclusions.

Chapter 2

Dataset

For accurate solar nowcasting and forecasting, it is essential to have a high-quality database. This chapter describes the two used dataset in this work: SKIPDD [17] and ERA5 [20].

2.1 SKIPPD

The SKIPPD [17] provides three years (2017–2019) of quality-controlled, down-sampled sky images and corresponding PV power generation data, processed for use in short-term solar forecasting with deep-learning methods. The dataset includes 1-minute resolution sky images extracted from high-definition 360-degree fisheye videos recorded between 6:00 AM and 8:00 PM PST using a 6-megapixel camera (Hikvision DS-2CD6362F-IV2) mounted on the Green Earth Sciences Building at Stanford University. The images are aligned with PV output data collected from nearby solar panels on the Jen-Hsun Huang Engineering Center, recorded at a matching 1-minute frequency. The PV system consists of 30.1 kW-DC fixed polycrystalline panels, oriented at a 22.5° tilt and an azimuth of 195° . PV measurements reflect the average output per minute and are time-synchronized with the images.

SKIPPD [17] provides:

- Benchmark dataset: 3 years (2017 - 2019) of 64×64 sky images and synchro-

nized 1-min PV power energy output.

- Raw dataset: High resolution sky videos (2048×2048 , 20 fps), full-size sky image frames (2048×2048), and PV power output recorded at 1-minute intervals.

In this study it has been used the benchmark dataset for training the nowcasting models, with additional raw data –historical PV output and sky images frames– used to reconstruct missing data from the benchmark dataset in order to train the forecasting models. All images are downsampled to 64×64 , following [17], which shows this resolution offers a good trade-off between accuracy and efficiency.

2.1.1 Nowcast

For nowcasting tasks, it has been directly used the processed files provided in the SKIPPD benchmark dataset [17], which includes a development set (used for training and validation) and a test set, containing 349,372 and 14,003 samples, respectively—approximately a 96%:4% split ([17]). The test set consists of 20 days (10 sunny and 10 cloudy) manually selected from the years 2017 to 2019. This HDF5 file contains two main groups:

- `/trainval`: the development set, which includes:
 - `images_log` — a dataset of shape $(349372, 64, 64, 3)$ containing the processed RGB sky images with a resolution of 64×64 pixels. The data type is `uint8`.
 - `pv_log` — a dataset of shape (349372) containing the corresponding PV power values in kilowatts (kW), and stored as `float64` values. Each value represents the average power generated over a one-minute interval.
- `/test`: the test set, which mirrors the structure of `/trainval`, but with fewer samples:
 - `images_log` — shape $(14003, 64, 64, 3)$, type `uint8`.
 - `pv_log` — shape (14003) , type `float64`.

`times_trainval.npy` and `times_test.npy`: NumPy arrays containing the timestamps associated with the samples in the development and test sets, respectively, allowing temporal alignment of the image and PV data.

The overall dataset comprises a total of 363,375 aligned pairs of sky images and PV power values.

2.1.2 Forecast

The main difference between the `nowcast_dataset.hdf5` and `forecast_dataset.hdf5` files lies in the type of target used to train and evaluate prediction models. In contrast to the `nowcast_dataset.hdf5` file, which contains only sky images (`images_log`) and corresponding PV generation values (`pv_log`), the `forecast_dataset.hdf5` file incorporates a third element: `pv_pred`, which serves as the prediction target for forecasting tasks. This component represents the actual photovoltaic generation measured 15 minutes after the current time step and is used as ground truth during model training and evaluation. In this dataset, there is a 93.4%:6.1% split. This HDF file contains two main groups:

- `/trainval`: the development set, which includes:
 - `images_log` — a dataset of shape (129577, 16, 64, 64, 3) containing sequences of 16 RGB sky images (one per minute) covering the 15-minute window before each prediction point. The data type is `uint8`.
 - `pv_log` — a dataset of shape (129577, 16) with the actual PV generation values (in kW) for the same 15-minute window. Stored as `float64` values.
 - `pv_pred` — a dataset of shape (129577) containing the PV generation value recorded 15 minutes after the last image in the sequence. This is the target value for the forecasting model, stored as `float64`.
- `/test`: the test set, which mirrors the structure of `/trainval`, but with fewer samples:

- `images_log` — shape (7912, 16, 64, 64, 3), type `uint8`.
- `pv_log` — shape (7912, 16), type `float64`.
- `pv_pred` — shape (7912), type `float64`.

`times_trainval.npy` and `times_test.npy`: NumPy arrays containing the timestamps associated with each prediction target (`pv_pred`) in the development and test sets, respectively.

The generation of `forecast_dataset.hdf5` and the complementary files is being explained in Appendix A. Test days are different from the ones picked in `nowcast_dataset.hdf` in order to avoid missing data.

2.2 ERA5

2.2.1 Data coverage

ERA5 [20] is a climate reanalysis dataset produced by ECMWF (European Center for Medium-Range Weather Forecasts) [19]. It combines model data with observations from around the world to create a consistent and comprehensive record of past weather and climate. ERA5 [20] provides hourly estimates of a wide range of atmospheric, land, and oceanic variables, covering the period from 1940 to the present. Reanalysis involves using a weather model to assimilate all available past observations (from satellites, weather stations, etc.) to create a globally consistent and physically realistic dataset of past weather conditions. ERA5 [20] offers data on a 0.25° grid (31 km approx.) and resolves the atmosphere using 137 levels, as well as including information about uncertainties in the data as well as being continuously updated, with daily updates available five days behind real time. In order to download the data, it has been used four spatial points in GRIB format within the grid where the camera is located on top of the Green Earth Sciences Building (37.427° , -122.174°) at Stanford University. Specifically:

Given these geographic coordinates (see Fig. 1), we obtain a difference in latitude

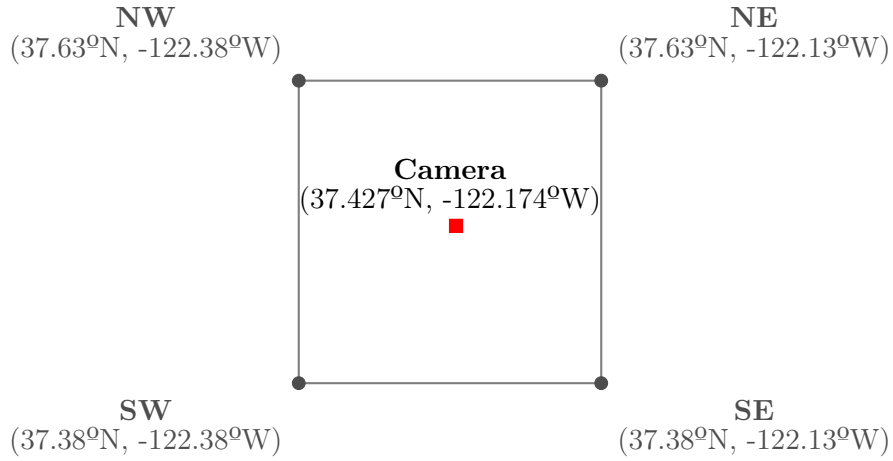


Figure 1: **Latitude and longitude of the four ERA5 grid points** surrounding the camera location in Stanford. These points, spaced at 0.25° resolution (NW, NE, SW, SE), define the area from which ERA5 variables are extracted. The red square indicates the camera position.

of:

$$\Delta\phi = 37.63^\circ - 37.38^\circ = 0.25^\circ, \quad (2.1)$$

and a difference in longitude of:

$$\Delta\lambda = -122.13^\circ - (-122.38^\circ) = 0.25^\circ. \quad (2.2)$$

One degree of latitude corresponds approximately to 111.132 km [21], a value that is nearly constant on the Earth's surface. Therefore, the physical displacement in the north–south direction is:

$$d_{\text{NS}} = \Delta\phi \cdot 111.132^\circ = 0.25^\circ \cdot 111.132^\circ = 27.78^\circ\text{km}. \quad (2.3)$$

The distance covered by one degree of longitude depends on the latitude and can be calculated as:

$$1^\circ \text{ longitude} \approx 111.320\text{km} \cdot \cos(\phi). \quad (2.4)$$

Using the average latitude between 37.63° and 37.38° :

$$\phi_{\text{avg}} = \frac{37.63^\circ + 37.38^\circ}{2} = 37.505^\circ, \quad (2.5)$$

$$\cos(37.505^\circ) \approx 0.7935, \quad (2.6)$$

$$1^\circ \text{ longitude} \approx 111.320 \cdot 0.7935 \approx 88.34\text{km}, \quad (2.7)$$

$$d_{EW} = \Delta\lambda \cdot 88.34 = 0.25^\circ \cdot 88.34 = 22.09\text{km}. \quad (2.8)$$

Therefore, the total spatial coverage of the defined rectangular area is approximately:

$$27.78\text{km (N-S)} \times 22.09\text{km (E-W)}. \quad (2.9)$$

2.2.2 Meteorological variables

In this study it has been considered the following meteorological variables extracted from GRIB files from the ECMWF Reanalysis v5 (ERA5) [20]. All variables have hourly resolution, which are then further processed in order to align with 1-minute frequency resolution:

- **Total cloud cover (tcc)**: represents the fraction of the sky covered by clouds, considering all vertical layers of cloud cover. It is expressed as a dimensionless number between 0 (completely clear sky) and 1 (completely overcast sky). This variable is provided as instantaneous analysis data (not forecasted), is not accumulative and no step dimension.
- **Wind gust (i10fg)**: represents the maximum wind speed expected at 10 meters above the surface for a given forecast interval. It is a forecast variable, measured in meters per second (m/s), and includes a step dimension. It is not accumulative, but instead represents the maximum instantaneous wind speed estimated from model variables (such as wind shear, surface friction, and stability).
- **Wind u component (100u)**: denotes the east-west (zonal) component of wind at 100 meters above ground. It is expressed in meters per second (m/s) and is provided as instantaneous analysis data. It is not forecasted and not accumulative.

- **Wind v component (100v)**: denotes the north-south (meridional) component of wind at 100 meters above ground. It is expressed in m/s and corresponds to instantaneous analysis data. It is not accumulative and does not include forecast steps.
- **Surface pressure (sp)**: indicates the atmospheric pressure at the surface level, measured in pascals (Pa). It is an instantaneous analysis variable (not forecasted), and it is not accumulative.
- **Surface long-wave (thermal) radiation downwards (strd)**: represents the net long-wave (thermal) radiation received by the Earth's surface and clouds. It is a forecast variable, accumulative over the forecast step, and expressed in J/m^2 .
- **Surface short-wave (solar) radiation downwards (ssrd)**: it is the total shortwave solar radiation reaching the Earth's surface, taking into account both direct and diffuse solar radiation. It is a forecast variable, accumulative and with a forecast step, measured in J/m^2 .
- **Surface net thermal radiation (str)**: it is the surface net thermal radiation that emits the Earth's surface. It is a forecast and accumulative variable, expressed in J/m^2 .
- **Top net solar radiation (tsr)**: it represents the net incoming shortwave solar radiation at the top of the atmosphere — defined as the difference between incoming and outgoing shortwave radiation. This is a forecast variable, accumulated over time, and expressed in joules per square meter (J/m^2).
- **Surface direct short-wave (solar) radiation (fdir)**: it quantifies the amount of direct solar radiation that reaches the Earth's surface (without being scattered). It is a forecast variable, accumulative, and expressed in J/m^2 .

Variable	Full name	Source	Step	Units	Variable type
tcc	Total Cloud Cover	an	No	Fraction (0–1)	Instantaneous
i10fg	10m Wind Gust	fc	Yes	m/s	Instantaneous
100u	U component of wind at 100m	an	No	m/s	Instantaneous
100v	V component of wind at 100m	an	No	m/s	Instantaneous
sp	Surface Pressure	an	No	Pa	Instantaneous
strd	Surface Thermal Radiation Downwards	fc	Yes	J/m ²	Accumulative
ssrd	Surface Solar Radiation Downwards	fc	Yes	J/m ²	Accumulative
str	Surface Net Thermal Radiation	fc	Yes	J/m ²	Accumulative
tsr	Top Net Solar Radiation	fc	Yes	J/m ²	Accumulative
fdir	Surface Solar Direct Radiation	fc	Yes	J/m ²	Accumulative

Table 1: **Summary of ERA5 variables** (**fc** = **forecast**; **an** = **analysis**). The table shows a summary of the official parameter name of the meteorological variables (1st column), the full name, the source (if it is a analysis or forecast variable), whether the data has or not a step, the units, and the variable type (whether it is an accumulative variable in time or not) in the last column.

2.2.3 Meteorological data pre-processing

Each variable was pre-processed in different ways to address its temporal structure and data type. The procedure applied is described below.

To integrate ERA5 [20] meteorological data in the different models, three different preprocessing techniques have been applied on the type of variable: **(i)** instantaneous analysis variables, **(ii)** forecast non-accumulative ones, and **(iii)** forecast accumulative ones.

- Instantaneous analysis variables: {**tcc**, **sp**, **100u**, **100v**}

Variables such as total cloud cover (**tcc**), surface pressure (**sp**), and wind components at 100 meters (**100u**, **100v**) are provided as instantaneous reanalysis data (**an**) with hourly resolution. Since they are not forecast-based, they do not include a **step** dimension, and each record corresponds to a specific analysis timestamp.

To synchronize these data with sky images (which are timestamped in local time), the timestamps are converted from UTC to the **America/Los_Angeles** timezone using `tz_convert()`.

Values are extracted at the four grid points (NW, NE, SW, SE) via nearest-neighbor spatial selection. The resulting time series is aligned to the image

timestamps using backward fill (`ffill`) followed by forward fill (`bfill`) to handle any missing values at the start.

Finally, the meteorological data are converted into NumPy arrays and normalized. Mean and standard deviation are computed on the training and validation split. If the standard deviation is zero (indicating constant values), it is forced to 1.0 to avoid division by zero. This normalization ensures that all variables contribute equally during model training, regardless of their original units and/or magnitudes.

- Forecast non-accumulative variable: `{i10fg}`

The variable `i10fg` (10-meter wind gust) is a forecast product (`fc`), but unlike radiation variables, it is not accumulative. Each record represents the maximum instantaneous gust expected during the forecast interval defined by `time + step`.

The real timestamps are reconstructed by summing the base forecast time (`time`) and the lead time (`step`). These are then converted to local time (`America/Los_Angeles`).

Values are extracted at the four grid points (NW, NE, SW, SE) and combined into a time series. The same re-indexing and interpolation strategy is applied as with analysis variables. Data are then converted into NumPy arrays and normalized using the same statistics as in the training split, ensuring consistency.

- Forecast accumulative variables: `{ssrd, strd, str, tsr, fdir}`

Variables such as surface solar radiation downwards (`ssrd`), surface thermal radiation downwards (`strd`), surface thermal radiation upwards (`str`), top solar radiation (`tsr`), and direct solar radiation at surface (`fdir`) are forecast-based and accumulative. Each value represents the cumulative energy (in J/m^2) received from the base time up to `time + step`.

To obtain hourly physical rates, the difference between successive steps is computed via `.diff('step')`. This transforms cumulative totals into specific values, and the first step (step = 0), which has no previous value to subtract from, is discarded.

After applying `.diff()` to the accumulative data, the first step is to slice out the initial step (which becomes NaN), resulting in misaligned dimensions. To address this, the data are flattened using a temporary `flat` dimension and assigned real timestamps (converted to `America/Los_Angeles` local time), forming a continuous hourly time series. Data from the four grid points are then extracted, merged, and interpolated to match the image timestamps. Normalization follows the same procedure as in previous cases.

All those reconstruction processes ensure that the resulting time series of the different variables are ready to be fed into the nowcast and forecast models.

2.3 Sun Position

In order to improve model prediction, solar position has been incorporated as an additional input variable. This variable is calculated using astronomical equations that determine the apparent location of the Sun in the image based on the date and time. Its inclusion helps the model identify lighting patterns and atmospheric conditions linked to the solar cycle, which can be particularly useful in scenarios with cloud cover or varying solar incidence throughout the day.

The calculation is based on two fundamental parameters that define the solar position: the solar azimuth (α), which is the horizontal angle measured clockwise from the true north to the sun's projection on the horizon plane (0° – 360°), and the solar elevation (ϵ), which is the angle between the horizon and the Sun (0° at the horizon and 90° at the zenith).

The camera used is a fisheye type with a 180° field of view, installed on the roof of Stanford's Green Earth Sciences building (latitude 37.427° , longitude -122.174°), oriented approximately 14° southwest. The `pvlib` library is used to calculate the sun's

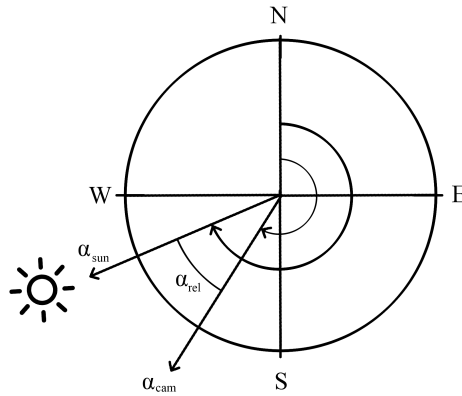


Figure 2: **Scheme on the conversion from traditional polar-to-Cartesian coordinate system to our image coordinate system.** In order to get the position of the sun relative to the camera, it is subtracted the angle of the sun α_{sun} to the orientation of the camera $\alpha_{\text{cam}} = 194^\circ$, so that we end up having α_{rel} , the angle which will be used for placing the sun position in the image sky camera.

position, accounting for latitude, longitude and timezone (`America/ Los_Angeles`), including seasonal changes due to daylight saving time.

The first step is to obtain the solar azimuth and elevation for each timestamp using `pvlib`. Since the camera is oriented 194° from the true north ($180^\circ + 14^\circ$), the solar azimuth is converted into a relative azimuth with respect to the camera's optical axis (see Fig. 2):

$$\alpha_{\text{rel}} = (\alpha_{\text{sun}} - \alpha_{\text{cam}}) \text{with } \text{mod } 360^\circ, \quad (2.10)$$

where:

- α_{rel} is the corrected relative azimuth,
- α_{sun} is the solar azimuth from `pvlib`,
- $\alpha_{\text{cam}} = 194^\circ$ is the camera's orientation.

This relative azimuth is then converted into a projection angle as

$$\theta = \text{deg2rad}(\alpha_{\text{rel}}). \quad (2.11)$$

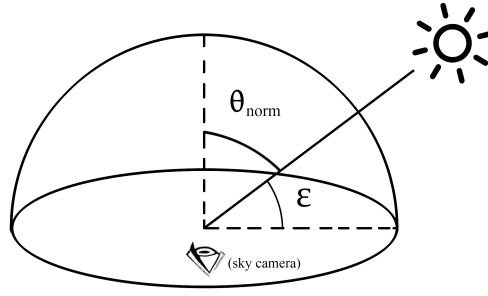


Figure 3: **Scheme on the transformation from the elevation angle of the sun to the zenith angle.** The `pvl` transforms the latitude and longitude to an elevation angle, but as the reference point is the camera, it is needed to convert the coordinate system relative to the camera point of view.

To estimate the radial distance r of the Sun from the center of the image, the zenith angle is normalized as (see Fig. 3):

$$\theta_{\text{norm}} = \frac{90^\circ - \epsilon}{90^\circ}. \quad (2.12)$$

This normalized angle ranges from 0 (Sun at the zenith) to 1 (Sun at the horizon). The radial projection is defined empirically as:

$$r = R \cdot (\theta_{\text{norm}})^\gamma, \quad (2.13)$$

where:

- R is the maximum image radius (e.g., 32 px for a 64×64 image),
- $\gamma = 1.2$ is an exponent that adjusts the non-linearity of the projection.

This type of non-linear projection is inspired by the equidistant fisheye projection model, where the projected radial distance relates linearly to the incidence angle (see Fig. 4) as:

$$r_d = f \cdot \theta, \quad (2.14)$$

with f being the focal length and θ the incident ray angle [22]. In this case, the relationship is adapted to better match the actual camera geometry.

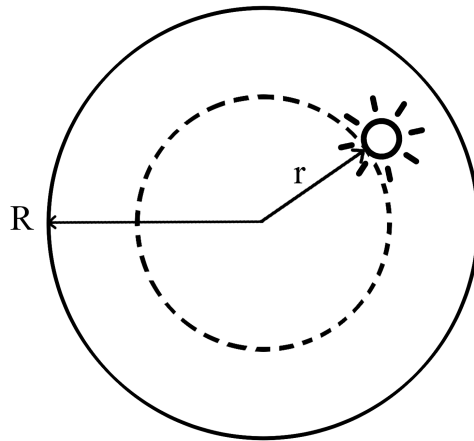


Figure 4: **Scheme on the calculation of the radial distance of the Sun (r) from the center of the image R .** By multiplying the distance from the center of the image R by the normalized angle θ_{norm} that gives the separation between the camera point of view axis and the position of the sun plus a non-linear correction, it is obtained the final radial distance of the sun r needed to calculate the positions in the sky image (x, y) .

Due to slight tilts in the camera installation, a systematic vertical shift in the sun's projected position is observed as a function of elevation. To correct that, a vertical offset is applied as:

$$\delta_y = y_0 + m \cdot (\epsilon - 45^\circ), \quad (2.15)$$

with:

- $y_0 = -3.0$ the vertical offset at 45° elevation,
- $m = 0.03$ an empirical correction slope.

Although the standard polar-to-Cartesian conversion is expressed as $x = r \cos(\theta)$, $y = r \sin(\theta)$ [23], in the image coordinate system (with its origin in the upper-left corner, the Y -axis increasing downwards, and angles measured clockwise from the vertical), the roles of sine and cosine are swapped for the X and Y coordinates, respectively, compared to the standard mathematical convention (see Fig. 5). This is because the reference axis for the angle is the vertical (Y -axis) instead of the horizontal (X -axis), and the direction of increasing angle is clockwise instead of counterclockwise.

Standard polar-to-Cartesian
coordinate system

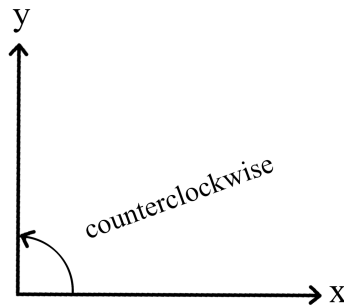


Image coordinate system

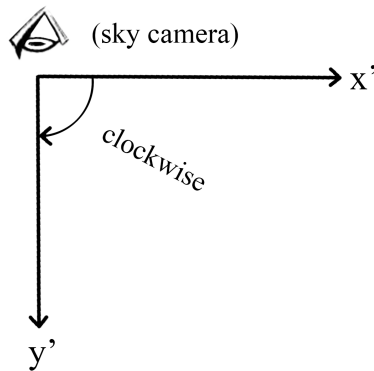


Figure 5: **Scheme on the standard polar-to-Cartesian coordinate system compared with the image coordinate system used in this work.** In the standard polar-to-Cartesian coordinate system, y points upward and the angles are calculated as a counterclockwise direction, whereas in the image coordinate system, the origin starts in the upper-left corner of the image, i.e., y' axis pointing downwards and clockwise.

Thus, the following adapted equations are used:

$$x = C + r \cdot \sin(\theta), \quad (2.16)$$

$$y = C + r \cdot \cos(\theta) + \delta_y, \quad (2.17)$$

where:

- C is the image center (e.g., 32 px for 64×64),

- r is the radial distance,
- θ is the solar projection angle in radians,
- δ_y is the vertical correction defined above.

The resulting (x, y) coordinates represent the projected sun position at each timestamp. As can be seen in Fig. 6, the sun position is not perfectly aligned with the actual position of the sun in the sky despite having made some empirical corrections in the sun position equations, as there were trade-offs depending on whether the sun is in the center of the image or in the corners.

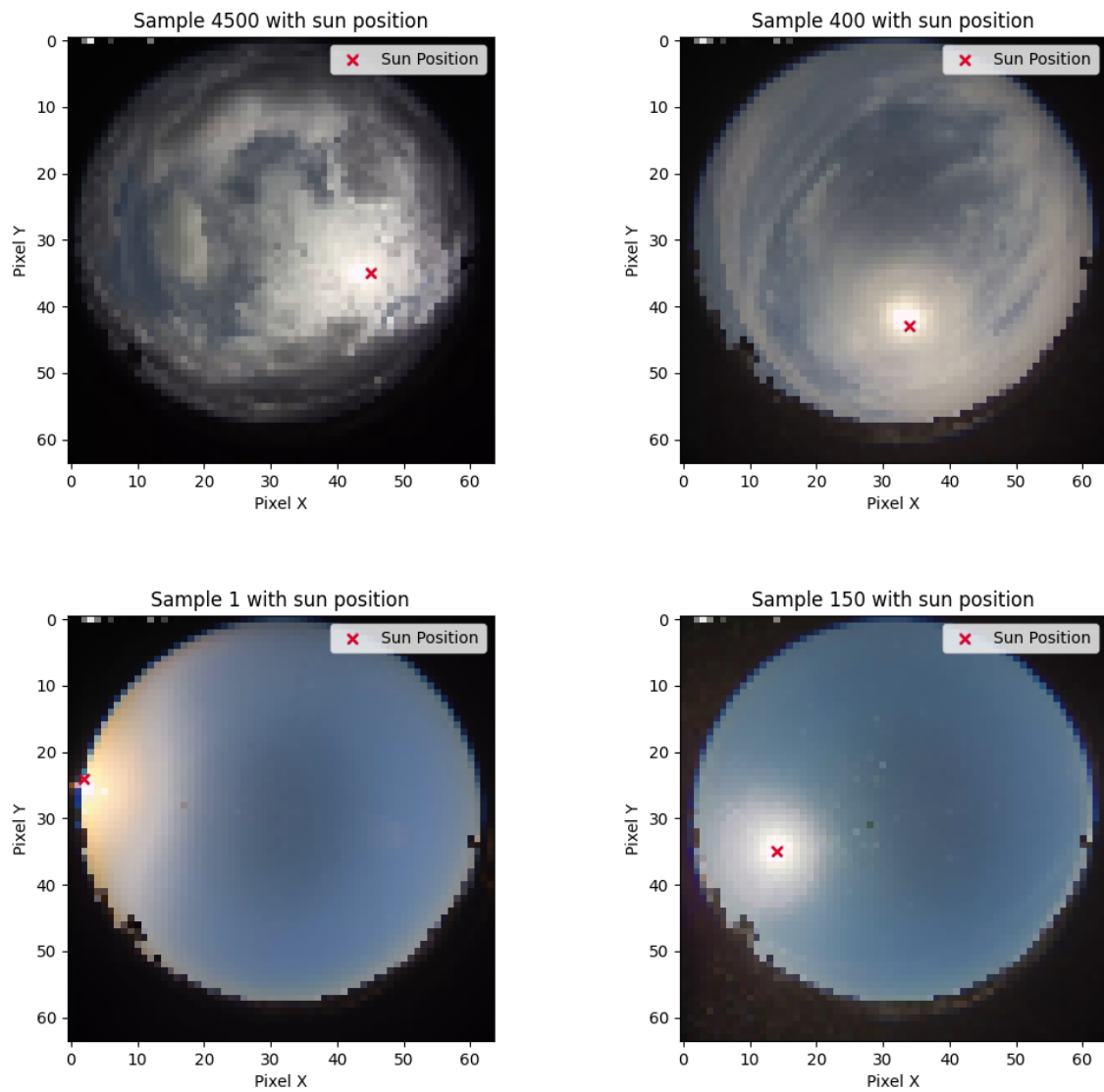


Figure 6: **Sun position estimation.** Example of four different sun position observations where the actual sun position in the sky image is displayed by means of a \times red mark.

Chapter 3

Model

3.1 Differences between nowcast and forecast

Before addressing the technical details of the work, it is worth clarifying the differences between *nowcasting* and *forecasting*, in order to correctly contextualize the approaches used and improve understanding of the models employed.

In meteorology, *nowcasting* and *forecasting* are two complementary but clearly differentiated strategies in terms of their time horizon, the methods used, and the purpose of their predictions.

Nowcasting is a very short-term weather forecasting technique, typically covering between 0 and 3 hours, although some sources extend this range up to 6 ones. Its objective is to provide immediate and highly accurate information on atmospheric phenomena that are occurring or will occur in the next few minutes. To achieve this, it relies on real-time observations from radars, satellites, weather stations, and other sources with high spatial and temporal resolution [24, 25, 26]. This technique is particularly useful for anticipating rapidly evolving events such as thunderstorms, heavy precipitation, or fog. It generally operates on very local scales (microscale and gamma mesoscale), with spatial resolutions down to a few kilometers.

In contrast, traditional *forecasting* addresses broader time horizons, from several

hours to multiple days or even weeks. It is typically based on numerical weather prediction (NWP) models that simulate the behavior of the atmosphere using initial conditions and physical equations. These models assimilate large volumes of data and provide valuable forecasts for planning in sectors such as energy, logistics, agriculture, and tourism. Forecasting usually operates on larger spatial scales (mesoscale and synoptic scale) and does not require such frequent updates as nowcasting.

In the context of the SKIPP'D dataset and the study presented by [17], the terms *nowcasting* and *forecasting* are used with more specific meaning, focused on predicting PV power generation based on sky images.

Nowcasting refers here to the instantaneous estimation of solar power generation using a sky image captured at the same time. In this case, a model learns a function of the form:

$$f_N : I_i \rightarrow P_i, \quad (3.1)$$

that is, a direct mapping from sky image I_i to PV output P_i at the same instant.

Forecasting, on the other hand, aims to predict PV output 15 minutes into the future, based on a sequence of sky images and PV values from the past 15 minutes. This formulation follows:

$$f_F : (I, P)_{t_i-H:\delta:t_i} \rightarrow P_{t_i+T}, \quad (3.2)$$

where $H = 15$ minutes of history, $\delta = 1$ minute sampling interval, and $T = 15$ minute prediction horizon.

In practical implementations, this approach involves using 16 consecutive images (from $t - 15$ to t) and their corresponding 16 PV values as input context to predict PV output at $t + 15$. The use of 16 steps arises from the inclusion of the current time step t in the historical window.

Although the term *short-term forecast* is used in the article, it should not be confused with its more general use in meteorology, where it may refer to forecasts up to 12

or 24 hours ahead. In this specific case, *short-term* refers strictly to a 15-minute prediction horizon, placing the task within the lower end of the *very short-term forecasting* range, and even overlapping with some classical definitions of nowcasting.

Therefore, although the task is labeled as forecasting, the time horizon is extremely short, and the main distinction between nowcasting and forecasting lies not in the prediction time itself, but in whether or not historical context is included in the input sequence.

3.2 SUNSET_nowcast model architecture

The SUNSET_nowcast base model performs nowcasting, i.e., predicting PV power at the current moment based on a fisheye image of the sky. The model seeks to directly correlate the visual state of the sky with the power generated in real time.

It is a direct inference model that maps visual patterns to continuous solar production values, using a design based on deep CNN.

Hyperparameters

- **Number filters:** 24

Each filter learns to detect a specific visual pattern (e.g., edges, textures, shapes, etc.). More filters increase the capacity to extract features.

- **Kernel size:** 3×3

This is the window that slides across the image to apply the convolution operation.

- **Activation function:** ReLU (Rectified Linear Unit)

Applied after each convolutional and dense layer. It introduces non-linearity, accelerates training, and helps mitigate the vanishing gradient problem.

- **Batch normalization:** Applied after each convolutional layer

It normalizes the layer inputs to have zero mean and unit variance, which

stabilizes learning and enables higher learning rates. (Note: not used after dense layers in this model.)

- **Pooling window size:** 2×2

Pooling reduces spatial resolution while retaining the most relevant features, decreasing the number of parameters.

- **Number of units in dense layers:** 1,024

The fully connected layers contain 1,024 neurons. The more units, the greater the learning and representation capacity, although with a higher risk of overfitting.

- **Dropout rate:** 0.4

During training, 40% of the neurons are randomly deactivated in each pass. This acts as a regularizer and helps prevent overfitting.

- **Number of epochs:** 200

This is the maximum number of times the model sees the entire training set. It can be stopped earlier by early stopping if performance stabilizes.

- **Cross-validation:** 10 partitions (10-fold)

The training data is split into 10 subsets; 9 are used for training and 1 for validation in each iteration.

- **Batch size:** 256

It tells how many samples are processed together before each weight update.

- **Learning rate:** 3×10^{-6}

Controls how much the model weights are adjusted per update. A small rate allows for gradual/stable convergence, which can be a benefit in tasks such as PV prediction, due to its high variability.

Model structure

The model is structured into three main blocks:

1. **Image input:** the model receives a single RGB image with resolution 64×64 pixels and 3 color channels.
2. **Visual feature extraction:** the image passes through two convolutional blocks. Each block includes:
 - A Conv2D layer with ReLU activation,
 - A BatchNormalization layer,
 - A MaxPooling2D layer.

These blocks extract features obtained from the images, which could be textures, cloud edges, cloud geometries, for example.

3. **Fully connected layers:** the output is flattened and passed through two dense layers with 1,024 units, ReLU activation and 40% dropout. A final dense layer with a single unit produces the predicted PV output (in kW).

Figure 7 shows the overall structure of the nowcast and forecast models, where it can be seen the differences in how input data is fed into each architecture.

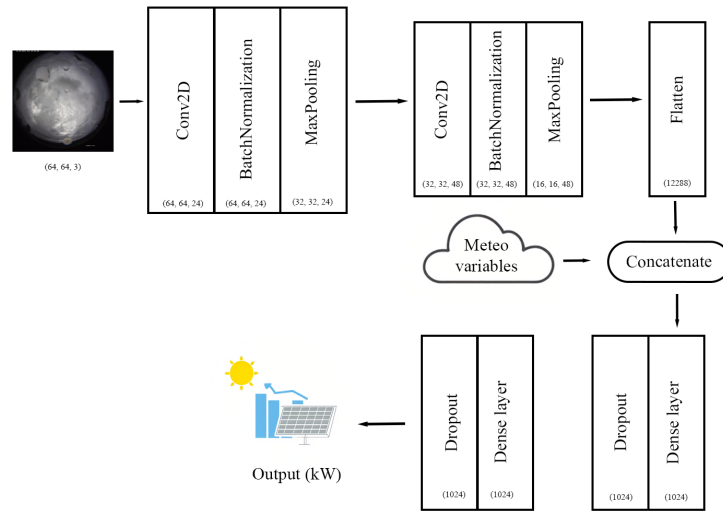
Data loading and optimization

The sky images and PV power data are stored in an HDF5 file (fields: `images_log`, `pv_log`) from the SKIPP'D dataset [17]. Then, the function `data_loader` reads data according to specified indices, normalizes the images to $[0, 1]$, and packages them into batches using the `tf.data` API.

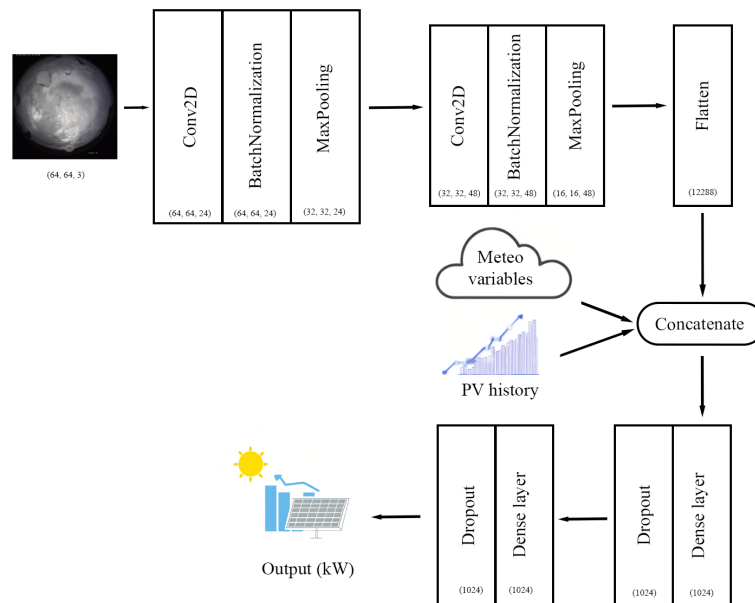
The model is optimized using the Adam algorithm [27], a stochastic gradient descent method that adaptively estimates the first and second-order moments of the gradient to improve stability and convergence.

3.3 *SUNSET__forecast model architecture*

The base model `SUNSET_forecast` performs the forecasting task, i.e., predicting PV power 15 minutes into the future based on a sequence of sky images and the



(a) Structure of the SUNSET model for the nowcast task.



(b) Structure of the SUNSET model for the forecast task.

Figure 7: **Architectures of the SUNSET model for both nowcast (a) and forecast (b) tasks.** The diagram shows the different layers, the input variables fed into the model, and the output shapes at each stage.

history of PV production. The goal of the model is to anticipate the evolution of photovoltaic solar energy production, taking into account both the visual situation of the sky and the recent trend in PV power generation.

The internal architecture of `SUNSET_forecast` is identical to that of the `SUNSET_nowcast` model in terms of the design of the convolutional layers, dense layers, and hyperparameters, except for the number of cross-fold validation, which has been reduced to 5 in order to speed up the result. However, it has two key differences:

- **Sky image input:** instead of a single RGB image ($64 \times 64 \times 3$), the forecast model uses a sequence of 16 RGB images, which are concatenated along the channel axis, forming a $64 \times 64 \times 48$ input. This allows the model to capture recent temporal information about the evolution of the sky.
- **Additional PV history input:** the forecast model introduces a second input vector with the 16 values corresponding to the power generated in the previous 16 minutes. This vector is concatenated with the output of the convolutional network before the dense layers.

Data loading and optimization

The sky image sequences, PV history, and target values are stored in an HDF5 file (`images_log`, `pv_log`, `pv_pred`) from the SKIPP'D dataset [17]. The model uses the `data_loader` function to load and prepare these data, following the same procedure as used in the `SUNSET_nowcast` model.

As in the nowcast model, the model is also optimized using the Adam algorithm [27].

3.4 Introduction of variables into the models

In order to improve photovoltaic predictions for the nowcast and forecast model, different variables are introduced. Below is a detailed explanation of how these variables are added to the model.

3.4.1 Single meteorological variable

In an initial configuration, the model is trained with images of the sky as the main input, to which a single meteorological variable is added. This variable is incorporated into the model as a 1D vector of 4 elements, corresponding to the four defined spatial points: NW, NE, SW, and SE. This vector is concatenated with the features extracted from the image just after the flatten operation (which converts the image tensor into a 1D vector) and before the fully connected layers. In this way, the physical information is incorporated directly into the latent feature space.

3.4.2 Multiple meteorological variables

In a second configuration, multiple meteorological variables are incorporated, each with their respective values at the four cardinal points. All variables are normalized individually and then concatenated into a single one-dimensional vector. The resulting vector (of length $4 \times$ number of variables) is concatenated as in the previous case: after the image is flattened and before the first fully connected layer.

3.4.3 Sun position

The position of the Sun is represented as a pair of normalized coordinates (`sun_x`, `sun_y`), with shape 1D and two values, which is also concatenated to the model just after the image is flattened, following the same procedure as in the previous cases.

3.5 Testing and evaluation metrics (RMSE, MAE)

In line with prior work using the SKIPP'D dataset for image-based solar forecasting, this study adopts Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as evaluation metrics. These metrics quantify the difference between model

predictions (\hat{y}_i) and ground truth values (y_i), and are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (3.3)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (3.4)$$

These metrics are used to evaluate the performance of in both *nowcasting* (short-term prediction) and *forecasting* (future prediction), as done in previous work such as SUNSET [17].

Nowcasting: The models are evaluated using multiple training repetitions. For each repetition, the model with the lowest validation RMSE is selected, and its predictions on the test set are stored. These predictions are then averaged across all repetitions to produce an ensemble forecast. The final performance is reported as the RMSE of the ensemble prediction:

$$\text{RMSE}_{\text{ensemble}} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(y_i - \frac{1}{K} \sum_{k=1}^K \hat{y}_i^{(k)} \right)^2}, \quad (3.5)$$

where K is the number of models included in the ensemble.

Forecasting. A similar procedure is followed: each model is evaluated individually, and predictions are averaged to obtain the ensemble output. Final performance is again reported using the ensemble RMSE.

Cloudy/Sunny evaluation: Additionally, the performance is analyzed separately for sunny and cloudy conditions. The overall RMSE/MAE are computed as a weighted average of the individual values as:

$$\text{RMSE}_{\text{total}} = \sqrt{\frac{n_{\text{sunny}} \cdot \text{RMSE}_{\text{sunny}}^2 + n_{\text{cloudy}} \cdot \text{RMSE}_{\text{cloudy}}^2}{n_{\text{total}}}}, \quad (3.6)$$

$$\text{MAE}_{\text{total}} = \frac{n_{\text{sunny}} \cdot \text{MAE}_{\text{sunny}} + n_{\text{cloudy}} \cdot \text{MAE}_{\text{cloudy}}}{n_{\text{total}}}. \quad (3.7)$$

Chapter 4

Results

This section presents the results obtained from the nowcast and forecast models. To assess whether the inclusion of meteorological data leads to an improvement or worsening of the results, the RMSE and MAE metrics have been used. To provide an overall view of how the errors improve or worsen, error percentage changes have been calculated. All percentage changes in RMSE and MAE are computed relative to the baseline values reported in the study [17] on which this work is based.

4.1 Nowcast results

In order to evaluate the performance of the nowcast model with the different meteorological variables, those have been fed into the model, and then RMSE and MAE errors have been calculated. In Table 2 results are summarized.

Sunny days

The variables that contribute most to improving the forecast on sunny days are: `i10fg`, `ssrd`, `str`, `strd`, `wind100` (although it does not reduce the MAE), `sun position`, as well as the combinations `i10fg + wind100` and `i10fg + wind100 + sun position` (again, with no improvement in the MAE). The variables that achieve the greatest reduction in error are `str`, with a decrease in RMSE of -7.214% and

Data variable	RMSE	MAE
	(sunny / cloudy / global)	(sunny / cloudy / global)
SUNSET nowcast [17]	0.804 / 3.335 / 2.428	0.657 / 2.337 / 1.499
RSUNSET*[16]	0.790 / 3.300 / 2.400	0.650 / 2.300 / 1.480
TCC	0.783 / 3.343 / 2.430	0.643 / 2.345 / 1.496
i10fg	0.764 / 3.305 / 2.401	0.636 / 2.302 / 1.471
SSRD	0.768 / 3.355 / 2.436	0.640 / 2.360 / 1.502
STR	0.746 / 3.369 / 2.442	0.612 / 2.370 / 1.493
TSR	0.828 / 3.332 / 2.430	0.678 / 2.327 / 1.504
STRD	0.732 / 3.316 / 2.404	0.600 / 2.319 / 1.461
SP	0.847 / 3.333 / 2.434	0.708 / 2.336 / 1.524
FDIR	0.821 / 3.358 / 2.447	0.681 / 2.370 / 1.527
Wind100m	0.798 / 3.328 / 2.423	0.673 / 2.320 / 1.498
Sun position	0.791 / 3.372 / 2.452	0.645 / 2.374 / 1.512
i10fg+wind100+strd	0.864 / 3.283 / 2.403	0.756 / 2.266 / 1.513
i10fg+wind100	0.782 / 3.314 / 2.410	0.660 / 2.285 / 1.474
i10fg+strd+wind100+tcc	0.820 / 3.297 / 2.405	0.699 / 2.273 / 1.488
i10fg+strd+str+wind100	0.818 / 3.300 / 2.406	0.703 / 2.281 / 1.493
i10fg+wind100+sun position	0.799 / 3.297 / 2.401	0.689 / 2.276 / 1.484
i10fg+wind100+sun position+strd	0.826 / 3.302 / 2.409	0.707 / 2.274 / 1.493

Table 2: **Quantitative comparison of RMSE and MAE metrics** (sunny / cloudy / global) for different combinations of meteorological variables. *referring to RSUNSET + sun position + SAMPI results [16]. In bold are marked the variables that give the best results.

MAE of -6.849%, and **strd**, with improvements of -8.955% (RMSE) and -8.676% (MAE).

Cloudy days

In cloudy conditions, the variables that improve errors are: **i10fg**, **tsr**, **strd**, **wind100**, and various combinations such as **i10fg + strd + wind100**, **i10fg + wind100**, **i10fg + strd + wind100 + tcc**, **i10fg + strd + str + wind100**, **i10fg + wind100 + sun position**, and **i10fg + wind100 + sun position + strd**. The combinations **i10fg + strd + wind100 + tcc** and **i10fg + wind100 + sun position + strd** stand out in particular, with improvements in RMSE of -1.139% and MAE of -2.739% in the first case, and -1.139% and -2.610% in the second case.

Overall results

In general (without distinguishing between clear or cloudy conditions), the variables that most improve the prediction are: `tcc` (although it does not reduce the RMSE), `i10fg`, `strd`, `wind100`, as well as combinations such as `i10fg + wind100`, `i10fg + strd + wind100 + tcc`, `i10fg + strd + str + wind100`, `i10fg + wind100 + sun position`, and `i10fg + wind100 + sun position + strd`. The variables with the greatest reduction in error are `i10fg` (-1.112% RMSE, -1.868% MAE), `strd` (-0.988% RMSE, -2.535% MAE), `i10fg + wind100` (-0.741% RMSE, -1.668% MAE), and `i10fg + wind100 + sun position` (-1.112% RMSE, -1.001% MAE).

4.2 Forecast results

Once the combinations of variables that improve the prediction results in the `nowcast` task were identified, it was decided to use those same variables in the `forecast` model. This is because training the model with a +15-minute horizon requires significantly more computing time than `nowcast` (more than double), even after reducing cross-validation from 10 to 5 partitions. For this reason, it was considered more appropriate to use only those combinations of meteorological variables that performed best in `nowcast`, in order to provide more direct and efficient results.

In Table 3 it can be seen that both the `SUNSET_forecast**` — using the new `forecast_dataset` file — and the versions of the same model with different meteorological variables added achieve a significant improvement in the RMSE for cloudy days and overall, as well as in the MAE for cloudy days. Specifically, the reduction in error on cloudy days achieves an improvement of approximately 35% to 42% in the RMSE, 9% to 20% in the MAE, and 29% to 34% in the overall RMSE.

The combination that most improves cloudy days is `strd`, `i10fg + wind100 + sun position`, both in terms of RMSE and MAE, followed by the base model `SUNSET_forecast**`. In terms of overall RMSE, the greatest reduction in error is provided by the `SUNSET_forecast**` model with `strd` additional variable, with an improvement of 20.396%, closely followed by the model `SUNSET_forecast**` without meteorological variables (17.492%),

Data variable	RMSE	MAE
	(sunny / cloudy / global)	(sunny / cloudy / global)
SUNSET forecast [17]	0.610 / 4.270 / 3.030	0.500 / 2.950 / 1.710
RSUNSET* [16]	8.620 / 7.610 / 8.310	7.550 / 5.750 / 6.650
SUNSET forecast**	2.461 / 2.539 / 2.500	2.109 / 1.938 / 2.023
i10fg**	2.589 / 2.615 / 2.603	2.197 / 2.037 / 2.117
STRD**	2.331 / 2.489 / 2.412	1.924 / 1.882 / 1.903
i10fg+wind100**	2.831 / 2.669 / 2.751	2.402 / 2.056 / 2.229
i10fg+wind100+sun position**	2.570 / 2.494 / 2.532	2.269 / 1.934 / 2.101
i10fg+wind100+strd**	3.056 / 2.762 / 2.912	2.606 / 2.088 / 2.345

Table 3: **Quantitative comparison of RMSE and MAE metrics (sunny / cloudy / global) for different combinations of meteorological variables.** *referring to RSUNSET + sun position + SAMPI results [16]. **referring to the results of the forecast SUNSET base model [17] using the created `forecast_dataset.hdf5` for this work. In **bold** are marked the variables that give the best results.

and the combination `i10fg + wind100 + sun position`, which achieves an improvement of 16.436%.

Finally, comparing these results with those obtained using the `RSUNSET*` model, both the base `SUNSET_forecast**` model and its variants with additional meteorological variables outperform its results, showing a better overall prediction.

Figure 8 presents the predictions of the nowcast and forecast models across different test days, together with the corresponding ground truth. As can be seen, our solution is quite accurate even when rapid changes in power estimation occur.

The computational cost of training the `SUNSET_nowcast` model, evaluated over ten repetitions, has resulted in a total training time of approximately 1.3 hours. Each repetition took an average of 7–8 minutes, with each lasting around 41 seconds and each step required approximately 0.033 seconds. The training of the `SUNSET_forecast` model, which has been performed over five repetitions, has required a total of approximately 2.7 hours. Each repetition has lasted around 32.5 minutes, with average epoch and step durations of 165 seconds and 0.41 seconds, respectively.

All model training and evaluation in this work were trained on a system equipped with an NVIDIA GeForce RTX 3090 GPU and 16-core CPU.

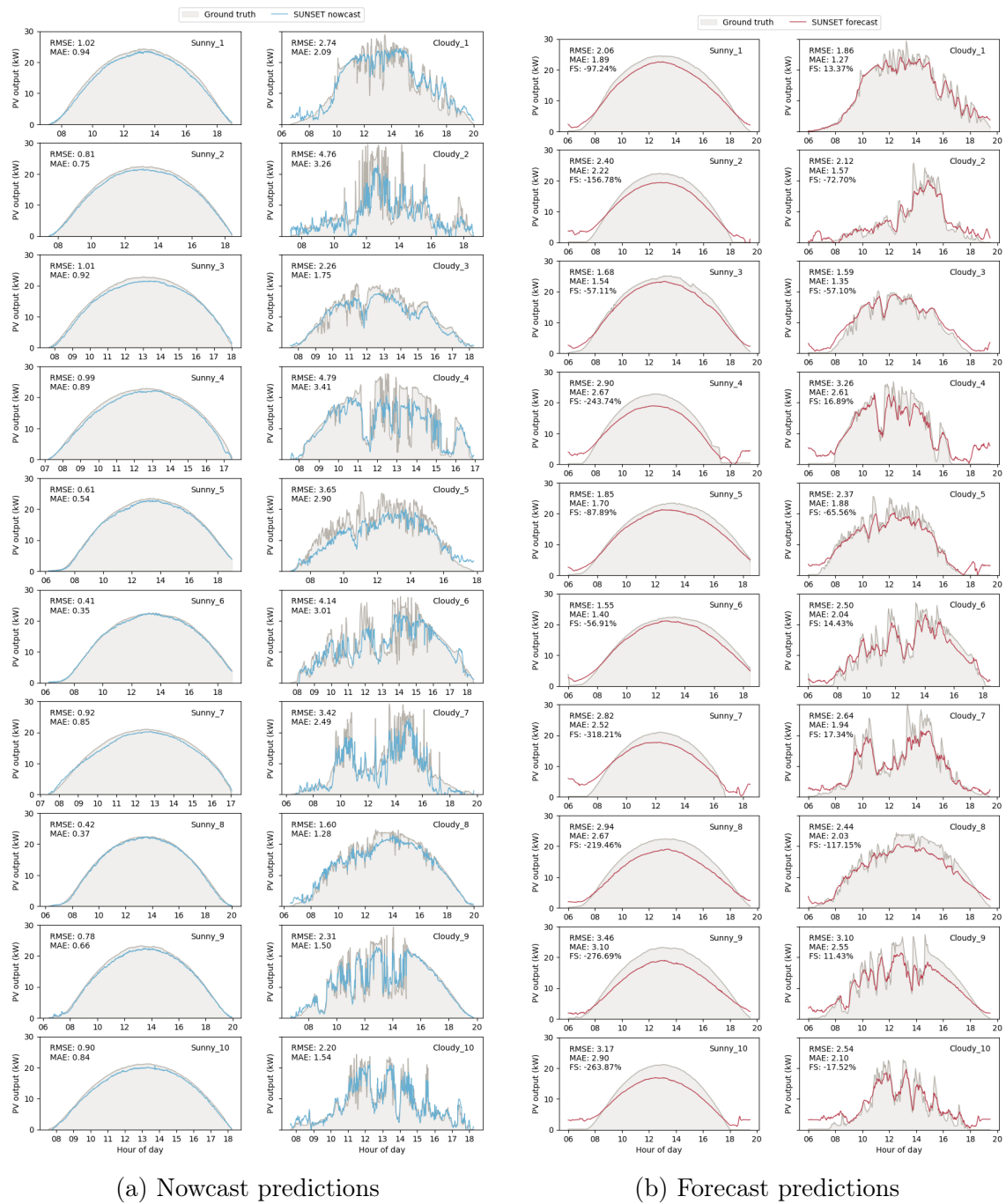


Figure 8: Nowcast (a) and forecast (b) predictions with *i10fg*, *wind100* and *sun position variable* inputs for different test days. Grey area corresponds to the ground truth and blue and red lines represent the result of the model prediction for nowcast and forecast, respectively.

Chapter 5

Conclusion

Based on the results obtained in the nowcast task, some unexpected results are observed. For example, the variable ‘top net solar radiation’ (**tsr**), which measures net solar radiation at the top of the atmosphere, curiously improves the prediction on cloudy days. Intuitively, one would expect a radiation variable to improve the prediction model on clear days, but since **tsr** captures the total amount of solar radiation entering and leaving the entire atmospheric system, an attenuation of this energy at the top of the atmosphere could be an indicator of the presence and density of clouds, and thus improve the performance of the nowcast model by learning information about how much solar radiation is intercepted before reaching the surface. On the one hand, the improvement in RMSE and MAE for the other radiation variables (**strd**, **ssrd**, and **str**) during sunny days could be explained by the fact that an absence of clouds leads to a simplification of the radiative state, and therefore, by interacting in a more direct and less complex way with the ground, it translates into more accurate forecasts.

On the other hand, surface pressure (**sp**) slightly improves the forecast on cloudy days, but shows no improvement on sunny days or in the overall set. This variable was selected because of its relationship with wind generation (due to the pressure gradient associated with isobars), since wind (**i10fg** and **wind100**) had shown an improvement in prediction. However, in this case, there has been no significant

improvement.

Another noteworthy aspect is that the combination `i10fg + wind100 + strd` significantly worsens both the RMSE and the MAE on sunny days and overall, even though these variables separately give positive results. This suggests that the model may have difficulty correctly capturing the interaction between these variables. Similarly, the combination `i10fg + wind100 + sun position` improves the RMSE compared to `i10fg + wind100`, while `sun position` alone worsens the results. This could indicate that the added value of the Sun position depends on the context provided by other already effective predictor variables, such as wind.

The fact that the `SUNSET_forecast**` model improves both the RMSE and the MAE on cloudy days, as well as the overall RMSE, could be due to the use of a different dataset, since the test days used do not exactly match those of the previous study. However, theoretically this should not have a significant effect, since the typical behavior of a sunny or cloudy day should be equally reproducible in both models. Therefore, a more likely explanation is the way in which the variable `pv_pred` has been generated, since in this version interpolation and smoothing techniques have been applied that may have favored more stable learning of the model.

Given that the model improves specifically on cloudy days and not on sunny days is also an interesting finding, since, in principle, clear days are more predictable due to the absence of interference from cloud cover. This difference suggests that the model has learned to better manage the complexity and variability present in cloudy scenarios.

In any event, there is a general improvement in the RMSE but not in the MAE. This difference could be explained by the nature of both errors: while the MAE measures the average of the absolute errors and gives the same weight to all errors, the RMSE penalizes large errors more heavily. Therefore, if the model has managed to reduce the most extreme errors in particular—even if it maintains similar small errors—the RMSE will decrease while the MAE will remain virtually unchanged.

As for variable combinations, the one that produces the best results in the prediction

is the `SUNSET_forecast**` model, `i10fg`, and `wind100` together with `sun position`, which coincides with the findings obtained in the `nowcast` task. It is important to note that the overall RMSE is lower in the `SUNSET_forecast**` model without additional variables, compared to the version that includes wind and sun position. This suggests that, by not incorporating additional variables, the model may have better adjusted the prediction for cloudy days, thereby reducing the overall error.

This result is striking, as one would expect that including the sun's position would improve the prediction on sunny days, as is the case in the `nowcast` task. However, the results show that, when combining the sun's position with other variables such as wind, the model improves mainly in cloudy conditions. This suggests that both the `nowcast` and `forecast` models are learning relevant patterns from the interaction between wind and sun position.

Moreover, the variable that most improves cloudy days and the total RMSE is `strd` in the `SUNSET_forecast**` model. This variable, surface long-wave (thermal) radiation downwards, measures the radiation emitted by the atmosphere and clouds towards the Earth's surface. The fact that it has such a positive impact on the forecast task contrasts with the results obtained in the `nowcast`, which improves the RMSE by almost 9% on sunny days vs. 0.6% on cloudy days. This difference could be due to the fact that, in the case of the `forecast`, the model has historical information that allows it to better exploit the thermal pattern associated with cloud cover, while on clear days this signal is less relevant or could even interfere with the model's pattern learning. In contrast, in `nowcast`, where only one sky image is available, downward thermal radiation can act as a more stable indicator in clear sky conditions.

It can therefore be concluded that, in general, the `strd` meteorological variable and the combination of wind (`i10fg` and `wind100`) and solar position (`sun position`) provide the greatest improvements in both prediction tasks, `nowcast` and `forecast`. In addition, for the `forecast` task, the `SUNSET forecast**` model also provides positive results, based on the use of the `forecast_dataset` database created for this work.

5.1 Future Work

One possible area for improvement in future work would be the incorporation of meteorological variables with higher temporal resolution, which would allow the models to extract more useful information in combination with fisheye images and photovoltaic energy records. Another aspect to consider is improving the calculation of the Sun's position in the image, as a greater accuracy in this estimate could translate into more reliable predictions.

Additionally, the use of image brightness as an additional input to the model could be explored. This variable could provide relevant information about the presence or absence of cloud cover around the sun's position, act as an indicator of the sun's height (e.g., associated with low radiation during sunrise or sunset, or maximum production during peak hours), and even serve as a possible alternative to the total cloud cover variable (τ_{cc}). Likewise, brightness could be used as an independent variable, complementary to the position of the Sun, to improve the estimation of photovoltaic production.

List of Figures

- 1 **Latitude and longitude of the four ERA5 grid points** surrounding the camera location in Stanford. These points, spaced at 0.25° resolution (NW, NE, SW, SE), define the area from which ERA5 variables are extracted. The red square indicates the camera position. 9

- 2 **Scheme on the conversion from traditional polar-to-Cartesian coordinate system to our image coordinate system.** In order to get the position of the sun relative to the camera, it is subtracted the angle of the sun α_{sun} to the orientation of the camera $\alpha_{\text{cam}} = 194^\circ$, so that we end up having α_{rel} , the angle which will be used for placing the sun position in the image sky camera. 15

- 3 **Scheme on the transformation from the elevation angle of the sun to the zenith angle.** The pvlib transforms the latitude and longitude to an elevation angle, but as the reference point is the camera, it is needed to convert the coordinate system relative to the camera point of view. 16

- 4 **Scheme on the calculation of the radial distance of the Sun (r) from the center of the image R .** By multiplying the distance from the center of the image R by the normalized angle θ_{norm} that gives the separation between the camera point of view axis and the position of the sun plus a non-linear correction, it is obtained the final radial distance of the sun r needed to calculate the positions in the sky image (x, y) 17

-
- 5 **Scheme on the standard polar-to-Cartesian coordinate system compared with the image coordinate system used in this work.** In the standard polar-to-Cartesian coordinate system, y points upward and the angles are calculated as a counterclockwise direction, whereas in the image coordinate system, the origin starts in the upper-left corner of the image, i.e., y' axis pointing downwards and clockwise. 18
- 6 **Sun position estimation.** Example of four different sun position observations where the actual sun position in the sky image is displayed by means of a \times red mark. 20
- 7 **Architectures of the SUNSET model for both nowcast (a) and forecast (b) tasks.** The diagram shows the different layers, the input variables fed into the model, and the output shapes at each stage. 26
- 8 **Nowcast (a) and forecast (b) predictions with `i10fg`, `wind100` and sun position variable inputs for different test days.** Grey area corresponds to the ground truth and blue and red lines represent the result of the model prediction for nowcast and forecast, respectively. 35

List of Tables

1	Summary of ERA5 variables (fc = forecast; an = analysis). The table shows a summary of the official parameter name of the meteorological variables (1st column), the full name, the source (if it is a analysis or forecast variable), whether the data has or not a step, the units, and the variable type (whether it is an accumulative variable in time or not) in the last column.	12
2	Quantitative comparison of RMSE and MAE metrics (sunny / cloudy / global) for different combinations of meteorological variables. *referring to RSUNSET + sun position + SAMPI results [16]. In bold are marked the variables that give the best results.	31
3	Quantitative comparison of RMSE and MAE metrics (sunny / cloudy / global) for different combinations of meteorological variables. *referring to RSUNSET + sun position + SAMPI results [16]. **referring to the results of the forecast SUNSET base model [17] using the created <code>forecast_dataset.hdf5</code> for this work. In bold are marked the variables that give the best results.	33

Bibliography

- [1] Global Energy Review 2025 – Analysis - IEA — [iea.org. https://www.iea.org/reports/global-energy-review-2025](https://www.iea.org/reports/global-energy-review-2025). [Accessed 30-03-2025].
- [2] Barnes, A. K., Balda, J. C. & Hayes, J. K. *Modelling PV Clouding Effects Using a Semi-Markov Process with Application to Energy Storage*. URL <http://energy.uark.edu/pv>.
- [3] Kang, P. & Li, J. Short-term photovoltaic power forecasting based on vmd-kpca-lstm. In Sun, F., Wang, H., Long, H., Wei, Y. & Yu, H. (eds.) *Proceedings of the 3rd International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM2024)*, 372–386 (Springer Nature Singapore, Singapore, 2025).
- [4] Kim, J., Obregon, J., Park, H. & Jung, J. Y. Multi-step photovoltaic power forecasting using transformer and recurrent neural networks. *Renewable and Sustainable Energy Reviews* **200** (2024).
- [5] Lorenz, E., Kühnert, J. & Heinemann, D. *Overview of Irradiance and Photovoltaic Power Prediction*, 429–454 (Springer New York, New York, NY, 2014). URL https://doi.org/10.1007/978-1-4614-9221-4_21.
- [6] Lin, F., Zhang, Y. & Wang, J. Recent advances in intra-hour solar forecasting: A review of ground-based sky image methods (2023).
- [7] Fabel, Y. *et al.* Combining deep learning and physical models: A benchmark study on all-sky imager-based solar nowcasting systems. *Solar RRL* **8** (2024).

- [8] Zahraoui, Y., Korotko, T., Mekhilef, S. & Rosin, A. Ann-lstm based tool for photovoltaic power forecasting. In *4th International Conference on Smart Grid and Renewable Energy, SGRE 2024 - Proceedings* (Institute of Electrical and Electronics Engineers Inc., 2024).
- [9] Zhang, X. *et al.* Prediction interval estimation and deterministic forecasting model using ground-based sky image. *IEEE Transactions on Industry Applications* **59**, 2210–2224 (2023).
- [10] Zang, H. *et al.* Improving ultra-short-term photovoltaic power forecasting using a novel sky-image-based framework considering spatial-temporal feature interaction. *Energy* **293** (2024).
- [11] Kong, W., Jia, Y., Dong, Z. Y., Meng, K. & Chai, S. Hybrid approaches based on deep whole-sky-image learning to photovoltaic generation forecasting. *Applied Energy* **280** (2020).
- [12] Kuo, W. C., Chen, C. H., Chen, S. Y. & Wang, C. C. Deep learning neural networks for short-term pv power forecasting via sky image method. *Energies* **15** (2022).
- [13] Lu, Z., Chen, W., Yan, Q., Li, X. & Nie, B. Photovoltaic power forecasting approach based on ground-based cloud images in hazy weather. *Sustainability (Switzerland)* **15** (2023).
- [14] Hu, K., Wang, L., Li, W., Cao, S. & Shen, Y. Forecasting of solar radiation in photovoltaic power station based on ground-based cloud images and bp neural network. *IET Generation, Transmission and Distribution* **16**, 333–350 (2022).
- [15] Zhen, Z. *et al.* Deep learning based surface irradiance mapping model for solar pv power forecasting using sky image. *IEEE Transactions on Industry Applications* **56**, 3385–3396 (2020).
- [16] Berresheim, A. & Agudo, A. *Photovoltaic power forecasting using sky images and sun motion*, 4260–4264 (Institute of Electrical and Electronics Engineers (IEEE), 2023). URL <http://hdl.handle.net/2117/409401>.

- [17] Nie, Y. *et al.* Skipp'd: a sky images and photovoltaic power generation dataset for short-term solar forecasting (2022). URL <http://arxiv.org/abs/2207.00913>.
- [18] ECMWF Reanalysis v5 — ecmwf.int. <https://www.ecmwf.int/en/forecasts/dataset/ecmwf-reanalysis-v5>. [Accessed 31-03-2025].
- [19] ECMWF — ecmwf.int. <https://www.ecmwf.int/>. [Accessed 31-03-2025].
- [20] C3S. ERA5 hourly data on single levels from 1940 to present (2018).
- [21] How Big is a Degree? — sco.wisc.edu. <https://www.sco.wisc.edu/2022/01/21/how-big-is-a-degree/>. [Accessed 02-07-2025].
- [22] Sahin, C. Comparison and calibration of mobile phone fisheye lens and regular fisheye lens via equidistant model. *J. Sens.* **2016**, 1–11 (2016).
- [23] Galadí-Enríquez, D., Soler, E. M., Martínez, V. J. & Miralles, J. A. *Astronomia fonamental (2a edició)* (Publicacions de la Universitat de València, 2008).
- [24] WMO OSCAR &A0; Application Area: 2.3 Nowcasting / Very Short-Range Forecasting — space.oscar.wmo.int. https://space.oscar.wmo.int/applicationareas/view/2_3_nowcasting_very_short_range_forecasting. [Accessed 03-07-2025].
- [25] Nowcasting — metoffice.gov.uk. <https://www.metoffice.gov.uk/research/weather/space-applications-and-nowcasting/nowcasting>. [Accessed 03-07-2025].
- [26] Bañón Peregrín, L. M. *Panorama del nowcasting en los países de nuestro entorno*, 5–12 (Agencia Estatal de Meteorología, 2019). URL <http://dx.doi.org/10.31978/639-19-010-0.005>.
- [27] Team, K. Keras documentation: Adam — keras.io. <https://keras.io/api/optimizers/adam/>. [Accessed 04-07-2025].

Appendix A

Creation of Forecast dataset

In order to train the `SUNSET forecast` model, the corresponding `.hdf5` file had to be created, as the file was not uploaded to the repository, and the alternative methods available (creating it from scratch, downloading the code from a database, building the forecast file from the nowcast file) were incomplete or lacked data.

The methodology followed was to first download the `forecast_dataset.hdf5` from a repository (<https://torchgeo.readthedocs.io/en/stable/api/datasets.html#skipp-d>). This file did not have the necessary `pv_pred` that serve as targets for training the model, and the test file, `times_test.npy`, was also incomplete (only for cloudy days).

Starting from this file, the objective is to recreate `times_test.npy` with 10 test days for sunny days and 10 for cloudy days, and then calculate `pv_pred` for training/validation and testing.

- To create `time_test.npy`, it is necessary to save the timestamps for later testing in the model. From the manually selected dates for cloudy and sunny days, the timestamps available for those dates are extracted. They are then saved in a numpy file.
- Since the `times_trainval.npy` file contained some of the dates from `times_test`, these have been removed from the `times_trainval` file.

-
- To create the new `forecast_dataset.hdf5` file, both the images and the photovoltaic production data are processed, creating temporal stacks of 16 consecutive images and generating the target predictions at 15-minute intervals.

To do this:

- The photovoltaic production data `pv_output.csv` is loaded, and the data is reindexed at 1-minute intervals to fill in any gaps that may exist. To correspond to the criteria used in [17], negative values are removed, and missing values are interpolated using linear interpolation functions. This ensures that even if there is no exact measurement for a certain timestamp, an approximate estimate of `pv` can be obtained.
- Next, the `times_test` file is taken and the valid timestamps that have the 16 consecutive images available are filtered to form each temporal stack. For each timestamp, a stack of 16 images is loaded, going back minute by minute. Each image is resized to 64 x 64 and converted from BGR to RGB space to maintain consistency (`images_log`). Simultaneously, the historical photovoltaic production values corresponding to each image in the stack are interpolated (`pv_log`), and the target prediction (`pv_pred`) is calculated for time $t + 15$ minutes.
 - * To generate `pv_pred`, a Savitzky-Golay filter is first applied to smooth the `pv_pred` values. This technique has been applied to eliminate unrealistic high-frequency fluctuations. This technique has been applied because in previous versions, very pronounced drops and peaks were observed, which did not represent the actual behavior.
 - * In addition to smoothing, limits are implemented on sudden drops in production, setting a maximum of 0.6 kW per minute.
 - * Finally, values corresponding to nighttime hours (before 6 AM) are forced to 0, and a minimum production is established during dawn hours (6 and 8 AM) to try to reproduce the gradual behavior of the start of generation.
- For the training and validation set, the process is more efficient since it

takes advantage of the existing data from the downloaded `forecast_dataset`.

The datasets from `images_log` and `pv_log` are copied directly, and the photovoltaic predictions `pv_pred` are generated using the same methodology as for the test dataset.

- Since the original implementation in [17] uses a sampling frequency of every 2 minutes instead of every 1 minute, the final `forecast_dataset` is re-sampled.