



UNIVERSITY OF LIÈGE

Big Data Project - Report

Renewable Energy Production Forecast

Yann CLAES
Gaspard LAMBRECHTS
François ROZET

May 10, 2020

Contents

I Solar power	2
1 Photovoltaic panels mapping	2
1.1 Wallonia aerial images	2
1.2 Detection model	2
1.2.1 DeepSolar	2
1.2.2 U-Net	3
1.3 Trainings set	3
1.3.1 Data augmentation	3
1.4 Results	4
1.4.1 California	4
1.4.2 Fine tuning	5
1.4.3 Province of Liège	6
1.5 Limitations and possible improvements	7
2 Photovoltaic power production	8
2.1 Overview of solar power forecasting	8
2.2 Data collection and processing	9
2.2.1 Photovoltaic production data	9
2.2.2 Irradiance data	9
2.3 Models description and practical details	10
2.3.1 Panelwise model – python/solar_panelwise.py	10
2.3.2 Provincial model – python/solar_provincial.py	13
2.4 Analysis of the models accuracy	14
2.5 Limitations and possible improvements	19
II Wind power	23
3 Wind power production	23
3.1 Overview of wind power forecasting	23
3.2 Data and computation flow	25
3.2.1 Data	25
3.2.2 Computation flow	25
3.2.3 Feasibility	26
3.2.4 Feature selection	28
3.3 Model selection	28
3.3.1 Supervised learning method selection	29
3.3.2 Quantile predictions	30
3.3.3 Hyper-parameters tuning	30
3.4 Results	30
3.4.1 Graphical results	31
3.4.2 Discussion	32
3.5 Limitations and possible improvements	34
A Figures	38

Introduction

Nowadays, renewable energies, such as wind and solar power, are playing an increasingly important role in the global production of electrical energy.

Unfortunately, these new energy sources are very dependent on the weather and time conditions. Therefore, it is sometimes hard to forecast their production and, consequently, to adapt adequately the non-renewable sources production, such as nuclear centrals.

Hence, the question that we needed to answer in this project

“How much renewable energy will be produced tomorrow in Liège ?”

Global approach

A few tips were initially given, such as

- Choosing a renewable energy source (photovoltaic panels, wind turbines, etc.);
- Estimating the number of units of that source;
- Forecasting tomorrow's weather and predicting tomorrow's energy production;
- ...

Starting from there, we decided not to restrict ourselves as far as energy sources are concerned, and therefore chose to focus on both photovoltaic and wind productions.

For the estimation of the number of production units, we first had in mind to retrieve it *online* if that was possible, which appeared to be the case for the wind turbines. However, no precise records of photovoltaic installations were available.

From there, we divided our approach in three main branches that we would merge later.

1. Assessing the number and location of photovoltaic units;
2. Estimating the photovoltaic energy production;
3. Estimating the wind energy production.

However, due to data availability reasons (which will be explained later on), we decided to apply both photovoltaic branches to the province of Liège and the wind branch to the Wallonia Region.

We also chose to focus only on energy production forecasts. Therefore, all weather measurements and forecasts come from online (preferably free) API systems.

This report will be structured around the three main branches evoked above. Section 1 will cover the approach we used as well as the obtained results for the photovoltaic panels mapping. Section 2 will present in details the models used for estimating the photovoltaic power in the province of Liège, along with summary results. Section 3 will cover the methods and models built to forecast the wind power production in Wallonia, with corresponding results. Finally, we will conclude with a general overview of the project and a critical discussion about the limitations of our models.

Part I

Solar power

1 Photovoltaic panels mapping

As mentioned in the introduction, we couldn't find any public records of photovoltaic installations in Liège (or even in Belgium). In fact, the only data we found was proposed by the CWaPE and consisted in the estimated *power* of photovoltaic installations of less than 10 kW per municipality. Furthermore, these records were 3 years old (January 2017).

Therefore we decided to estimate the number and position of photovoltaic panels in the considered area using *remote sensing*, in particular using *aerial* or *satellite* imagery. The strength of this approach is that it is not restricted to any area or scale, supposing we have the images.

1.1 Wallonia aerial images

We were happy to find out that the Wallonia Region possesses a geographic information website, WalOnMap, from where it is possible to retrieve high resolution georeferenced aerial images. However, we were less happy about their quality, as they were compressed and therefore lost a lot of details. Unfortunately, we couldn't find any other free of access source.

WalOnMap images are organized under the *Web Map Tiles Service* standard (Masó, Pomakis, and Julia 2010), which implies it has an API that we can leverage to request the 512×512 tiles.

Based on the `owslib` Python library, we built a little package (`adopptrs/python/wmts.py`) so that we could interact with the API easily.

1.2 Detection model

In order to choose a detection model, we looked after well referenced papers linked to the subject, *i.e.* detection of objects in aerial images. Most of them were based on *neural networks* and more specifically on *segmentation neural networks*.

1.2.1 DeepSolar

One of the most reviewed was the “DeepSolar” (Yu et al. 2018) project conducted at Stanford University. DeepSolar’s goal is very similar to ours, *i.e.* detecting all PV panels in the United States using aerial imagery. However, the model used by DeepSolar is fairly complicated. It incorporates both image classification with the “Google Inception V3” network (Szegedy et al. 2016) and semantic segmentation in a single convolutional neural network.

Unfortunately, we believed that reproducing such network was not within our capabilities.

Furthermore, the reason why DeepSolar was first built as a classification network was because the researcher(s) didn't have segmentation masks as training set. In fact they only had images with labels indicating the presence or absence of panels. Therefore they had to solve a problem of semi-supervised segmentation.

Conversely, we had segmentation masks (cf. section 1.3) and therefore we could train directly a segmentation model.

1.2.2 U-Net

Looking for a segmentation model, we rapidly found one of the most famous one : U-Net (Ronneberger, Fischer, and Brox 2015). Initially designed for biomedical image segmentation, it actually works very well for a lot of applications. We also found a paper (Bischke et al. 2019) studying the addition a second head to SegNet (Badrinarayanan, Kendall, and Cipolla 2017) to improve the detection of edges. We decided to apply and implement this technique to U-Net using PyTorch Python library.

Note. This part of the project was conducted in parallel with our project of *Deep Learning*. Therefore, we won't explain the implementation and training process (algorithm, loss function, etc.) in this report. If you wish to know more, a GitHub repository ([francois-rozet/adopptrs](https://github.com/francois-rozet/adopptrs)) was created especially for this part and will contain all the source codes as well as the report for the *Deep Learning* project.

1.3 Trainings set

During our research of a model we found the open access “Distributed Solar PV Array Location and Extent Data Set for Remote Sensing Object Identification” (Bradbury et al. 2016).

This dataset contains the geospatial coordinates and border vertices of over 19 000 solar panels across 601 high resolution (5000×5000) images from four cities in California.

The geospatial coordinates are very precise and provided as polygon *vertices*, which allowed us to transform them into accurate black and white *segmentation masks*. Such masks were very handy for the model training. We also have sliced the original images into smaller 256×256 images as most networks cannot handle 5000×5000 images.

A small sample of the images and their respective masks can be found in the appendix (cf. Figure 22) or in the `adopptrs/notebooks/california.ipynb` Jupyter notebook.

1.3.1 Data augmentation

Even though the quantity and quality of the obtained images are very good, they are also very different from the images of WalOnMap. Indeed, the climate and landscapes of California are quite dissimilar from those of Wallonia, so is the *colorimetry* of the images.

In order to overcome this problem, we wished to make our model more robust and therefore chose to *augment* our dataset. Our approach was to randomly apply some transformation(s) to the Californian images *while training*. Therefore at each epoch, the training set is slightly different and becomes much harder to overfit on.

Some of the transformations we implemented (mostly using the PIL Python library) are : rotations (90° , 180° , 270°), horizontal and vertical flips, brightness alteration, saturation alteration, contrast alteration, blurring, smoothing, sharpening, etc.

A small sample of Californian images and one random augmentation of them can be found in the appendix (cf. Figure 23) or in the `adopptrs/notebooks/california.ipynb` Jupyter notebook.

1.4 Results

1.4.1 California

After having trained our model for about 80 epochs on the training set (14 236 images), we chose the network iteration (actually its weights) with the lowest average loss on the validation set (2078 images) and evaluated it on the test set (3105 images).

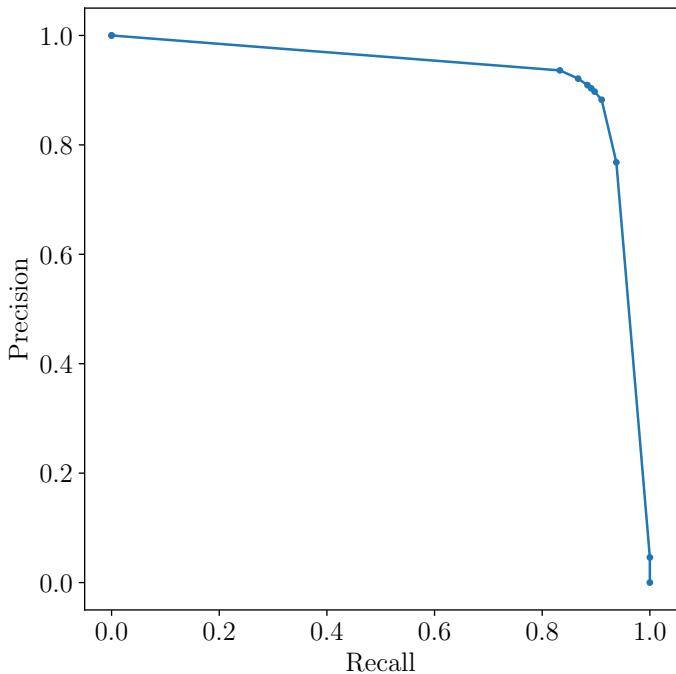


Figure 1 – Precision-recall curve of our model on the Californian test set.

As one can see in Figure 1, our model performed quite well on the test set, reaching an average precision of 92.51 %. With a threshold of 0.5, the model detects 89.1 % of the photovoltaic panels (recall) and what it detects has 90.3 % of chance to be a photovoltaic panel (precision).

As comparison, to quote DeepSolar's author,

“DeepSolar achieves a precision of 93.1 % with a recall of 88.5 % in residential areas and a precision of 93.7 % with a recall of 90.5 % in non-residential areas.”

We could say that our network performances are quite close to those of DeepSolar, but DeepSolar's test set is quite different from ours, therefore the comparison doesn't make much sense. However, the publishers of the dataset we use also have written a paper (Malof et al. 2016) about using convolutional neural networks for photovoltaic array detection, where they use the dataset they have published. Quoting them,

“The CNN is capable of excellent performance, detecting nearly 80 % of true panels with a precision measure of 72 %.”

Our model significantly outperforms these results.

1.4.2 Fine tuning

However, when we tried our model on WalOnMap images, it wasn’t performing nearly as well, as a lot of shadows¹ and dark roofs were detected as panels.

Therefore, we *hand-annotated*, using the “VGG Image Annotator” (Dutta and Zisserman 2019), 661 Wallonian images (512×512) in order to *fine tune* our model for a few (20) more epochs. This improved dramatically the performances of the network (cf. Figure 2) on WalOnMap images.

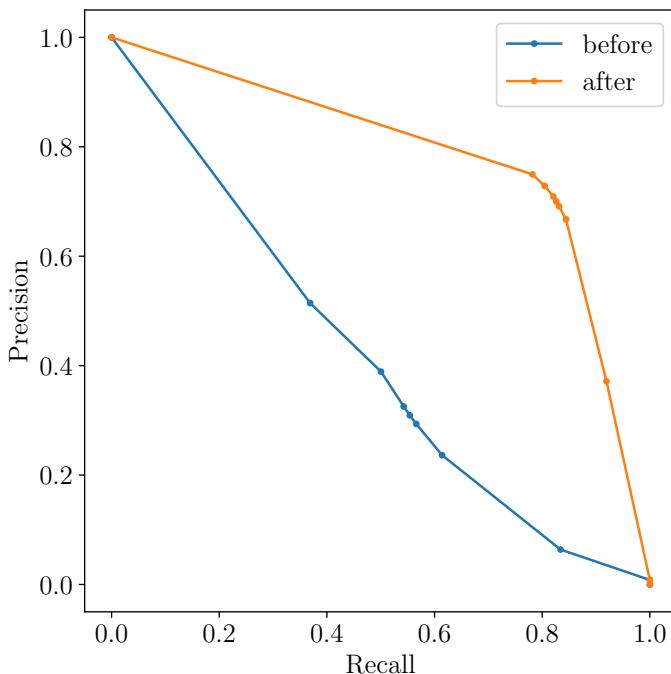


Figure 2 – Precision-recall curves of our model on the WalOnMap test set, before and after fine-tuning.

The final average precision reached by our network is 78.3 %, with 71.3 % precision and 82.6 % recall at a threshold of 0.5. Considering we hand-annotated “only” 661 images, this result is respectable, yet it could certainly profit significantly from a few more images.

Note. The WalOnMap test set is composed of 111 of the 661 images we annotated. They were never used for training.

¹There are very few building shadows in the Californian images while there are plenty in the Wallonian ones. We believe this is either due to the time of the year where the images were taken or the fact that California is much nearer to the equator than Belgium.

1.4.3 Province of Liège

Finally, we applied our fine-tuned model on the 846 703 tiles of the province of Liège. After about one day and an half of computations², we obtained the location and shape of 65 463 photovoltaic installations covering 2 554 505 m² of surface.



Figure 3 – Sample of detected photovoltaic panels in the province of Liège.

As expected from the previous section results, most panels are accurately detected and segmented, but there are still numerous errors, especially with shadows and canopies (*verrière* in french).

Note. As such, the locations and shapes are not usable for the power production model (cf. section 2). Therefore, we summarized each detected installation into 3 quantities :

²The computations have been performed using one GTX 1080 Ti GPU of the Alan cluster.

its geospatial location (latitude and longitude), its surface and an estimate of the surface azimuth angle (cf. section 2.3.1).

1.5 Limitations and possible improvements

Although it is still acceptable, the main limitation of our final model is its (average) precision on the WalOnMap images. We don't believe it to be due to the network itself, but rather to the low number of images we hand-annotated. Furthermore, none of the images we annotated were located *in-town*, therefore it is not surprising that most mistakes are made in such areas³. The most obvious way to improve our results is to annotate much more (varied) images, which we unfortunately had not the time to do.

Another limitation of our model, is that there is no measure of uncertainty about the area of the panels. We tried to use different thresholds in order to compute a kind of variance of the area of each detected installation, but we didn't succeed as, when the threshold changes, some installations merges or aren't detected anymore. This is definitely a sector of possible improvement.

³If you wish to see a small (2000) sample of detected panels along with aerial images, you can visit this website [link]. However, one should keep in mind that Google's aerial images are much better than those we had access to. Also, the WalOnMap's geospatial localisation is a little translated from the one of Google.

2 Photovoltaic power production

2.1 Overview of solar power forecasting

Classification of solar forecasting problems There are several time-horizons that can be considered for solar power forecasting tasks:

- Very-short-term tasks: forecasts for a few hours ahead;
- Short-term tasks: mostly day-ahead forecasts;
- Long-term tasks: forecasts for several days.

As was explained in the introduction, the goal of this project is to forecast the renewable production for the next day. Hence, we are interested in short-term tasks.

State of the art approaches for day-ahead solar forecasting From existing comparisons of photovoltaic forecasting methods (Nespoli et al. 2019) and other case studies (Bocquet et al. 2016), we observed three main standards in solar forecasting:

1 Statistical methods These methods are based on sequences of observations measuring different features at successive time instants. They generally only rely on collected data and are not interested in specific features such as panel locations, etc.

Among the mostly used methods, we have regression methods and artificial neural networks. The latter appear to be the most effective ones, allowing to capture and model sharp changes in observed data. Nevertheless, they rely on huge amounts of training data and a well-adapted model architecture, which might be hard to find.

2 Physical methods These methods are mainly based on *Numerical Weather Prediction* (NWP) and rely on panel characteristics in order to make an appropriate prediction. They show some inherent drawbacks: their resolution is quite small (up to maximum 50 km) and they achieve better performance when weather conditions are stable, which is certainly not ensured with irradiance data. Hence, they are less robust to sharp changes in observed data.

3 Hybrid methods These methods consist in a combination of the two previous ones, and should solve the issues of one another. Nevertheless, computational complexity increases and the hybrid performance depends on the individual performances of both models, which should be designed for a single plant/location.

Selected approach Since we only have access to irradiance measures/forecasts, building a highly complex statistical model around this data seemed to be a pointless task. Furthermore, we were not really interested in reproducing an existing artificial neural network architecture, nor in building one ourselves. Therefore, we moved away from statistical models to opt for physical methods.

As far as physical methods are concerned, we were interested in *physical intuition*; thus, we decided to derive very simple physical models in order to compare them with reference data, and see if such simplified models could actually reach a reasonable level of performance.

Note that we also swiped away hybrid methods since they involve statistical models.

As suggested by the guidelines, we first focused on a smaller scale than Liège, and started looking for a way to model the photovoltaic production at ULiège. The idea was to build a *panelwise* physical model that we would then scale up or simply adapt for the case of Liège, using panels data obtained by processing satellite images in our panel enumerator. By panelwise, we mean that we compute the output power for each panel, or at least for each panel installation. This computation would thus be dependent on the coordinates (latitude and longitude) of each installation, the tilt angle, the area of the installation, ... The results would therefore be inherently dependent on the way and accuracy with which we assess the number of photovoltaic units. Note that, since the ULiège model is only a *sketch model* that we built to check if equation 1 gave plausible results, without processing any satellite image, we thought computing its error metrics was not relevant. Hence, we will only display some graphical results where we find it appropriate.

In a second time, we decided to build a second model, which we named *provincial* model, that computes the photovoltaic production for the whole province of Liège. As will be explained in subsection 2.2, moving from the city scale to the province scale was done for the sole reason to have reference production data to fit to.

These two models will be explained in further details in later sections. First, we will introduce the data we managed to collect for this sub-problem.

2.2 Data collection and processing

2.2.1 Photovoltaic production data

The first step was to gather photovoltaic production data. For our panel-wise model, we needed to get production data from the photovoltaic panels installed at the parking lots of the Sart-Tilman. We obtained access to a live database that provided production data for each minute, which gave us a high-confidence reference for our example model at ULiège.

For production data in Liège, the only publicly available source appeared to be Elia, but it only provided data at a provincial scale. This is the main reason why we considered both photovoltaic models over the whole province of Liège instead of the city of Liège; otherwise we would have had no reference data to compare to. Note that the panelwise model could be applied at any scale, provided that we process the corresponding satellite images. We simply won't have any reference data to compare to.

Elia provides data for each 15-minutes step and gives access to measurements and forecasts, the latter being helpful to compare our model predictions to a reference forecast.

Requesting data to Elia might lead to empty responses or simply missing data for some timestamps. For that reason, we had to treat data carefully to avoid runtime issues in our models. Concerning the photovoltaic production data, we decided to perform a simple interpolation filling for the potential missing data, which seemed quite reasonable.

2.2.2 Irradiance data

Finding free of access irradiance data seemed like an impossible task at first. While we collected past irradiance data at the Sart-Tilman (which allowed us to complete and test our first example of panelwise model), the discovery of a data source that could

provide both irradiance measures and irradiance forecasts at any location was harder than expected. Considering two independent data sources seemed to be a bad idea due to the likely bias that would exist (adding on top of that the inherent bias present in the forecasts).

Due to the bound on API calls that we would get with any existing free irradiance API, we decided to restrict our data measures to the city of Liège. Obviously, this restriction will impact our production estimates since we aim at predicting at a provincial scale. We decided to opt for Solcast, which offers some credits for academic research purpose and which provides 1000 API calls. As it only allows to request irradiance data for a single production site (for our account type), it perfectly fitted our restriction.

As far as irradiance measures and forecasts are concerned, no complex pre-processing step was needed.

The first step was to decide for the date and time associated to each measure. Indeed, the API provides measures at a 30-minutes granularity, and each measure consists in an average measure over the past 30 minutes. For instance, the global horizontal irradiance (GHI) reported at 9:00 corresponds to the average GHI measured between 8:30 and 9:00. Hence, we arbitrarily decided to associate to each such measure the date and time corresponding to the middle of the considered averaging period (in the case of our example, it would correspond to 8:45). It also appeared to yield more precise results when doing so rather than when considering the date and time of the end of the period.

Furthermore, the second step in irradiance pre-processing consisted in building the appropriate data frames in order to store them for caching purposes. Finally, we had to detect potential missing data and perform some action regarding this missing data. To solve this missing data issue, we decided to perform a linear interpolation, as is done for Elia's missing measures.

2.3 Models description and practical details

This section will explain in further details the two models described earlier.

2.3.1 Panelwise model – `python/solar_panelwise.py`

As was already evoked earlier, this model consists in computing the output power for each panel installation. What we plan to do is to use in this model the data (coordinates, area, azimuth) produced by our photovoltaic panels mapping.

After certain research, we derived a formula that estimates the production of a photovoltaic panel depending on several particular angles:

- The **incidence angle** θ between the sun rays and the normal vector of the panel installation. This angle itself depends on other angles that will be described below.
- The **solar altitude**, alt , in order to rescale irradiance measures since what is measured is the horizontal irradiance, not the irradiance that is perpendicular to the panel. It simply corresponds to the angle between the horizontal plane a line drawn from the sun towards the panel.

All the following angles intervene in the cosine of the incidence angle θ :

1. The **surface azimuth**: it corresponds to the angle measured between a vertical line going from North to South and the projection on a horizontal plane of the normal to the panel (positive when measured clockwise);
2. The **declination angle** δ , which only depends on the day of the year;
3. The **latitude** and **longitude** of the panel;
4. The **hour angle** ω , which depends on the location of the panel;
5. The **tilt angle** of the panel, β .

Hence, whenever the cosine of θ or the sine of alt is negative, it means that the sun is behind the panel or that the sun has set, respectively.

The derived model is the following:

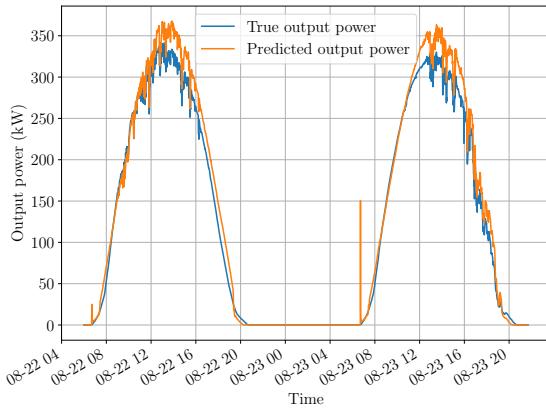
$$P = \frac{\eta I \cos(\theta) A}{\sin(alt)} \quad [\text{W}] \quad (1)$$

where η is the efficiency of the panel installation, A is the area of the panel and I is the irradiance. For the sake of readability, we decided not to write down the formula of $\cos \theta$. However, it can still be found on this website or in the corresponding source code, `python/solar_panelwise.py`.

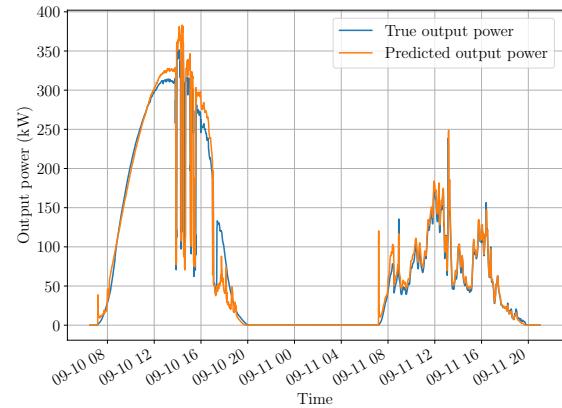
Note that, for the case of ULiège, since we had access to technical specifications of the panel installations, we knew the exact efficiency, tilt, etc. and we also had access to other information such as:

- The maximum power bound of the panels
- The temperature influence

Nevertheless, even if they added a bit of precision, most of these parameters did not play a key role in estimating the output production. Only for the sake of visualization, samples results of the model implemented in equation 1 can be observed on Figure 4a and 4b. The dates have been chosen according to the available data.



(a) August 22nd and 23rd 2019.



(b) September 10th and 11th 2019.

Figure 4 – Samples of the physical model derived in equation 1, for ULiège.

Note that the presence of peaks is due to the fact that both the cosine of the incidence angle and the sine of the altitude are very small (but positive) at these time steps. This seems to happen because of the small time granularity we used for ULiège (1 minute).

From this model, we can thus get an estimate of the output power of photovoltaic panels at any scale, provided that we have access to specific information about them, namely the latitude, longitude, area and surface azimuth.

Taking uncertainties into account for parameters such as η and β was only slightly performed for this model, for several reasons. Firstly, modeling uncertainties for β seemed to be a hard task since it is entangled in $\cos \theta$ and thus we couldn't directly interact with it through a PyStan⁴ model.

Hence, the only remaining parameter we could toy with was η , which was not completely sufficient to model correct and reasonable uncertainties. However, we still managed to model some uncertainty about η , making the panelwise model less rigid, yet not perfect.

Note. All the following equations are expressed in MW.

Modeling uncertainty about η was done using as prior a normal distribution centered around the predefined value for η (which we set equal to the efficiency of the panels at ULiège, i.e. 0.162), with a scale of 0.1:

$$\eta \simeq \mathcal{N}(0.162, 10^{-2})$$

To fit the posterior distribution of this parameter, we used Elia's production measurements over the last 7 days and modeled these measurements as a normal distribution centered around η times *naive_power* :

$$true_power \simeq \mathcal{N}(\eta \times naive_power, 100^2)$$

where

$$naive_power = \frac{I \cos(\theta) A}{\sin(alt)}$$

The large scale seemed to increase convergence speed.

Finally, we obtain the predicted production power using the posterior distribution on η as well as by modeling a new quantity, *forecast_power*, which corresponds in some sense to the true forecasted power. This true forecasted power is obviously unknown, and we model it as the mean of a normal distribution from which the *naive_power* computed using irradiance forecasts is sampled:

$$naive_forecast_power \simeq \mathcal{N}(forecast_power, 10^2)$$

Then, we compute the predicted production by modeling it as a normal distribution centered around η times *forecast_power*, where both factors are sampled from their posterior distributions :

$$predicted_power \simeq \mathcal{N}(\eta \times forecast_power, 10^2)$$

⁴PyStan is a Python probabilistic framework, which is particularly useful to model uncertainties about parameters in physical models, when it comes to making inference using these parameters.

Nevertheless, one should keep in mind that the main contribution to this model will correspond, in the end, to the total estimated area of photovoltaic units, which will be mainly responsible for the computed output power. The uncertainty should thus rather come from this area than from panels hyperparameters.

Note that we also tried to model these uncertainties in a different way, considering each panel installation's output power (computed in 1) as being weighted by some unknown, which would represent our uncertainty, and fitting these weights with PyStan with respect to Elia's production measurements. Unfortunately, the corresponding model became extremely slow and we decided to drop that idea.

2.3.2 Provincial model – `python/solar_provincial.py`

In this second model, we wish to estimate the photovoltaic production at the provincial scale, without having detailed information about the locations of the panels, their areas, etc.

However, we found a data file containing detailed information, among others about the installed power (in kVA, which can be assimilated to kW, for simplifications) of photovoltaic panels for each municipality in the province of Liège. From there, we could get a first estimate of the installed power for the whole province.

To get an estimate of the power production, we need to compute a quantity expressed in Watts. Since we have the installed power (kWPeak) and the irradiance (W/m^2), we need another quantity that would, in some sense, *cancel* the installed power to get something in m^2 in the end. For that purpose, we defined a parameter *peak_area*, corresponding to the average surface of panels per kWPeak (m^2/kWPeak). Hence, multiplying the installed power by this parameter gives us the quantity we are looking for: the area of photovoltaic panels installed in the province of Liège.

From there, the provincial model is quite straightforward:

$$P = \eta IA \quad [\text{W}] \tag{2}$$

where η is the panel efficiency, I is the irradiance and $A = \text{peak_area} \times \text{kVA}$ is the area of panels in the province (with kVA being the installed power).

Exactly like for the previous model, all parameters are arbitrarily fixed and the model is thus very rigid and will very likely perform well only by chance. To solve this issue, we decided to implement a PyStan program that would be in charge of modeling appropriate uncertainties for all factors of the equation.

In this program, we model the uncertainties with very simple equations and distributions, in the following manner⁵. We will compute the posterior of 4 quantities, using Elia's measurements (modeled as a normal distribution centered around the product of these 4 quantities).

Note. The order of magnitude of the units used in this model are kW, not W nor MW.

1. The prior of the efficiency η is modeled using a normal distribution, centered around the value we arbitrarily fixed (ETA) and with a scale of 0.01.

$$\eta \simeq \mathcal{N}(\text{ETA}, 0.01)$$

⁵For a more code-like representation, please refer to `python/solar_provincial.py`

2. The prior of the peak area is modeled using a normal distribution too, centered around the predefined value (PEAK_AREA) and with a scale of $1 \text{ m}^2/\text{kWPeak}$.

$$\text{peak_area} \simeq \mathcal{N}(\text{PEAK_AREA}, 1)$$

3. The observed irradiance measures are also assumed to be uncertain. Hence, we modeled a new quantity (I) representing the true observed irradiance, as the mean of a normal distribution where the API measures (I_{obs}) are sampled from, with a scale of 1 W. The uncertainty will thus be quite small for the irradiance but this is for the sake of coherence with respect to the real world. Indeed, when setting a higher scale, the mean distribution of the irradiance was always strictly positive, which made no sense for night observations. Nevertheless, this behavior still slightly remains, yet it can be considered as negligible.

$$I_{\text{obs}} \simeq \mathcal{N}(I, 0.001^2)$$

4. The installed power's prior is modeled as a uniform distribution. Therefore, the uncertainty will be initially higher for that parameter, which is something we wish: as it represents the installed power for the whole province, we want to allow this parameter to vary as much as possible in order to fit well Elia's measurements.

As was said above, we will build their posterior distributions thanks to the following distribution, where *true_power* corresponds to Elia's production measurements (expressed in kW):

$$\text{true_power} \simeq \mathcal{N}(\eta \times I \times \text{peak_area} \times \text{kVA}, (10^5)^2)$$

Building the posterior predictive model is done in a similar way. Indeed, the forecast irradiance is modeled in the same manner as the observed irradiance, and we simply model the predicted output power as a normal distribution centered around the product of the 4 previous quantities, replacing I by I_{for} , the quantity representing the true irradiance forecasts. We thus have:

$$\text{predicted_power} \simeq \mathcal{N}(\eta \times I_{\text{for}} \times \text{peak_area} \times \text{kVA}, 100^2)$$

From this predicted power, we can retrieve the mean distribution and compare this mean distribution with Elia's measurements (if they are available, which won't be the case in a real-world setting) or with Elia's forecasts.

2.4 Analysis of the models accuracy

In order to assess the quality of both models, we applied the following procedure. For both models, we fit and construct the posterior models using Elia's measurements over the past 7 days, then we predict for the current day using the predictive model defined in the previous section. For instance, if we wish to predict the power production for April 15th, we build our posterior model using measurements from April 8th up to April 14th and predict using the forecasts requested for April 15th.

In order to get representative results, we firstly conducted this procedure for the whole year of 2019, i.e. we predicted for each day of 2019. However, since we had no access to past irradiance forecasts, we had to perform these tests using irradiance measures only.

Doing so will necessarily bias the results in an optimistic way since irradiance measures will always be more accurate than irradiance forecasts. This can be observed on Figure 5 and 6, where we compared the use of irradiance measures and irradiance forecasts for our predictions.

We chose to compute 2 specific metrics, namely the mean-square error (MSE) and the root mean-square error (RMSE), for each day. The RMSE, in some sense, will give us a measure of the error that we will make for the predicted day. Note that these metrics are computed after removing night observations: that is, we only consider points at time steps where Elia's production measurements are non zero. Then, we simply compute the average metrics along with their standard deviation.

As baseline models, we constructed 2 very simple models:

1. The first one consists in predicting as produced power for day D the power measured by Elia for day $D - 1$.
2. The second one consists in predicting as produced power for day D the average power measured by Elia over the past 7 days.

Along with those, we also assessed the forecasts made by Elia as well as our naive predictive model (which uses fixed parameters, hence being more rigid). Note that, for our posterior model, we compute the metrics with respect to the mean distribution of the posterior model.

A general remark to note for our two posterior models is that the quality of their predictions will mainly depend on the training period, i.e. on the past 7 days of measurements. Hence, if the training period is not very representative of the power that will be produced on the predicting day, it is very likely that our parameters will wrongly fit the training period and thus yield poor results for the predicting day. This could potentially be solved by training on a much larger period, although we did not really try doing so. We tested increasing arbitrarily to 3 months of training data but it yielded very poor performance as far as time complexity is concerned. Furthermore, fitting on the past 7 days seemed intuitively to be a reasonable assumption to represent how the irradiance would evolve through time.

These results can be found in Table 1. From these, we can see that some are very accurate and perform better or equivalently to the baselines, while some others perform very poorly. For most cases, Elia's production forecasts are very close to the measured production curve. This could be explained by the fact that Elia probably does not rely on such a simple model to compute its forecasts. The simplicity of our models is clearly a limitation, even though we are still satisfied with the results produced, on average. Furthermore, the irradiance curve seems to mainly control our predictions, as will be discussed in Section 2.5.

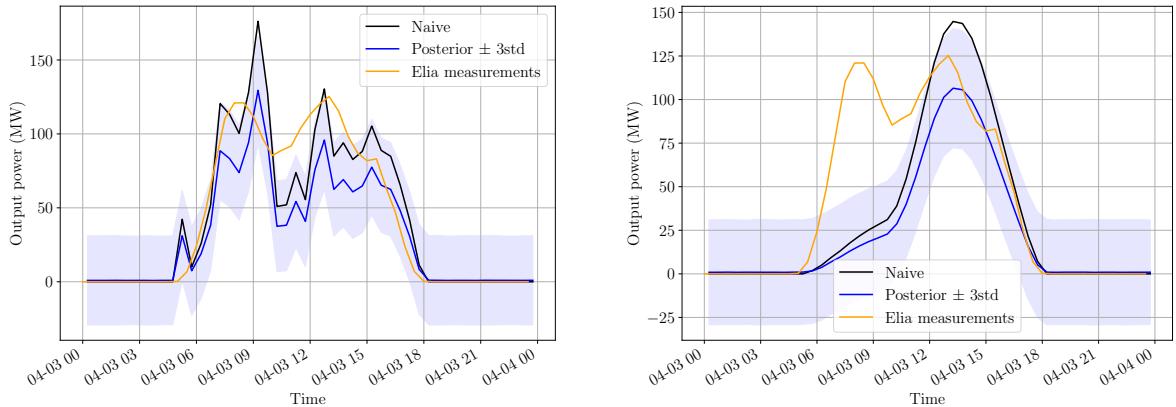
We can observe from Table 1 that both posterior predictive models perform similarly. However, the prior models show completely different performances. Indeed, the prior panelwise model has a much higher MSE on average. This could be due to the accuracy of our photovoltaic panels mapping, which seems to slightly overestimate the total area of detected panels (as can be seen on Figure 8 and 9). Hence, since we use a fixed efficiency η , this efficiency seems to be too high most of the time, and too low for some other cases.

	MSE	RMSE
Prior panelwise model	5316.57	62.85 ± 37.66
Posterior panelwise model	1563.30	32.23 ± 23.33
Prior provincial model	2210.83	42.44 ± 20.62
Posterior provincial model	1595.59	32.63 ± 23.48
Baseline 1	2616.99	36.34 ± 36.68
Baseline 2	2339.52	40.87 ± 26.34
Elia's forecast	324.85	15.16 ± 10.11

Table 2 – Mean MSE and mean RMSE along with standard deviation, for April 2020 (using irradiance forecasts), in MW.

	MSE	RMSE
Prior panelwise model	2852.73	49.25 ± 20.68
Posterior panelwise model	510.84	20.06 ± 10.42
Prior provincial model	608.98	21.94 ± 11.31
Posterior provincial model	431.23	18.49 ± 9.46
Baseline 1	2195.60	38.87 ± 26.19
Baseline 2	1816.74	37.34 ± 20.56
Elia's forecast	381.85	16.79 ± 10.01

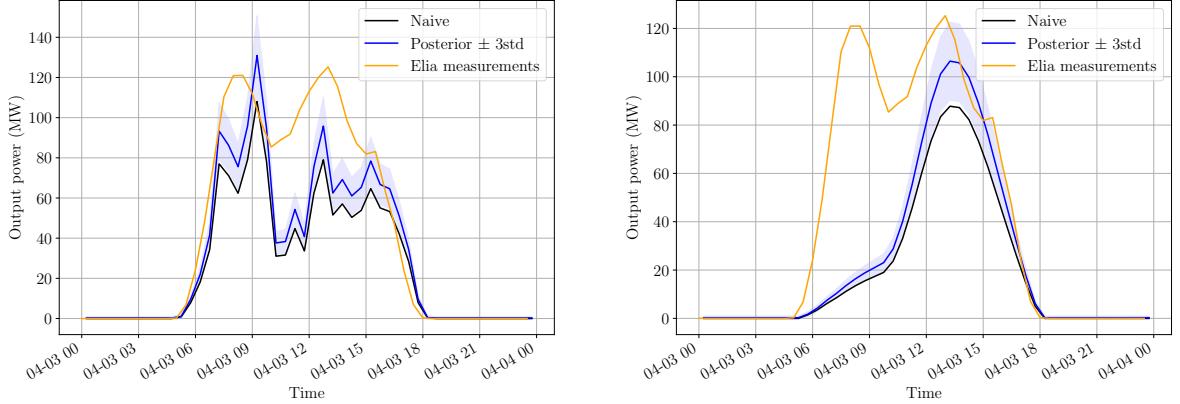
Table 1 – Mean MSE and mean RMSE, along with standard deviation, for 2019 (using irradiance measurements), in MW.



(a) Panelwise model predicting with irradiance measurements.

(b) Panelwise model predicting with irradiance forecasts.

Figure 5 – Comparison of panelwise model with irradiance measurements/forecasts (April 3rd 2020).



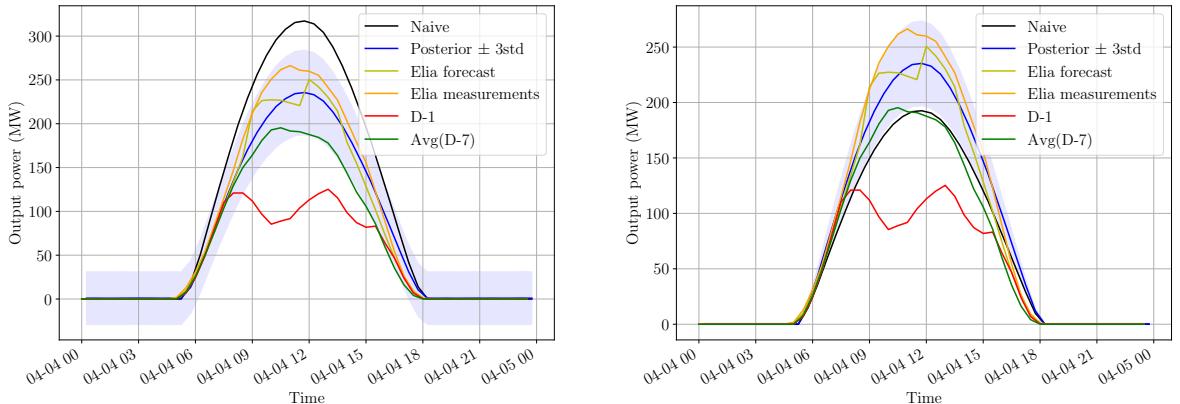
(a) Provincial model predicting with irradiance measurements.

(b) Provincial model predicting with irradiance forecasts.

Figure 6 – Comparison of provincial model with irradiance measurements/forecasts (April 3rd 2020).

To get representative results using irradiance forecasts instead of irradiances measurements, we conducted the same testing procedure for the whole month of April. Consequently, the results (in Table 2) appeared to be worse than those obtained for 2019, as could be expected. Some sample executions can be observed on Figure 7, 8, 9 and 10. Once again, we can see that Elia outperforms all models, and that both posterior models perform similarly. However, as could be noticed in Table 1, the prior panelwise model performs much worse than the prior provincial model.

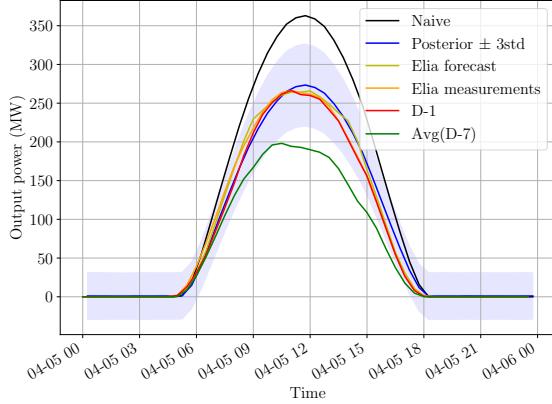
Nevertheless, we can see that in both cases (using irradiance measurements and irradiance forecasts), both posterior models seem to perform better than our baseline models. The performance of the latter will totally depend on the measured production in the past. Hence, if that past production is not representative at all of the future production, results will be bad. Overall, the *averaging* baseline model should be a little bit more robust than the one that simply reproduces the production of the previous day, unless production measurements are very similar for several consecutive days.



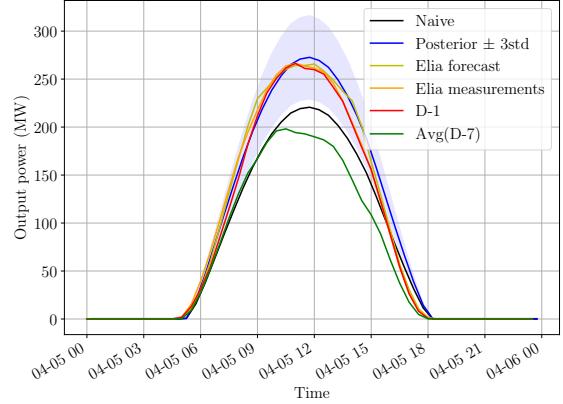
(a) Panelwise model (using irradiance forecasts).

(b) Provincial model (using irradiance forecasts).

Figure 7 – Samples of accurate results with both models (April 4th 2020) and the baselines.

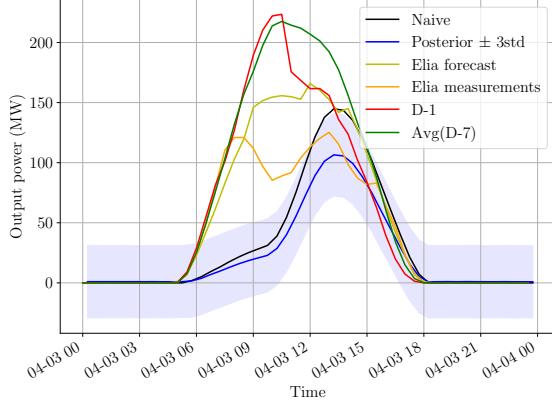


(a) Panelwise model (using irradiance forecasts).

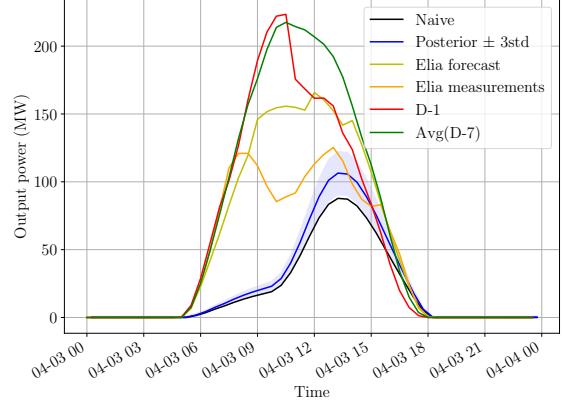


(b) Provincial model (using irradiance forecasts).

Figure 8 – Samples of accurate results with both models (April 5th 2020) and the baselines.

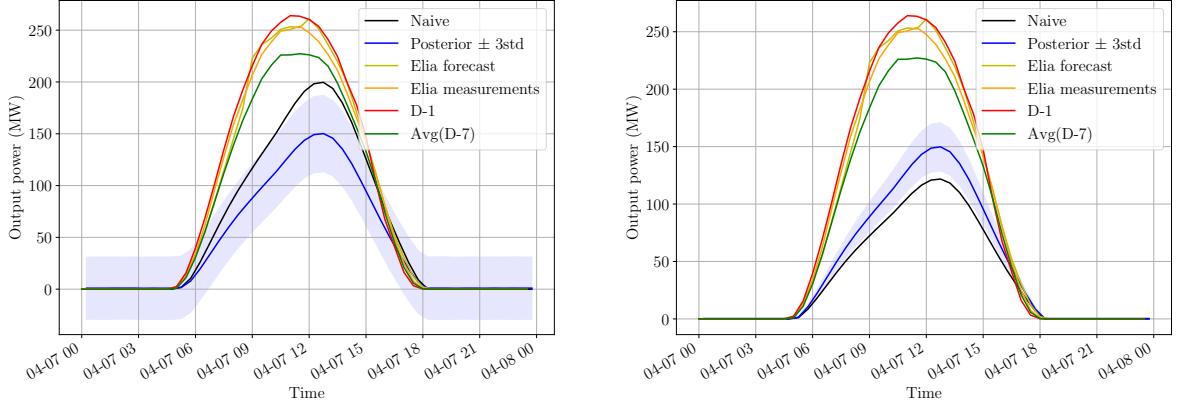


(a) Panelwise model (using irradiance forecasts).



(b) Provincial model (using irradiance forecasts).

Figure 9 – Samples of inaccurate results with both models (April 3rd 2020) and the baselines.



(a) Panelwise model (using irradiance forecasts).

(b) Provincial model (using irradiance forecasts).

Figure 10 – Samples of inaccurate results with both models (April 7th 2020) and the baselines.

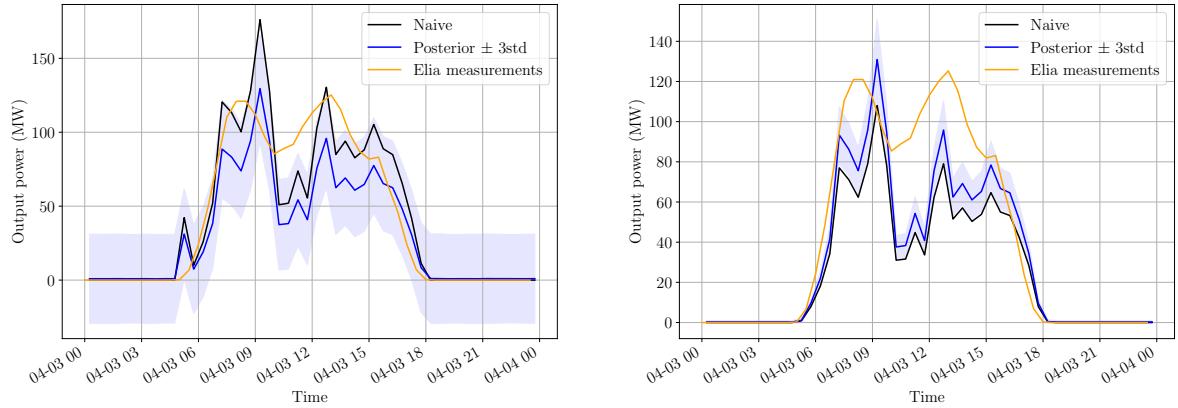
Since irradiance forecasts are linked to irradiance measures, in the sense that they are very likely computed from irradiance measures, an alternative to this procedure is to fit to Elia’s measures by using an irradiance quantity that would be conditioned on both the irradiance measures and the past irradiance forecasts, assuming that irradiance forecasts represent an unbiased estimate of the true irradiance measures. We conducted tests for the same period (April) but they did not show any improvement, which is why we only decided to mention this alternative without expanding too much.

2.5 Limitations and possible improvements

Even though we obtain satisfactory results for the tests conducted on the whole 2019 year (Table 1), we feel a little bit less satisfied about the results obtained for a *real-life* setting (Table 2), which is quite natural since one uses irradiance measurements while the other one deals with irradiance forecasts.

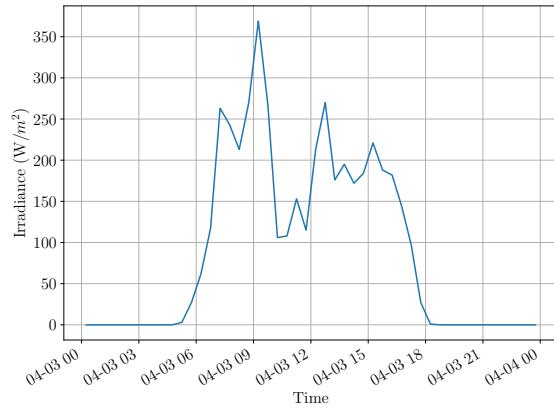
Nevertheless, this performance loss is something we were prepared for. Indeed, both models rely on simplifying assumptions.

Our provincial model does not use any information about all the panel installations in the province of Liège. Furthermore, it models the photovoltaic production in a very simple manner, with a normal distribution centered around the product of 4 quantities. Therefore, the only kind of improvements we can expect with respect to the naive model is a re-scale of the original curve. In addition, this production curve (for the naive forecasting model or for the posterior forecasting model) is mainly driven by the irradiance curve. Hence, if the irradiance forecast curve is not representative of the true irradiance that will be measured, it is very likely that the corresponding forecasting models will wrongly estimate the future production. This can be observed on Figure 11 and 12, where the forecast curve is clearly non-representative of the measured power, thus non-representative of the true irradiance measured. Note that this influence is as strong even when we use irradiance measurements.



(a) Panelwise model (using irradiance measurements).

(b) Provincial model (using irradiance measurements).



(c) Irradiance measured.

Figure 11 – Influence of the irradiance measures on the predictions.

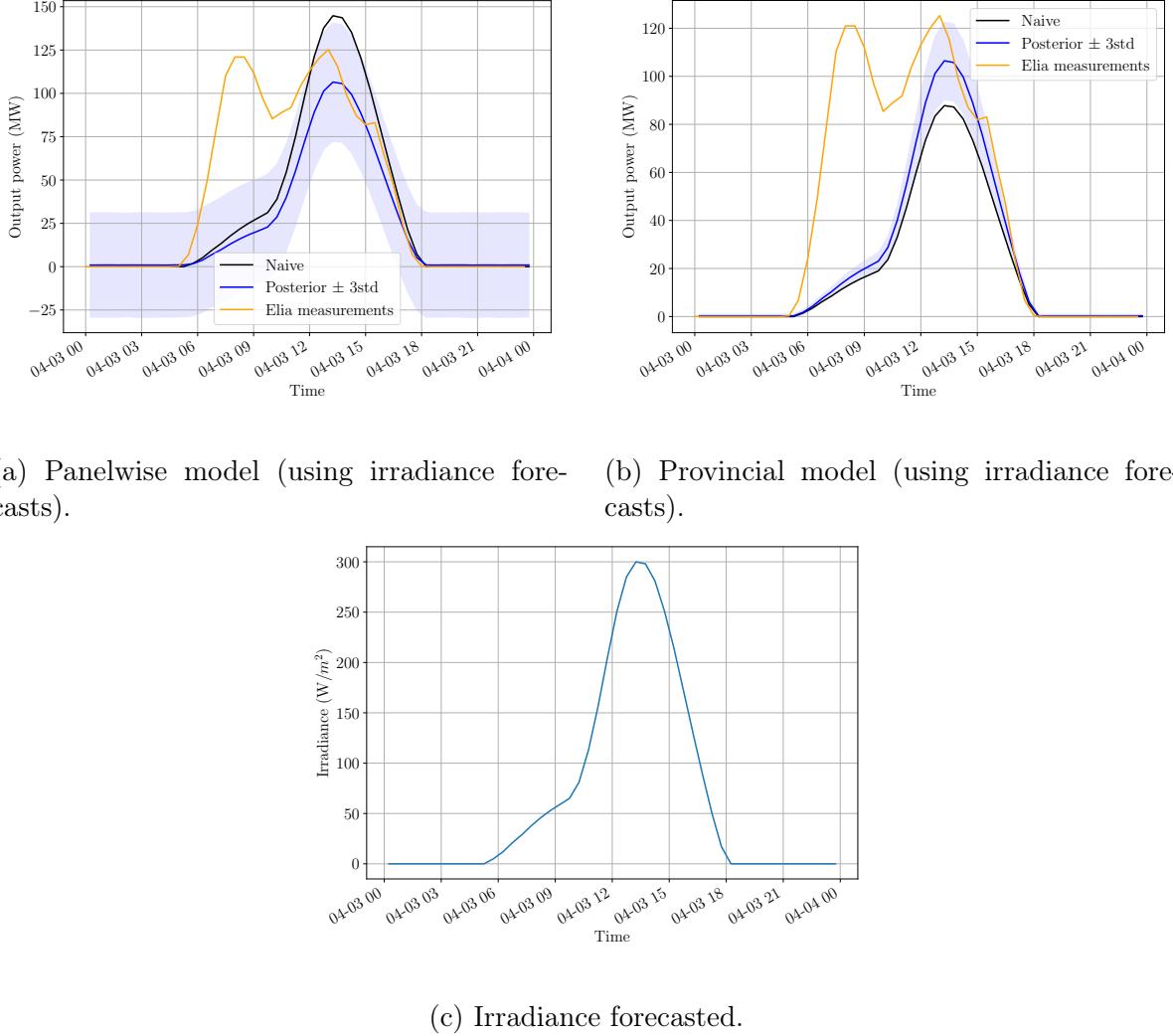


Figure 12 – Influence of the irradiance forecasts on the predictions.

Furthermore, we only retrieve irradiance data at a single location, which we chose to be the city of Liège⁶. This restriction thus naturally implies that the irradiance curve will, most of the time, not be of the same shape as the measured power.

As far as our panelwise model is concerned, it is somehow more sophisticated since it considers panel-specific characteristics: its location (latitude and longitude), its estimated area, azimuth, etc. Therefore, it should be more precise about the produced power. It is the case for the panelwise model applied to ULiège, as could already have been seen on Figure 4. This is quite logical since we had access to very accurate irradiance data and technical specifications. Note that no uncertainty was introduced in the ULiège model because we knew exactly the number of panels, their areas, azimuths, etc.

It is no longer the case when we extend this model to the whole province, as can be seen from Figure 9a and 10a. Indeed, the estimated azimuths and areas are, as their name indicates, *estimates*; hence less reliable. Yet, one could have expected consistently more accurate results than for the provincial model. This is not true, since we use the exact same irradiance data. Therefore, we also have the restriction of a single irradiance location.

⁶Coordinates: (50.6326°, 5.5797°)

Note that, even more than for the provincial model, it would have been unfeasible to request irradiance data at each panel location, as there are so many in the province of Liège.

A way of partly solving this issue could have been the following: assuming we could request irradiance data at multiple sites, we could try and *aggregate* (in some sense) the panels locations into a much reduced number of locations, according to their coordinates. Obviously, when computing the solar angles, we would still consider each panel's true coordinates. Then, we could compute each installation's output power according with their corresponding irradiance, chosen according to their closest aggregate location. Although we did not try this alternative (which we couldn't because of the API limitations), we feel that this would very likely help our model.

Part II

Wind power

3 Wind power production

3.1 Overview of wind power forecasting

Classification of wind forecasting problems In wind power forecasting, one can be interested in different problems:

- Very-short-term or immediate-short-term (a few hour ahead);
- Short-term or medium-term (few hours to several days);
- Long-term (multiple days ahead);

Each of these problems answers different needs (load balancing, power production planning, long-term power needs prognostication, etc), but above all, each of these problems makes use of different mathematical, statistical or algorithmic tools. Our project addresses the day-ahead task, falling into the short-term category.

Below, we will discuss the most appropriate approach to use among the classical approaches in wind power forecasting.

Classical approaches in wind power forecasting From different reviews of wind power forecasting (Sweeney et al. 2019; Messner et al. 2020), we have identified the following approaches in this field.

1 **Physical modeling:** based on Numerical Weather Prediction (NWP), the model aims at outputting the power produced by one power plant using the power plant characteristics (wind turbine power curves, surface roughness and obstacles, terrain wake, hub heights, even the wind farms wake effect on themselves is taken into account in recent works). These approaches are said to be very computationally expensive, and quite hard to apply on very large scale situation such as the Wallonia which is constituted of 67 power plants.

2 Statistical modeling

- a. On the very short term: *time series based forecasting*: the NWP data is not used, only the power time series is used to make the prediction. This approach is said to be hard to beat on 4 to 6 hours ahead predictions (very short term).
- b. On the longer term: *post processing of NWP*: the temporal aspect is not considered. The power predictions are based on NWP. Roughly, this approach aims at mimicking the physical model when some information are missing such as the surface roughness and terrain wake effect for each power plant.

3 Hybrid method: uses *post processing of NWP and of physical modeling*.

The most promising and adapted approach seems to be the *post processing of NWP*. Indeed, we address the day-ahead forecasting task, weather data and aggregated power

measurements are available, and power plant physical characteristics are not available to build a complete physical model.

Physical model However, a simple physical model has been built as a baseline. Unfortunately, it performs poorly, as can be seen on figure 14b. This model uses the power curves retrieved from the *wind turbine models* website (Matysik and Bauer 2020), or, when not available, builds naive theoretic power curve from the cut-in, cut-out and rated wind speed, using the well known cubic relation, as can be seen on figure 13a and 13b. This is done for the 35 models of wind turbines present in Wallonia. The automatic retrieval of wind turbines characteristics (power curve, cut-in, cut-out and rated wind speed) is done in `cache/wind_farms.py`.

Then, it computes the power produced at each wind turbine, using as input the wind speed measured at the power plant of this wind turbine. Finally, all results are aggregated. The implementation of this simple physical model can be found in `misc/wind_phys_model.py`. On the month of March 2020, the MAE (mean absolute error) was of 355.16 MW.

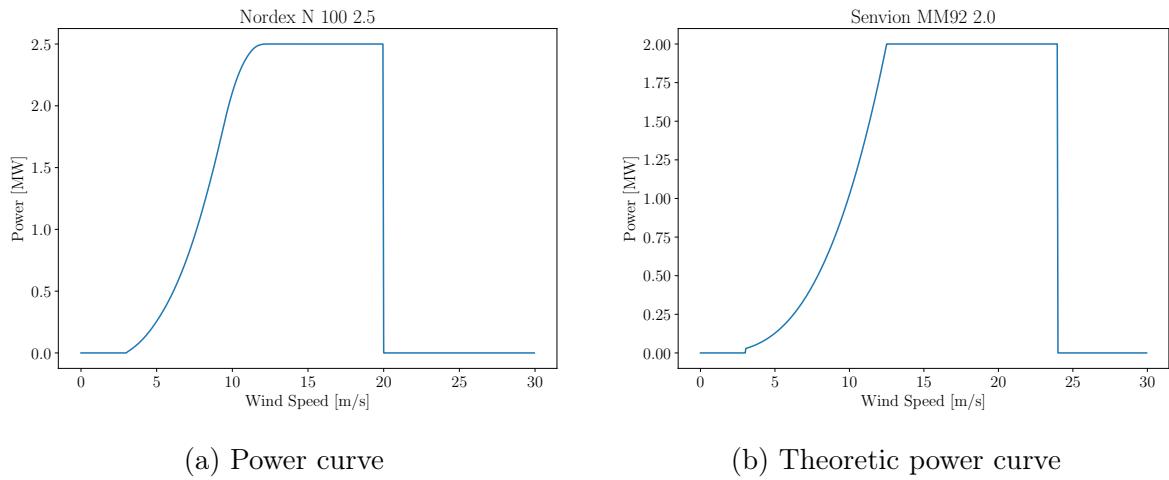


Figure 13 – Examples of power curves

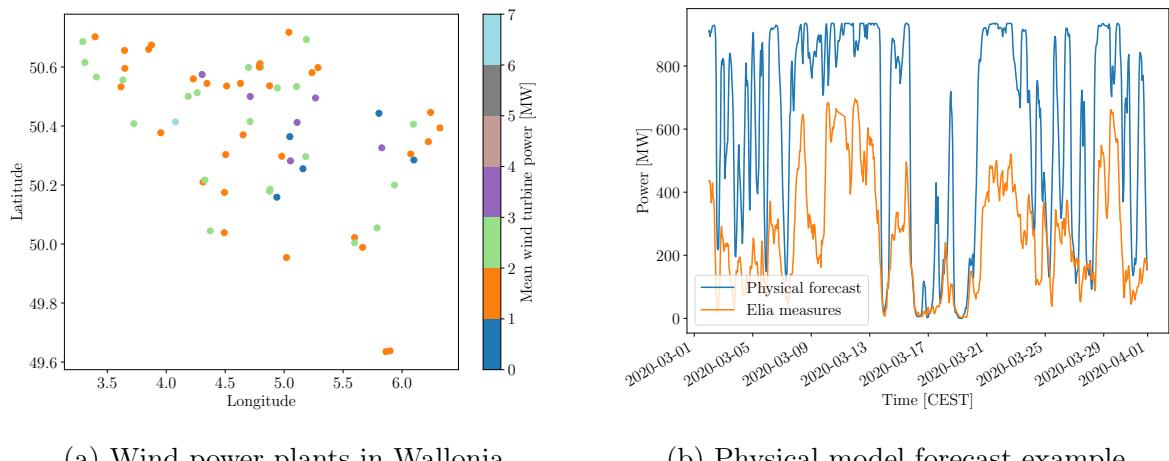


Figure 14 – Wind power plants location and physical model

3.2 Data and computation flow

3.2.1 Data

Several source of data were available and have been used in the wind forecast computation:

- **List of wind turbines** in Wallonia (SPW-Energie 2018): list of all wind turbines in 2018 and projects for 2019. This data can be visualized on figure 14a (aggregated by wind power plants). For each wind turbine, the following data is available:
 - Location (latitude and longitude);
 - Power plant unique name;
 - Wind turbine model reference (brand - type - nominal power).
- **Wind turbine models characteristics** (Matysik and Bauer 2020)
 - Cut-in, cut-out and rated wind speeds;
 - Power curve.
- **Wind power measures and forecast for Wallonia** (Elia 2020): measures and day-ahead forecast in MW every 15 min
- **Wind weather measures and forecast** (*Dark Sky API* 2020): one 7-measure every hour at any location requested.
 - Wind speed, wind gust, wind bearing, temperature, humidity, pressure, air density.

As can be seen here, no data is available to evaluate the wind power production forecast at the Liege province level. Thus, the whole wind forecasting task is performed *at the regional level*.

3.2.2 Computation flow

The figure 15 helps clarifying the way data has been used in the computation of the forecast.

As the NWP post-processing approach has been selected, the first step was to build a learning set on which to build models. As can be seen from figure 15, the learning set is built in the following way:

- Computation of the mean locations of the 67 wind power plants from the *SPW Energie* data (`cache/wind_farms.py`)
- Retrieval of 2.25 years of 7-dimensional weather measures at the 67 locations from *Darksky* (`cache/wind_weather.py`)
- Retrieval of 2.25 years of total wind power measures in Wallonia Region from *Elia* (`cache/wind_power.py`)
- Union of the two sets of data: weather (input) and power (output) into a learning set, with the measures of elia being averaged per hour, constituting a *weather measures* learning set (`cache/wind_ls.py`)

In parallel, from April 5th, a *weather forecast* test set is built. It is constituted of weather forecast as input instead of weather measurements. Indeed, doing so will allow us to quantify the loss in performance when using our model in real conditions (*i.e.* on forecast). As no data is available for past weather forecast, we had to request it each day (at 11 o'clock) for the next day and store it. This explains why this test set is unfortunately available from the 5th of April only.⁷

Then, the model is built using a part of the learning set as train set, and evaluated on weather measures using another part of the learning set set. Finally, its performance in real conditions is evaluated using the *weather forecast* test set.

As most of the steps of this computation flow are very slow and unreliable (rely on external API), we have decided to use a cache-based approach. It means that each step uses the data that has been cached by its parents in the graph, and never asks them to re-compute it in live. For most of the data, it is not necessary to refresh the cache very often.

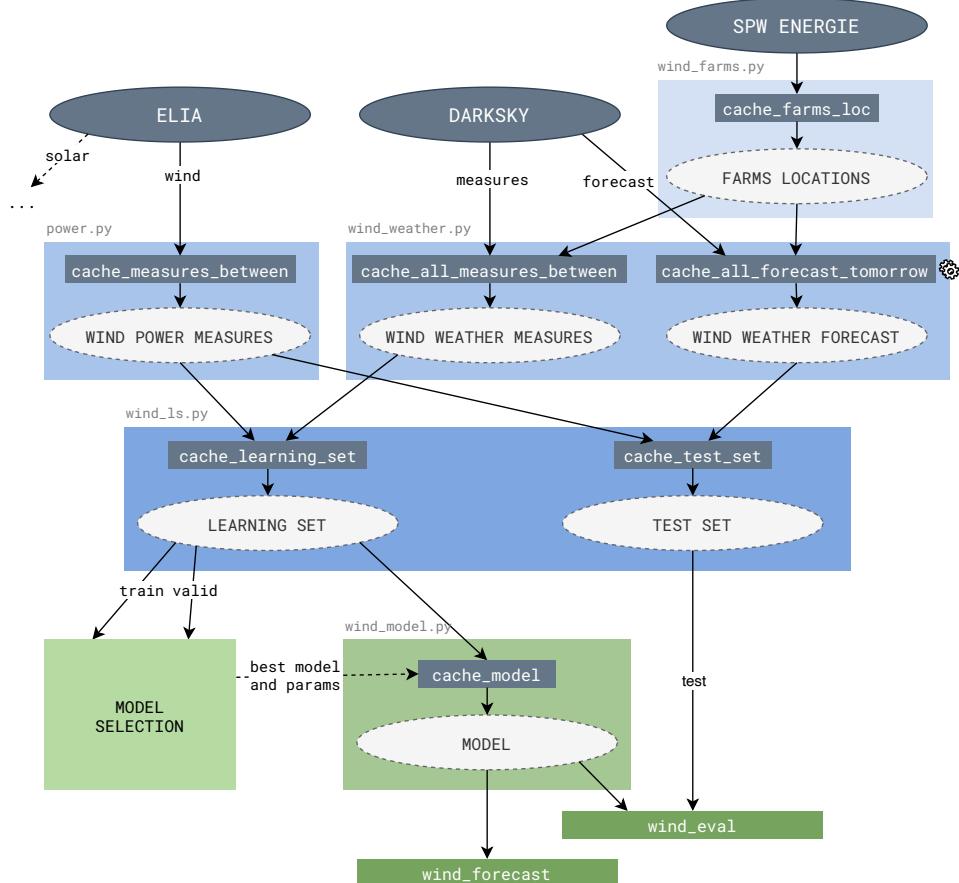


Figure 15 – Wind forecast computation flow

3.2.3 Feasibility

The above mentioned computation flow has allowed constituting a learning set of 67 7-dimensional inputs, resulting in 469 inputs, for one output: the wind power in Wallonia. The feasibility of the regression has quickly been concluded, by running a quick model

⁷As will be discussed later, the forecasting turned out to be unusable because the considered time period showed signs of non representability with respect to the usual wind power.

(random forest) on only one feature (wind speed measure in Namur) to predict the output. This model already achieved a MAE (mean absolute error) below 100 MW, whereas the maximum wind power observed in Wallonia, according to *Elia*, is 727 MW.

Furthermore, as can be seen on figure 16, the missingness on the 469 variables is really moderate, with almost only one chunk of 3 months that is not represented. If the wind present seasonal patterns, the fact that 2.25 years of data has been retrieved will solve the problem of these three months.

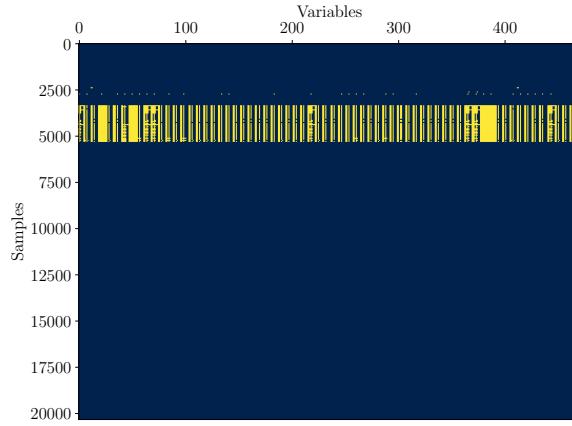


Figure 16 – Missingness in the learning set (2069 NA corresponding to 0.024 %)

Finally, by looking at figure 17, which corresponds to a kind of Wallonian power curve, we can see that a certain relation seems to exist between the mean wind speed at the 67 locations and the wind power in Wallonia.

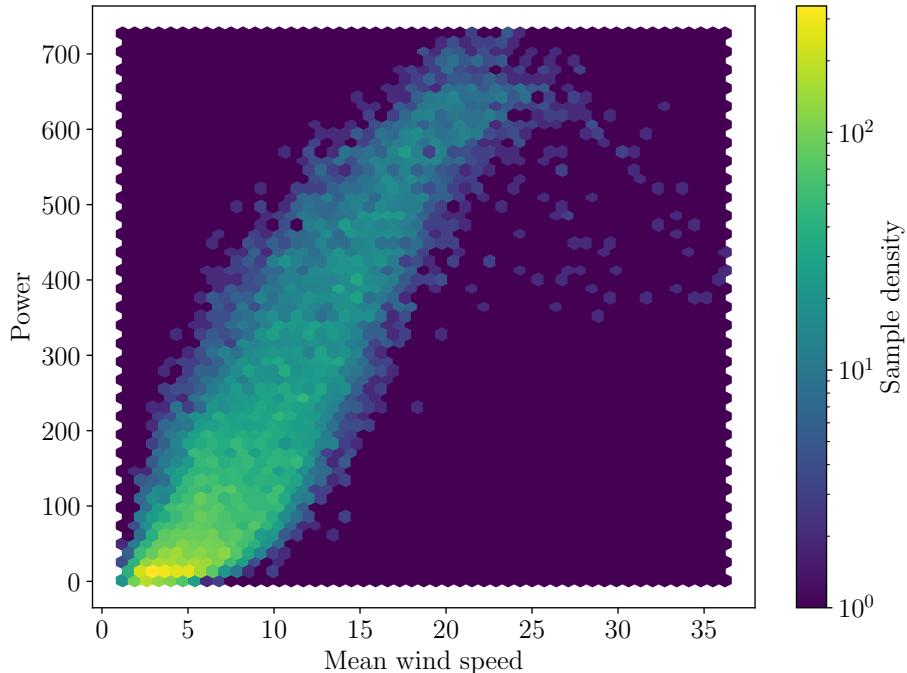


Figure 17 – Learning set sample density with respect to the total power and the mean wind speed at the 67 wind power plants location

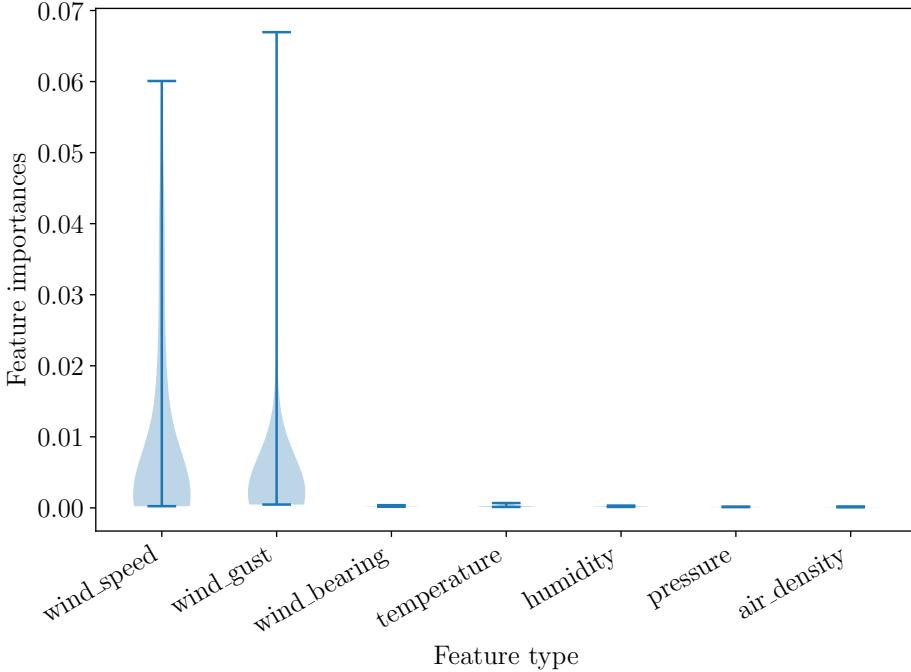


Figure 18 – Feature importances distribution with respect to the type of feature

3.2.4 Feature selection

After having identified that models such as random forests and extra trees performed very well on the whole learning set (469 inputs), we have considered feature selection, since some of the weather measurements are certainly not relevant to the problem.

To assess the feature importances, we have used the mean impurity reduction in the extra trees forest for each variable (mean variance reduction in the case of regression). This resulted in the explicit results of figure 18. The feature importances of each of the 469 variables are grouped by type of variable (wind speed, temperature, etc) and the density of feature importances for each type of variable has been estimated on the violin plot.

As can be clearly seen from the figure, the two most important features are the wind speed and wind gust, which totally makes sense, the other variables having negligible importance. We thus decided to keep these 2 types of variables only. Indeed, with mean feature importances of respectively 0.88 % and 0.52 %, the 134 variables constituted by these two type of features gather around 93.8 % of the mean impurity reduction.

Furthermore, by comparing the results obtained on unseen data, especially when the train-test split is not shuffled, the algorithms generalized better when working only on these two types of variables than when working with all of them. This feature selection have thus allowed overfitting reduction.

3.3 Model selection

The model selection has been performed on the *weather measures* learning set, without using any data coming from the *weather forecast* set. This set has been shuffled and divided in three parts:

- LS:

- Train set (50%);
 - Validation set (25%);
- TS:
 - Test set (25%);

We can already note that this way of splitting the learning set will probably lead to overly optimistic results because of the shuffling. Indeed, the wind power and wind weather are highly correlated from one hour to another. Fortunately, the *weather forecast* test set whose construction has been discussed above will fix this problem.

3.3.1 Supervised learning method selection

A certain number of supervised learning methods have been experimented. The performance of each model has been estimated by computing the MAE on the validation set for several combination of parameters for each model. All models have yielded validation set MAE between 31 MW and 33 MW, for all parameters combinations. The best scores obtained on the validation set are gathered in table 3.

Supervised Learning Method	MAE	nMAE [%]	Time ⁷ [s]
Random Forest (500 trees)	32.24	4.43	68
Extra Trees (500 trees)	31.05	4.27	24
Extra Trees (500 trees, ‘sqrt’ max features)	32.59	4.48	3
Extra Trees (500. ‘sqrt’ max features. ‘mae’ criterion)	32.76	4.50	248
MLP (hidden layers: 1024, 1024, 512, 128, 64)	31.99	4.40	430
Gradient Boosting (1000 trees, .1 learning rate, .7 subsample)	32.29	4.44	160

Table 3 – Best scores on validation set and computation times (MAE and MQL are expressed in MW)

Even if we can't assess the performance of each of these models on their validation set scores (having tested lots of parameters combination, one could be low by a stroke of luck), they leave use confident on the fact that no supervised learning method should be that much better than the others (see distribution of scores obtained on all the considered hyper-parameters in notebook `wind_models.ipynb`: all between 31 MW 33 MW).

We decided to focus our selection of supervised learning method on two well known family of supervised learning algorithms: tree ensembles and tree boosting, namely the Extra Trees and the Gradient Boosting methods. We will immediately focus on the MSE minimization criterion instead of MAE, since it is far more computationally efficient for datasets as large as the one that we use (around 18 000 samples). Moreover, the MAE criterion tested on extra trees (because they are the fastest method) has not shown better performance in terms of MAE on the validation set.

⁷Model trained with 4 threads available (for the extra trees), on a 2.9 GHz Intel Core i5 double core with 8 Go of 2133 MHz LPDDR3 RAM

3.3.2 Quantile predictions

In addition to the traditional estimation of the conditional expectation $\mathbb{E}\{Y|X\}$, we would like to have confidence intervals.

The gradient boosting method can be adapted straightforwardly, using an appropriate quantile loss in the boosting procedure. However, three separate trees will have to be built to predict the lower quantile, the upper quantile (using the quantile loss), and the expectation estimation (using a least square loss for this one).

For the extra trees, another approach is used, allowing to build a single forest. The idea is to have a higher minimum number of sample per leaf (for exemple, 10). The trees are built as usual, but instead of keeping only information such as the mean and variance for each leaf, we keep information on all the target values that have ended up in each leaf.

Then, at prediction time, a certain input will end up in T leaves (one leaf per tree), and its q -quantile can be estimated by interpolating between the all the samples that are in these T leaves, several interpolation methods are possible (Wikipedia — Percentile Interpolation). We have used the library `scikit-garden` that implements this by aditionnaly weighting the samples in each of the T considered leaves by the proportion of train samples that have ended up in each leaf (Meinshausen 2006). Of course, this method is thus far more computationally expensive at prediction time.

Two wrappers have been built to use these two models with a single `fit` and a single `predict` method, they can be found in the python module `cache/quantile_models.py`.

3.3.3 Hyper-parameters tuning

The hyper-parameters have been tuned by using 4-folds cross-validation on the LS (train and valid). The parameters having lead to the minimum mean MAE on the 4-folds are kept. These hyper-parameters are the following:

- Quantile Extra Trees: 1000 estimators, minimum number of samples required to split: 10, number of features to consider when looking for the best split: all.
- Quantile Gradient Boosting: 1000 estimators, learning rate: 0.1, number of features to consider when looking for the best split: all, subsample: 0.7.

3.4 Results

The two models performances are then assessed by computing the following metrics:

- MAE (mean absolute error);
- nMAE (normalized mean absolute error);
- nRMSE (normalized root mean squared error);
- MQL10 (mean quantile loss for 10th quantile);
- MQL90 (mean quantile loss for 90th quantile).

The normalized metrics (nMAE and nRMSE) are the usual metrics divided by the maximum observed true value.

The mean quantile loss is defined by

$$\alpha(y, q) = |y - q| \begin{cases} \alpha & \text{if } y > q \\ (1 - \alpha) & \text{else} \end{cases}$$

which allows to penalize both when quantiles aren't tight and when they are misplaced.

The two models are trained in 188 s and 995 s on the LS for the extra trees and the gradient boosting methods respectively⁸. The extra trees take advantage of their parallelization ability.

The results of the two algorithms on the two test sets are available in table 4. It contains the results on the following test sets (training each time on appropriate data for the considered test set):

- Test set obtained by the shuffled sample split discussed earlier (TS: 25 % of the data)
- Test set constituted of the last 25 % of the data, corresponding approximately to six months.
- *Weather forecast* test set where the weather forecast for day D has been requested on day $D - 1$ at 11 o'clock. This allows to assess the quality of our model in real conditions, where weather measures for the next day are obviously not available. Unfortunately, this test set is only constituted of weather forecast between April 5 and April 27, 2020.
- *Weather measures* test set constituted of the same period as the *Weather forecast* test set, in order to have a comparison.

Test set	Method	MAE	nMAE	nRMSE	MQL10	MQL90	Time ⁹ [s]
Shuffled TS (Elia MAE: 34.87)	Extra Trees	31.14	4.28 %	5.97 %	53.65	57.78	752
	Gradient Boosting	32.90	4.52 %	5.99 %	52.54	51.90	0
Last 6 months (Elia MAE: 40.16)	Extra Trees	40.43	5.56 %	7.45 %	64.58	55.58	710
	Gradient Boosting	41.91	5.76 %	7.69 %	59.25	44.58	0
Weather forecast (Elia MAE: 21.51)	Extra Trees	36.86	5.07 %	7.51 %	52.51	52.71	95
	Gradient Boosting	35.71	4.91 %	7.05 %	51.33	47.14	0
Weather measures (Elia MAE: 21.51)	Extra Trees	33.26	4.57 %	6.44 %	55.66	47.26	115
	Gradient Boosting	31.62	4.35 %	5.87 %	56.18	41.77	0

Table 4 – Results of the two algorithms on the learning sets (MAE and MQL are expressed in MW)

These results will be discussed below. In addition, for each of the test sets we indicate the MAE obtained by Elia on day-ahead forecast for the considered test set.

3.4.1 Graphical results

As far as the *weather forecast* test set is concerned, we can visualize the predictions (because they are contiguous in time), and because we are on equal terms with Elia (day-

⁸See footnote 7

⁹See footnote 7

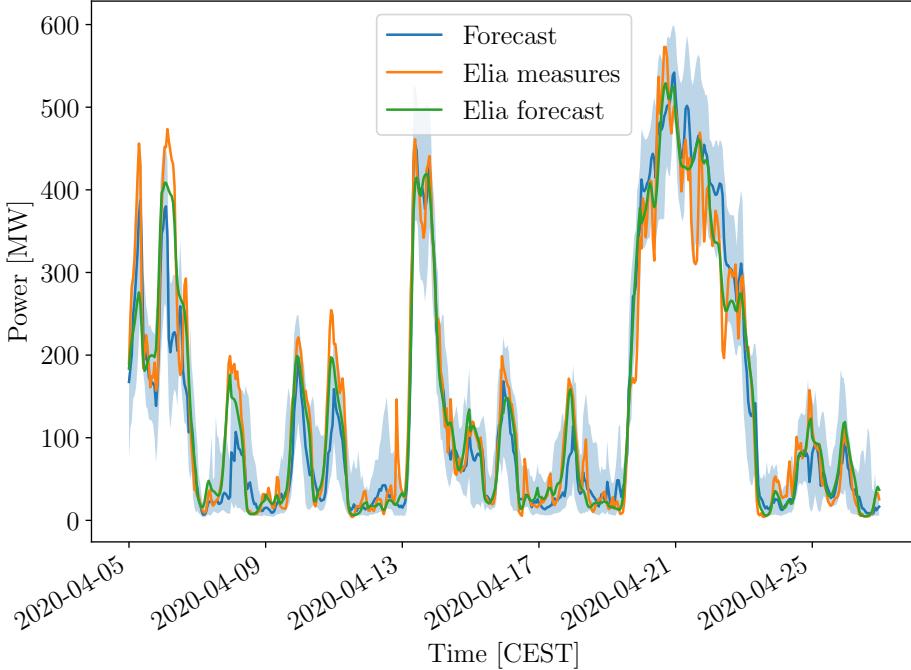


Figure 19 – Quantile Extra Trees forecast

ahead forecast using weather predictions). Our predictions and Elia predictions are shown on figures 19 and 20, together with the observed power. These graphs give us the insight that the model is really able to capture the main trends in the power produced, with relatively good predictions. It seems to suffer from the extreme cases where there is a lot of wind or very few. Maybe, this can be explained by the fact that these situations are less represented in the learning set or because the power have a residual error that is higher in these situations ($\mathbb{V}\{Y|X \in \text{high wind, low wind}\}$ is higher).

Moreover, on figures 21a and 21b, we can see the absolute residuals in order to get a more precise idea of the distribution of errors, and how they compare with Elia's errors.

3.4.2 Discussion

Let's go back to the numerical results of table 4.

By comparing the first two test set, that have exactly the same size, we can notice, as was predicted earlier, that the strong correlation between the samples seems to have introduced a leakage of information from the learning set to the test set in the case of the shuffled TS. Then, we should not consider the results of the first test set as very reliable. Indeed, while the scores are comparable to those of Elia on the second test set, we are fraudulently better on the first test set.

On the second test set, however, we have something that is quite encouraging, with a MAE around 41 MW (corresponding to 5.64 %) while Elia does 40.16 MW. Obviously, if we reach results so close to Elia, it is because we work on weather measures instead of weather forecast like Elia that makes its prediction the day before.

However, we have built a *weather forecast* test set in order to assess how our model degrade when using in real condition: on weather forecast data. To compare things that

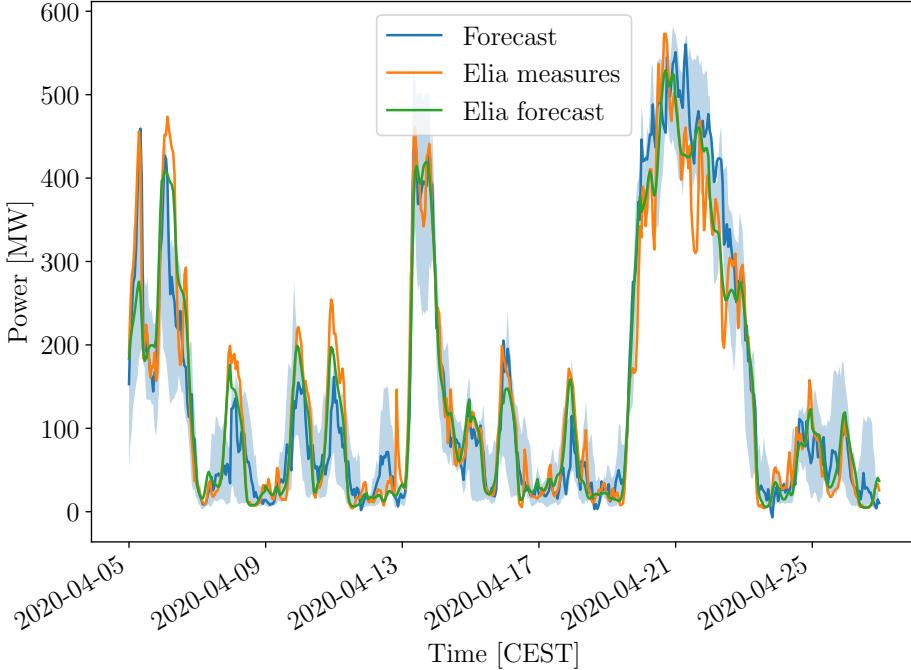


Figure 20 – Quantile Gradient Boosting forecast

are comparable, we have also used a *weather measures* test set on the same period. This results are extremely good, with the MAE going from 33.26 MW only to 36.86 MW when going from weather measures to weather forecast for the Quantile Extra Trees, and from 31.62 MW only to 35.71 MW for the Quantile Gradient Boosting.

Nevertheless, these results have to be nuanced. Indeed, on this very specific period our model has not degraded much at all. But this period is far from being exhaustive (only 22 days in April 2020), and we have a sign that this perior is not very representative of the general wind weather-power distribution. This can be inferred from the fact that while Elia has a MAE of only 21 MW on this test set, while it has reached a MAE of 34 MW on the shuffled test set, and 40 MW on the last 6 months. That may be an sign that the considered period corresponds to a period were the forecasting was more “easy”.

That being said, given the derivative of the sum of all wind power curves in Wallonia, we are confident that the MAE should less than double when used in forecasting mode, if we can count on day-ahead MAE of wind speed forecast below 2.4 m/s (El-Fouly, El-Saadany, and Salama 2008), even if we have no formal guarantee of that, as the power curve are not directly related to the learnt models.

Nevertheless, the results are quite good, with a MAE on weather measurements very close than this of Elia (on weather forecast). Furthermore, we are quite happy with these results, as some of the best results at the regional level based on NWP are obtained by an hybrid approach called WPPT (Croonenbroeck and Dahl 2014). In this paper, they showed that they reach results between 3 % and 15 % in terms of nRMSE depending on the case study, and here the results are between 7 % and 8 % on weather measurements.

Finally, the two models are quite equivalents in term of performance, even if the Quantile Gradient Boosting seems to take the advantage for all the quantile losses (MQL). However, we think that it is preferable to use the Extra Trees for three reasons:

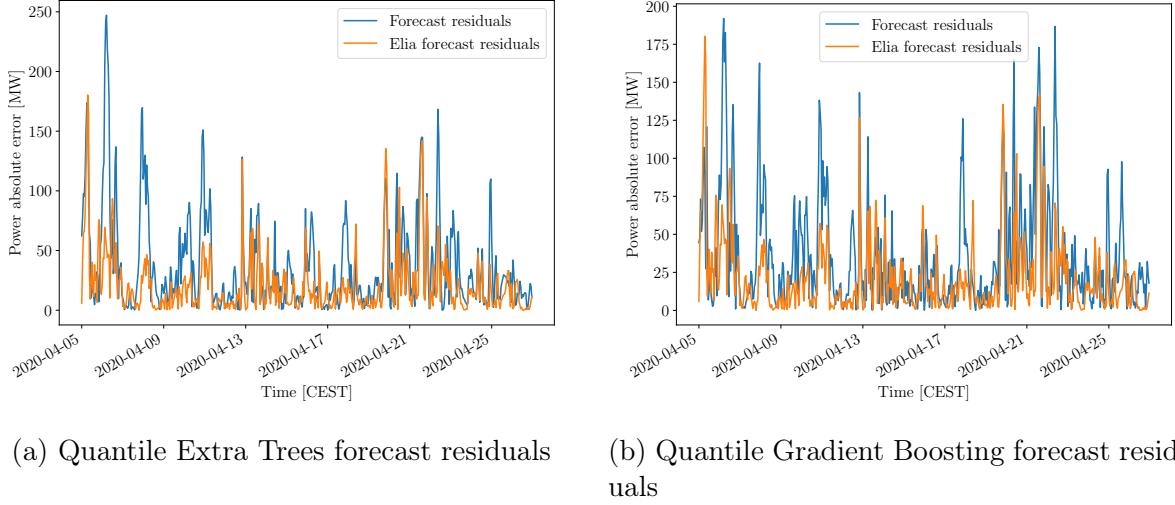


Figure 21 – Residuals on *weather forecast* test set

- They are really fast to train, we could thus profit from more data from Elia using a weather API that provide information every quarter, avoiding us the need to average the measures of Elia by four.
- They are slow to predict, but our task is to predict for the next day only, which is thus really fast.
- As a single tree is grown, and that quantiles for a given sample are estimated from the content of the same leaves, the quantiles are coherent. It means that a lower quantile is never above the upper quantile or the median prediction, while it can be the case with the three estimators built in the Quantile Gradient Boosting.

3.5 Limitations and possible improvements

To conclude on these results, we could say that they are very decent, although we have certainly not succeed in reaching the accuracy of Elia on the forecasting task (this could not be verified). In a word, we reach the results of Elia on forecast when using weather measures as input. The reason why Elia performs better could come from the fact that they have access to more precise data and weather measures, and maybe they have access to power data for each wind farm instead of aggregated data.

Another limitation of our work is that the availability of data forced us to work at the Wallonia level, while the project was focused on Liege province. If we had to make predictions for the Liege province only, we would follow the following procedure:

- Forecasting the total wind power in Wallonia
- Forecasting the wind power at each power plant using the physical model (even if it clearly overestimates, each power plant receives a weight proportional to the cubed wind speed, in addition to respect the cut-in, cut-out and rated speeds)
- Multiplying the total wind power by the ratio of wind power coming the Liege province according to the physical model.

On the other hand, if we had to make predictions for the whole Wallonia, the solar

power forecasting approach developed in the first two sections would be able to scale straightforwardly.

Finally, an interesting follow up work on this part would be to go for a hybrid approach, where we post-process the output of the physical model (in addition to the weather data, or not). Indeed, as can be seen from figure 14b, the physical model, seems to be promising if we consider reshaping the power curves, for example by using a machine learning model between the physical model output and the corrected output.

Conclusion

A first general remark is that we find regrettable that a large part of the data we needed were either very hard or impossible to obtain. For instance, we looked for a long time after a training dataset for the panels mapping and we never found quality aerial images for the province of Liège. Also, finding free of access irradiance data took us quite some time, and we are still facing limitations: we can only request for one location, and will eventually be limited to 10 API calls a day. The same applies for the weather data retrieved for the wind power production part.

Another important remark concerning the data availability is that we weren't able to collect past forecast for the weather data, but only measures for the past. This is really annoying as it completely prevents us to provide performance metrics in real conditions for our models. We are limited to the few forecast data retrieved manually for the period of April 2020. Nevertheless, we are satisfied with the results obtained for each branch of this project.

Our photovoltaic panels mapping yields precise results, to a certain extent, and is sufficiently fast to cover large areas in a relatively low amount of time. Furthermore, as we implemented it, it could very well be applied to any region for which we have satellite images, with few additional overhead. However, we genuinely regret the lack of uncertainty about our estimations.

As far as our photovoltaic power models are concerned, we are quite surprised, hence satisfied, with the obtained accuracy in experimental settings (year 2019) but also in real-life settings (April). Also, even though Elia outperforms our models in a real-life setting, they still perform much better than simple baseline models. In addition, it should be kept in mind that Elia has very likely access to more and more accurate information than us and they could be using a much more complex model.

In regards to the wind power models, we are really satisfied with the results obtained in view of what we have seen as results in the scientific literature, and in view of the initial performance of our physical model. We may regret being (supposedly) outperformed by Elia on the forecast task. But, as mentioned above, this could be due to the differences in data sources between Elia and us. Future works could consider hybrid models, more complex or fine-tuned supervised learning models, and, if the data was available, learning smaller independent models for each wind farm.

Finally, in order to showcase our final results, we have built a little interactive website : <https://znkvzr.com/apps/big-data/>. It should be updated with new forecasts daily.

The github repository including the photovoltaic panels mapping section can be found here: [francois-rozet/adopptrs](#).

The github repository including the photovoltaic power production and wind power production section can be found here: [lgaspard/repf](#).

References

- [1] Joan Masó, Keith Pomakis, and Nuria Julia. “OpenGIS web map tile service implementation standard”. In: *Open Geospatial Consortium Inc* (2010), pp. 04–06.
- [2] Jiafan Yu, Zhecheng Wang, Arun Majumdar, and Ram Rajagopal. “DeepSolar: A machine learning framework to efficiently construct a solar deployment database in the United States”. In: *Joule* 2.12 (2018), pp. 2605–2617.
- [3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. “Rethinking the inception architecture for computer vision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.
- [4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.
- [5] Benjamin Bischke, Patrick Helber, Joachim Folz, Damian Borth, and Andreas Dengel. “Multi-task learning for segmentation of building footprints with deep neural networks”. In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 1480–1484.
- [6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.
- [7] Kyle Bradbury, Raghav Saboo, Timothy L Johnson, Jordan M Malof, Arjun Devarajan, Wuming Zhang, Leslie M Collins, and Richard G Newell. “Distributed solar photovoltaic array location and extent dataset for remote sensing object identification”. In: *Scientific data* 3.1 (2016), pp. 1–9.
- [8] Jordan M Malof, Leslie M Collins, Kyle Bradbury, and Richard G Newell. “A deep convolutional neural network and a random forest classifier for solar photovoltaic array detection in aerial imagery”. In: *2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA)*. IEEE. 2016, pp. 650–654.
- [9] Abhishek Dutta and Andrew Zisserman. “The VGG image annotator (VIA)”. In: *arXiv preprint arXiv:1904.10699* (2019).
- [10] Alfredo Nespoli, Emanuele Ogliari, Sonia Leva, Alessandro Massi Pavan, Adel Mellit, Vanni Lughì, and Alberto Dolara. “Day-ahead photovoltaic forecasting: A comparison of the most effective techniques”. In: *Energies* 12.9 (2019), p. 1621.
- [11] Alexis Bocquet, Andrea Michiorri, Arthur Bossavy, Robin Girard, and Georges Kariniotakis. “Assessment of probabilistic PV production forecasts performance in an operational context”. In: 2016.
- [12] Conor Sweeney, Ricardo J Bessa, Jethro Browell, and Pierre Pinson. “The future of forecasting for renewable energy”. In: *Wiley Interdisciplinary Reviews: Energy and Environment* (2019), e365.
- [13] Jakob W Messner, Pierre Pinson, Jethro Browell, Mathias B Bjerregård, and Irene Schicker. “Evaluation of wind power forecasts—An up-to-date view”. In: *Wind Energy* (2020).
- [14] Silvio Matysik and Lucas Bauer. *Wind turbine models*. 2020. URL: <https://www.wind-turbine-models.com>.
- [15] SPW-Energie. *Situation Eolien Wallonie 2018*. 2018. URL: <https://energie.wallonie.be/fr/le-vent-pour-produire-de-l-electricite.html?IDC=6170&IDD=11276>.

- [16] Elia. *Production photovoltaïque*. 2020. URL: <https://www.elia.be/fr/donnees-de-reseau/production/production-photovoltaïque>.
- [17] *Dark Sky API*. 2020. URL: <https://darksky.net/dev>.
- [18] Nicolai Meinshausen. “Quantile regression forests”. In: *Journal of Machine Learning Research* 7.Jun (2006), pp. 983–999.
- [19] Tarek HM El-Fouly, Ehab F El-Saadany, and Magdy MA Salama. “One day ahead prediction of wind speed and direction”. In: *IEEE transactions on energy conversion* 23.1 (2008), pp. 191–201.
- [20] Carsten Croonenbroeck and Christian Møller Dahl. “Accurate medium-term wind power forecasting in a censored classification framework”. In: *Energy* 73 (2014), pp. 221–232.

A Figures

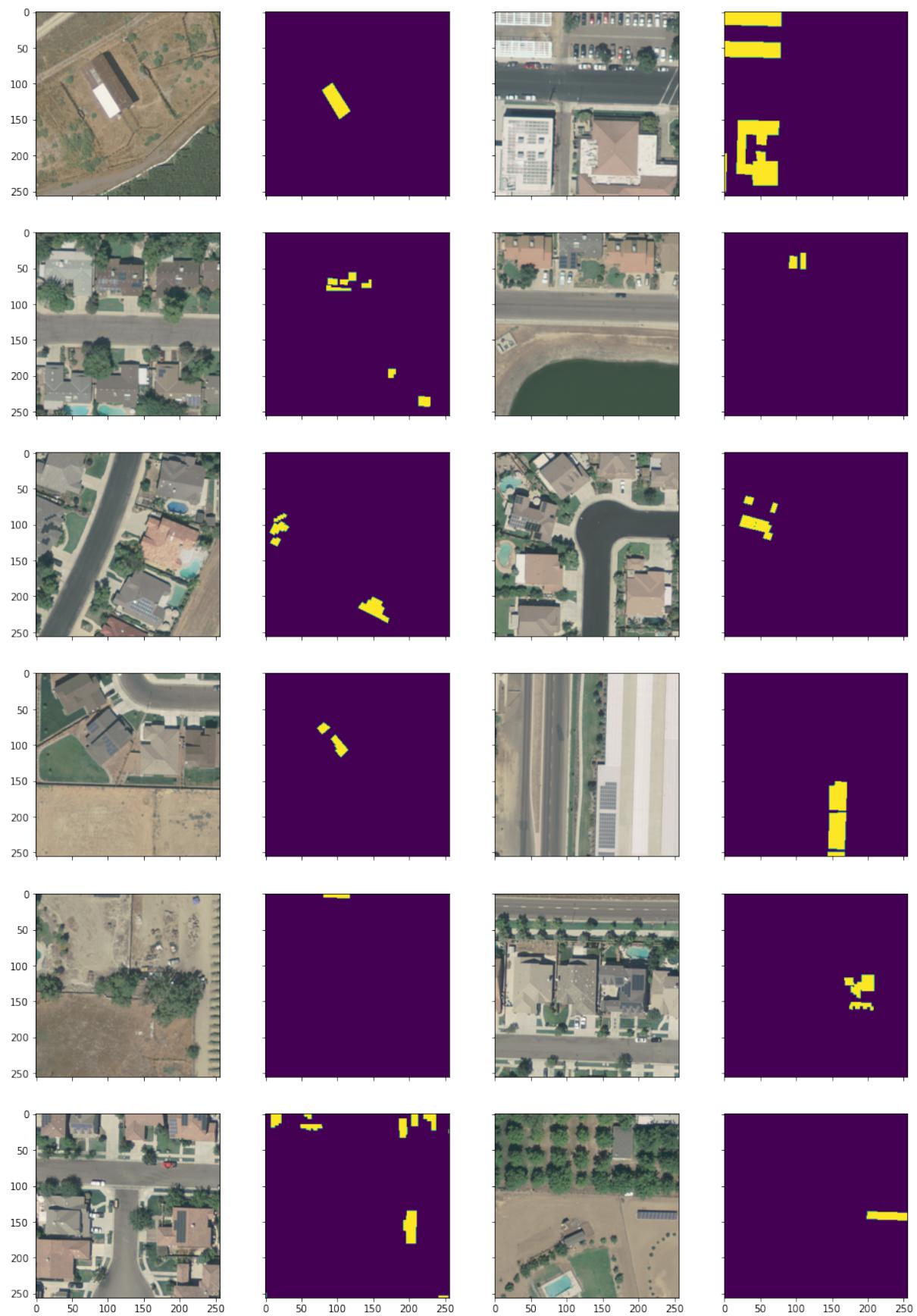


Figure 22 – Californian dataset sample, with the produced masks.



Figure 23 – Californian original (left) vs augmented (right) dataset sample.